Basler ace

USER'S MANUAL FOR GigE CAMERAS

Document Number: AW000893
Version: 16 Language: 000 (English)
Release Date: 8 August 2012

BASLER
the power of sight

For customers in the U.S.A.

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

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

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

For customers in Canada

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

Pour utilisateurs au Canada

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

Life Support Applications

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

Warranty Note

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

Contacting Basler Support Worldwide

Europe:

Basler AG
An der Strusbek 60 - 62
22926 Ahrensburg
Germany

Tel.: +49-4102-463-515
Fax.: +49-4102-463-599

bc.support.europe@baslerweb.com

Americas:

Basler, Inc.
855 Springdale Drive, Suite 203
Exton, PA 19341
U.S.A.

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

bc.support.usa@baslerweb.com

Asia:

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

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

bc.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 ace 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.

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 only, will be so designated.

1.2 General Specifications

Specification	acA640-90gm/gc	acA640-100gm/gc
Sensor Size (H x V pixels)	gm: 659 x 494 gc: 658 x 492	gm: 659 x 494 gc: 658 x 492
Sensor Type	Sony ICX424 AL/AQ Progressive scan CCD Global shutter	Sony ICX618 ALA/AQA Progressive scan CCD Global shutter
Optical Size	1/3"	1/4"
Pixel Size (H x V)	7.4 µm x 7.4 µm	5.6 µm x 5.6 µm
Max. Frame Rate (at full resolution)	90 fps	100 fps
Mono/Color	Mono or color (color models include a Bayer pattern RGB filter on the sensor)	Mono or color (color models include a Bayer pattern RGB filter on the sensor)
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)	
Pixel Data Formats	Mono Models: Mono 8, Mono 12, Mono 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed Color Models: Mono 8, Bayer BG 8, Bayer BG 12, Bayer BG 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed	
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal, via the Ethernet connection, or free run	
Exposure Control	Via external trigger signal or programmable via the camera API	
Camera Power Requirements	PoE (Power over Ethernet 802.3af compliant) or +12 VDC (± 10%), < 1% ripple, supplied via the camera’s 6-pin Hirose connector	
	~ 3.1 W when using Power over Ethernet ~ 2.7 W @ 12 VDC when supplied via the camera’s 6-pin connector	~ 2.3 W when using Power over Ethernet ~ 2.0 W @ 12 VDC when supplied via the camera’s 6-pin connector **Note:** When using extremely small AOIs, power consumption may increase to 2.4 W.

Table 1: General Specifications

Specification	acA640-90gm/gc	acA640-100gm/gc
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount; CS-mount	
Size (L x W x H)	42.0 mm x 29 mm x 29 mm (without lens adapter or connectors) 60.3 mm x 29 mm x 29 mm (with lens adapter and connectors)	
Weight	< 90 g	
Conformity	CE, UL (in preparation), FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE)	
Software Driver	Basler’s GigE Vision compliant pylon SDK including filter and performance drivers. Available for Windows or Linux in 32 and 64 bit versions.	

Table 1: General Specifications

Specification	acA645-100gm/gc	acA750-30gm/gc
Sensor Size (H x V pixels)	gm: 659 x 494 gc: 658 x 492	gm: 752 x 580 gc: 748 x 576
Sensor Type	Sony ICX414 AL/AQ Progressive scan CCD Global shutter	Sony ICX409 AL/AK Interlaced scan CCD Global shutter
Optical Size	1/2"	1/3"
Pixel Size (H x V)	9.9 µm x 9.9 µm	6.5 µm x 6.25 µm
Max. Frame Rate (at full resolution)	100 fps	30 fps
Mono/Color	Mono or color (color models include a Bayer pattern RGB filter on the sensor)	Mono or color (color models include a CMYeG color filter on the sensor)
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)	
Pixel Data Formats	Mono Models: Mono 8 Mono 12 Mono 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 Bayer BG 8 Bayer BG 12 Bayer BG 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed	Mono Models: Mono 8 Mono 12 Mono 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal, via the Ethernet connection, or free run	
Exposure Control	Via external trigger signal or programmable via the camera API	
Camera Power Requirements	PoE (Power over Ethernet 802.3af compliant) or +12 VDC (± 10%), < 1% ripple, supplied via the camera's 6-pin Hirose connector	
	~ 3.6 W when using Power over Ethernet ~ 3.3 W @ 12 VDC when supplied via the camera's 6-pin connector	~ 2.6 W when using Power over Ethernet ~ 2.4 W @ 12 VDC when supplied via the camera's 6-pin connector
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount; CS-mount	

Table 2: General Specifications

Specification	acA645-100gm/gc	acA750-30gm/gc
Size (L x W x H)	42.0 mm x 29 mm x 29 mm (without lens adapter or connectors) 60.3 mm x 29 mm x 29 mm (with lens adapter and connectors)	
Weight	< 90 g	
Conformity	CE, UL (in preparation), FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE)	
Software Driver	Basler's GigE Vision compliant pylon SDK including filter and performance drivers. Available for Windows or Linux in 32 and 64 bit versions.	

Table 2: General Specifications

Specification	acA780-75 gm/gc	acA1300-30gm/gc
Sensor Size (H x V pixels)	gm: 782 x 582 gc: 780 x 580	gm: 1296 x 966 gc: 1294 x 964
Sensor Type	Sony ICX415 AL/AQ Progressive scan CCD Global shutter	Sony ICX445 AL/AQ Progressive scan CCD Global shutter
Optical Size	1/2"	1/3"
Pixel Size (H x V)	8.3 µm x 8.3 µm	3.75 µm x 3.75 µm
Max. Frame Rate (at full resolution)	75 fps	30 fps
Mono/Color	Mono or color (color models include a Bayer pattern RGB filter on the sensor)	Mono or color (color models include a Bayer pattern RGB filter on the sensor)
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)	
Pixel Data Formats	Mono Models: Mono 8, Mono 12, Mono 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed Color Models: Mono 8, Bayer BG 8, Bayer BG 12, Bayer BG 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed	
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal, via the Ethernet connection, or free run	
Exposure Control	Via external trigger signal or programmable via the camera API	
Camera Power Requirements	PoE (Power over Ethernet 802.3af compliant) or +12 VDC (± 10%), < 1% ripple, supplied via the camera's 6-pin Hirose connector	
	~ 3.6 W when using Power over Ethernet ~ 3.3 W @ 12 VDC when supplied via the camera's 6-pin connector	~ 2.5 W when using Power over Ethernet ~ 2.2 W @ 12 VDC when supplied via the camera's 6-pin connector **Note:** When using extremely small AOIs, power consumption may increase to 2.9 W.
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount; CS-mount	
Size (L x W x H)	42.0 mm x 29 mm x 29 mm (without lens adapter or connectors) 60.3 mm x 29 mm x 29 mm (with lens adapter and connectors)	
Weight	< 90 g	

Table 3: General Specifications

Specification	acA780-75 gm/gc	acA1300-30gm/gc
Conformity	CE, UL (in preparation), FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE)	
Software Driver	Basler’s GigE Vision compliant pylon SDK including filter and performance drivers. Available for Windows or Linux in 32 and 64 bit versions.	

Table 3: General Specifications

<table>
<tr><th>Specification</th><th>acA1600-20gm/gc</th><th>acA2000-50gm/gc</th></tr>
<tr><td>Sensor Size
(H x V pixels)</td><td>gm: 1628 x 1236
gc: 1624 x 1234</td><td>gm: 2048 x 1088
gc: 2046 x 1086</td></tr>
<tr><td>Sensor Type</td><td>Sony ICX274 AL/AQ
Progressive scan CCD
Global shutter</td><td>CMOSIS CMV2000-2E5M / CMV2000-2E5C
Progressive scan CMOS
Global shutter</td></tr>
<tr><td>Optical Size</td><td>1/1.8"</td><td>2/3"</td></tr>
<tr><td>Pixel Size</td><td>4.4 µm x 4.4 µm</td><td>5.5 µm x 5.5 µm</td></tr>
<tr><td>Max. Frame Rate
(at full resolution)</td><td>20 fps</td><td>50 fps</td></tr>
<tr><td>Mono/Color</td><td colspan="2">Mono or color (color models include a Bayer pattern RGB filter on the sensor)</td></tr>
<tr><td>Data Output Type</td><td colspan="2">Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)</td></tr>
<tr><td>Pixel Data Formats</td><td>Mono Models:
Mono 8
Mono 12
Mono 12 Packed
YUV 4:2:2 Packed
YUV 4:2:2 (YUYV) Packed

Color Models:
Mono 8
Bayer BG 8
Bayer BG 12
Bayer BG 12 Packed
YUV 4:2:2 Packed
YUV 4:2:2 (YUYV) Packed</td><td>Mono Models:
Mono 8
Mono 12
Mono 12 Packed
YUV 4:2:2 Packed
YUV 4:2:2 (YUYV) Packed

Color Models:
Mono 8
Bayer GR 8
Bayer GR 12
Bayer GR 12 Packed
YUV 4:2:2 Packed
YUV 4:2:2 (YUYV) Packed</td></tr>
<tr><td>ADC Bit Depth</td><td colspan="2">12 bits</td></tr>
<tr><td>Synchronization</td><td colspan="2">Via external trigger signal, via the Ethernet connection, or free run</td></tr>
<tr><td>Exposure Control</td><td colspan="2">Via external trigger signal or programmable via the camera API</td></tr>
<tr><td rowspan="2">Camera Power Requirements</td><td colspan="2">PoE (Power over Ethernet 802.3af compliant) or
+12 VDC (±10%), < 1% ripple, supplied via the camera’s 6-pin Hirose connector</td></tr>
<tr><td>~ 3.4 W when using Power over Ethernet
~ 2.9 W @ 12 VDC when supplied via the camera’s 6-pin connector</td><td>~ 2.8 W when using Power over Ethernet
~ 2.5 W @ 12 VDC when supplied via the camera’s 6-pin connector</td></tr>
<tr><td>I/O Ports</td><td colspan="2">1 opto-isolated input line and 1 opto-isolated output line</td></tr>
<tr><td>Lens Adapter</td><td>C-mount; CS-mount</td><td>C-mount</td></tr>
<tr><td>Size (L x W x H)</td><td colspan="2">42.0 mm x 29 mm x 29 mm (without lens adapter or connectors)
60.3 mm x 29 mm x 29 mm (with lens adapter and connectors)</td></tr>
</table>

Table 4: General Specifications

Specification	acA1600-20gm/gc	acA2000-50gm/gc
Weight	< 90 g	
Conformity	CE, UL (in preparation), FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE)	
Software Driver	Basler’s GigE Vision compliant pylon SDK including filter and performance drivers. Available for Windows or Linux in 32 and 64 bit versions.	

Table 4: General Specifications

Specification	acA2000-50gmNIR	acA2040-25gm/gc
Sensor Size (H x V pixels)	2048 x 1088	gm: 2048 x 2048 gc: 2046 x 2046
Sensor Type	CMOSIS CMV2000-2E12M Progressive scan CMOS Global shutter	CMOSIS CMV4000-2E5M / CMV4000-2EM5C Progressive scan CMOS Global shutter
Optical Size	2/3"	1"
Pixel Size	5.5 µm x 5.5 µm	5.5 µm x 5.5 µm
Max. Frame Rate (at full resolution)	50 fps	25 fps
Mono/Color	Mono (NIR)	Mono or color (color models include a Bayer pattern RGB filter on the sensor)
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)	
Pixel Data Formats	Mono and Mono (NIR) models: Mono 8, Mono 12, Mono 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed Color Models: Mono 8, Bayer GR 8, Bayer GR 12, Bayer GR 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed	
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal, via the Ethernet connection, or free run	
Exposure Control	Via external trigger signal or programmable via the camera API	
Camera Power Requirements	PoE (Power over Ethernet 802.3af compliant) or +12 VDC (±10%), < 1% ripple, supplied via the camera’s 6-pin Hirose connector	
	~ 2.8 W when using Power over Ethernet ~ 2.5 W @ 12 VDC when supplied via the camera’s 6-pin connector	~ 2.9 W when using Power over Ethernet ~ 2.6 W @ 12 VDC when supplied via the camera’s 6-pin connector
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount	
Size (L x W x H)	42.0 mm x 29 mm x 29 mm (without lens adapter or connectors) 60.3 mm x 29 mm x 29 mm (with lens adapter and connectors)	

Table 5: General Specifications

Specification	acA2000-50gmNIR	acA2040-25gm/gc
Weight	< 90 g	
Conformity	CE, UL (in preparation), FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE)	
Software Driver	Basler’s GigE Vision compliant pylon SDK including filter and performance drivers. Available for Windows or Linux in 32 and 64 bit versions.	

Table 5: General Specifications

Specification	acA2040-25gmNIR	acA2500-14gm/gc
Sensor Size (H x V pixels)	2048 x 2048	gm: 2592 x 1944 gc: 2590 x 1942
Sensor Type	CMOSIS CMV4000-2E12M Progressive scan CMOS Global shutter	Aptina MT9P031 Progressive scan CMOS Rolling shutter
Optical Size	1"	1/2.5"
Pixel Size	5.5 µm x 5.5 µm	2.2 µm x 2.2 µm
Max. Frame Rate (at full resolution)	25 fps	14.6 fps
Mono/Color	Mono (NIR)	Mono or color (color models include a Bayer pattern RGB filter on the sensor)
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)	
Pixel Data Formats	Mono and Mono (NIR) Models: Mono 8 Mono 12 Mono 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 Bayer GR 8 Bayer GR 12 Bayer GR 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed	Mono Models: Mono 8 Mono 12 Mono 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 Bayer BG 8 Bayer BG 12 Bayer BG 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal, via the Ethernet connection, or free run	
Exposure Control	Via external trigger signal or programmable via the camera API	
Camera Power Requirements	PoE (Power over Ethernet 802.3af compliant) or +12 VDC (±10%), < 1% ripple, supplied via the camera’s 6-pin Hirose connector	
	~ 2.9 W when using Power over Ethernet ~ 2.6 W @ 12 VDC when supplied via the camera’s 6-pin connector	~ 2.5 W when using Power over Ethernet ~ 2.2 W @ 12 VDC when supplied via the camera’s 6-pin connector
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount	C-mount; CS-mount
Size (L x W x H)	42.0 mm x 29 mm x 29 mm (without lens adapter or connectors) 60.3 mm x 29 mm x 29 mm (with lens adapter and connectors)	

Table 6: General Specifications

Specification	acA2040-25gmNIR	acA2500-14gm/gc
Weight	< 90 g	
Conformity	CE, UL (in preparation), FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE)	
Software Driver	Basler’s GigE Vision compliant pylon SDK including filter and performance drivers. Available for Windows or Linux in 32 and 64 bit versions.	

Table 6: General Specifications

1.3 Spectral Response

1.3.1 Mono Camera Spectral Response

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

ⓘ	The spectral response curves exclude lens characteristics and light source characteristics.

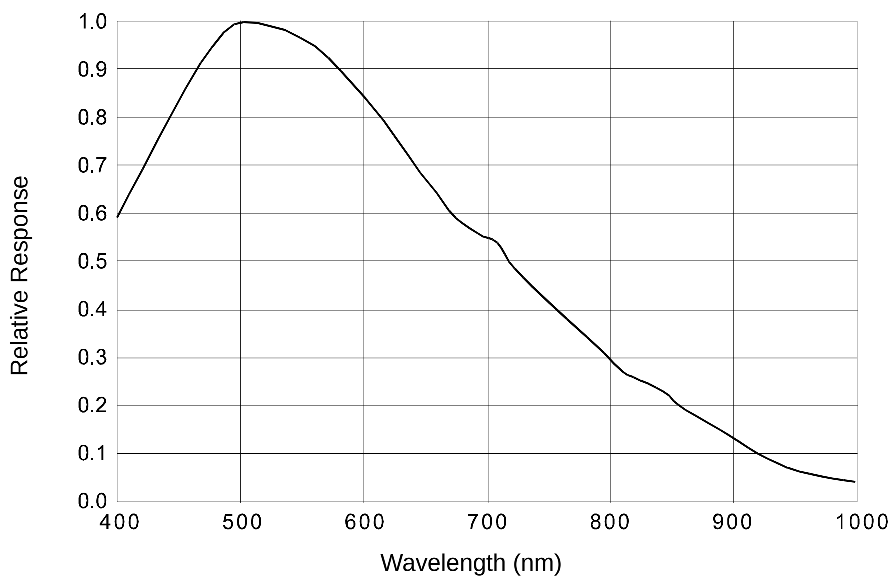


Fig. 1: acA640-90gm Spectral Response (From Sensor Data Sheet)

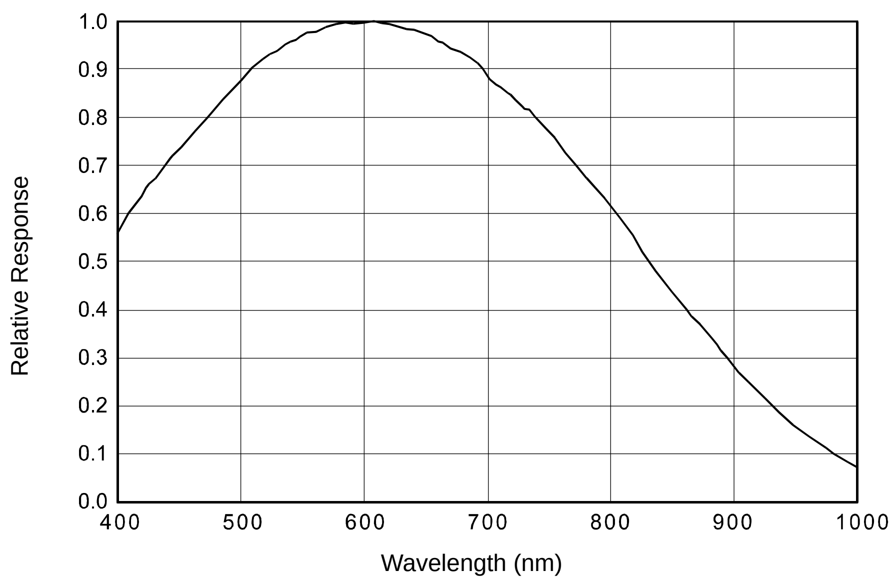


Fig. 2: acA640-100gm Spectral Response (From Sensor Data Sheet)

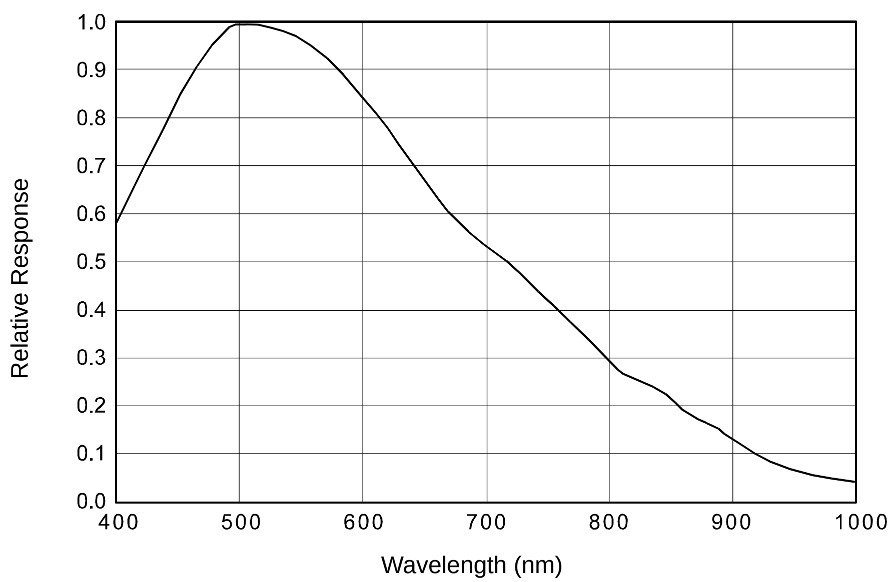


Fig. 3: acA645-100gm Spectral Response (From Sensor Data Sheet)

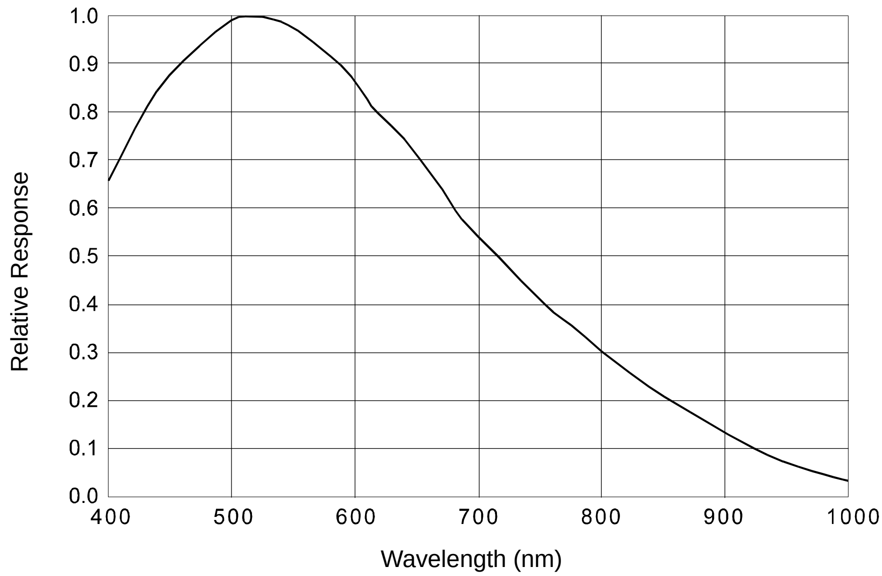


Fig. 4: acA750-30gm Spectral Response (From Sensor Data Sheet)

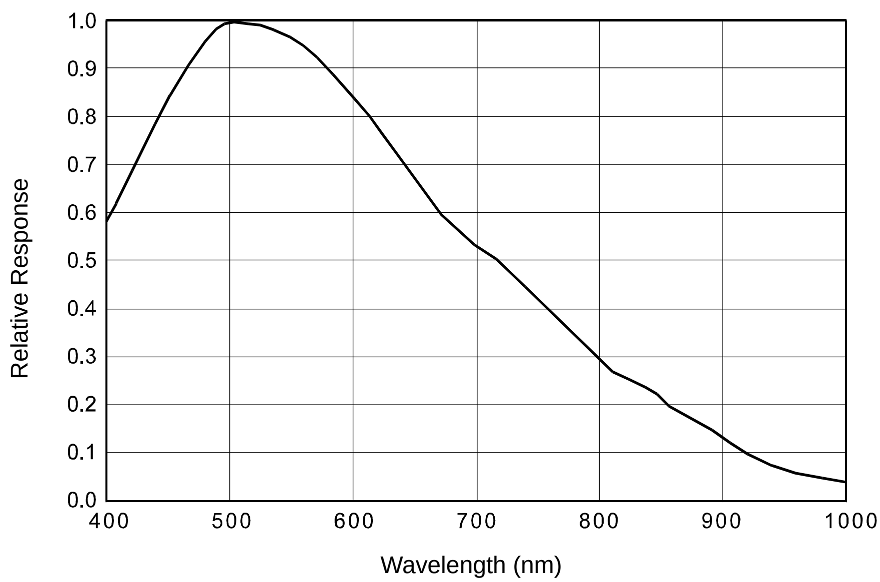


Fig. 5: acA780-75gm Spectral Response (From Sensor Data Sheet)

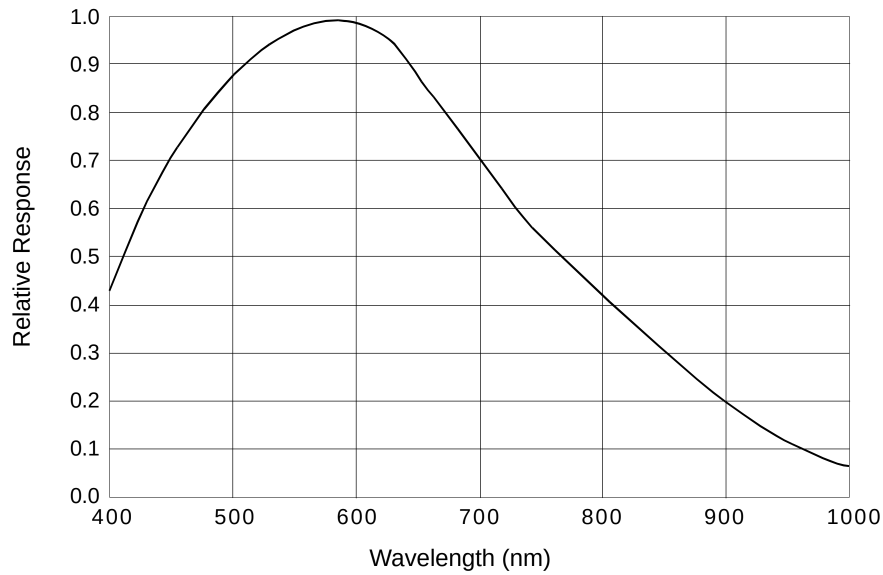


Fig. 6: acA1300-30gm Spectral Response (From Sensor Data Sheet)

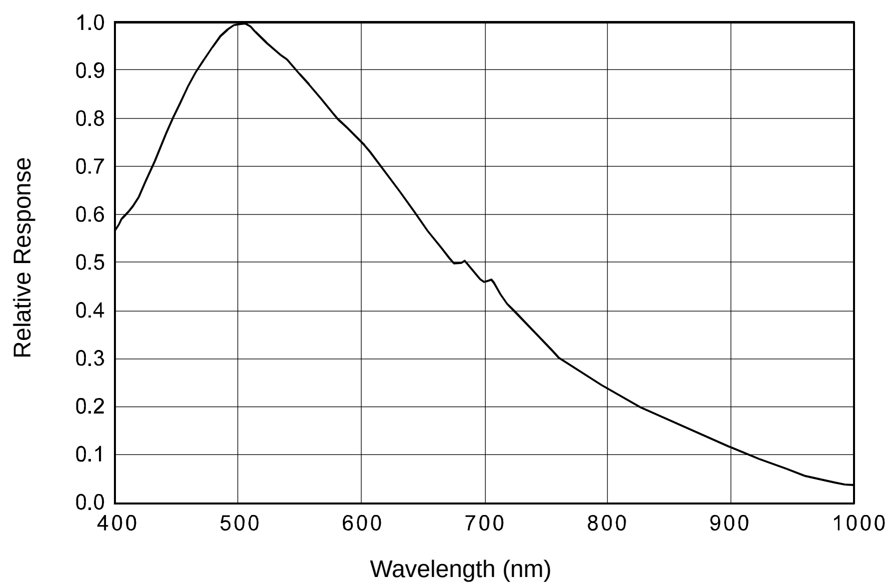


Fig. 7: acA1600-20gm Spectral Response (From Sensor Data Sheet)

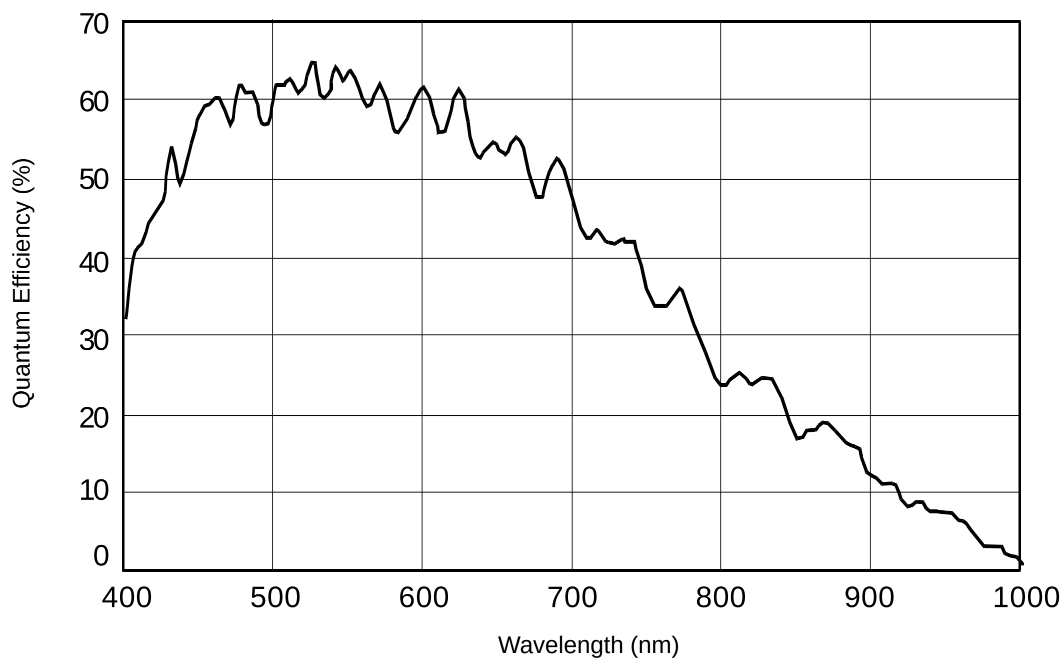


Fig. 8: acA2000-50gm, acA2040-25gm Spectral Response (From Sensor Data Sheet)

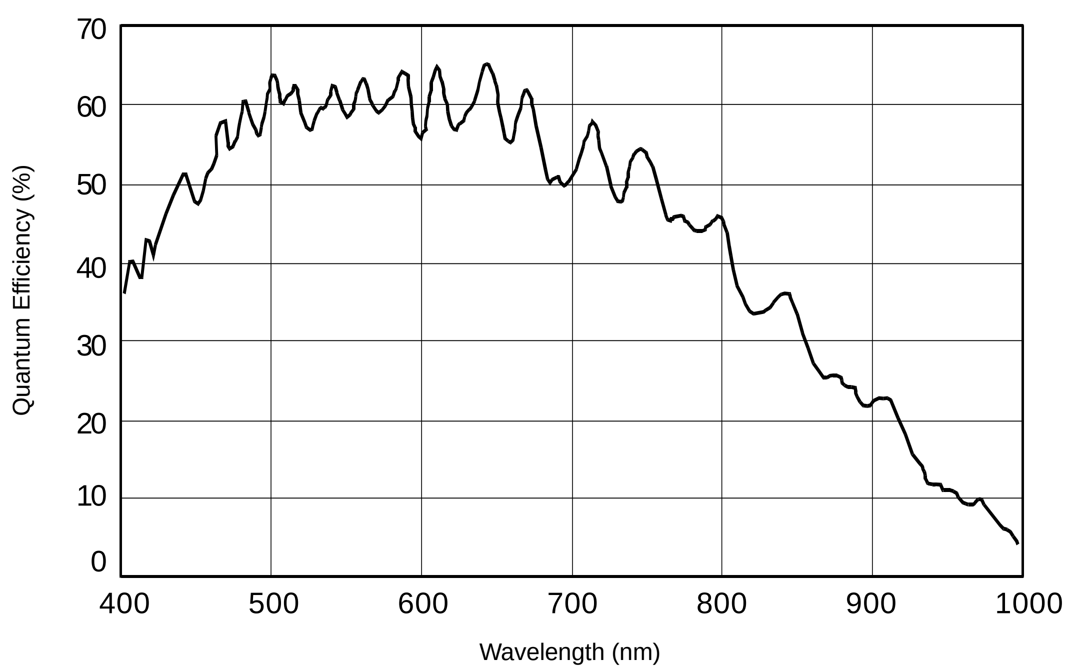


Fig. 9: acA2000-50gmNIR, acA2040-25gmNIR Spectral Response (From Sensor Data Sheet)

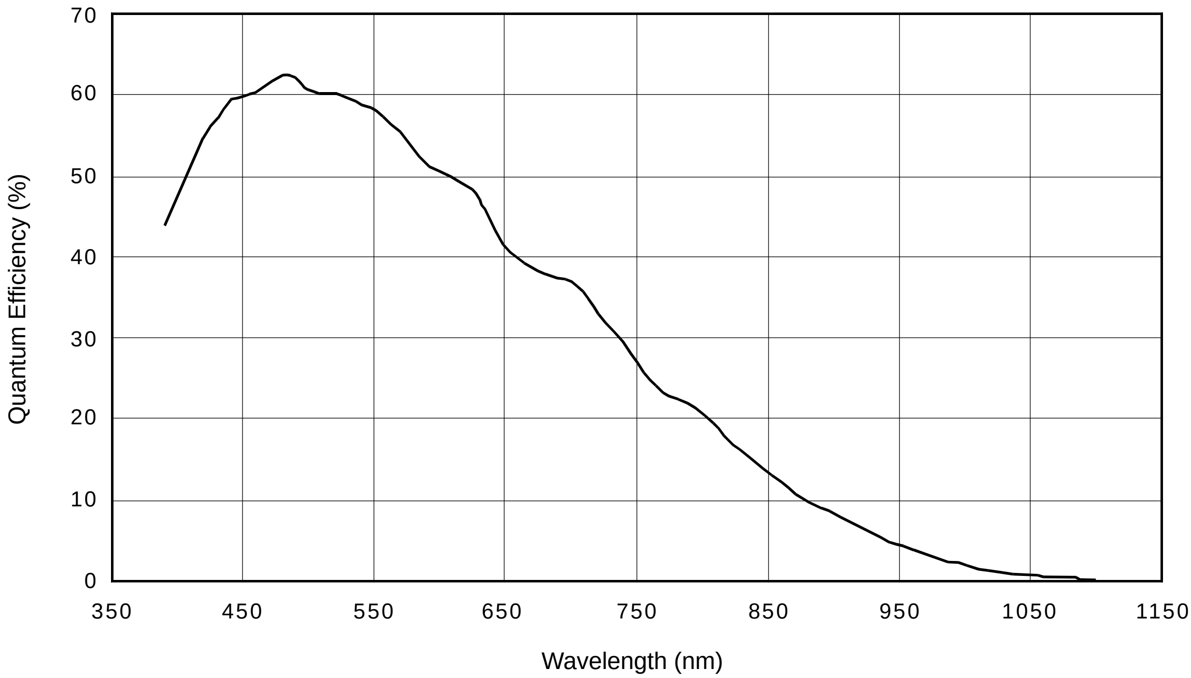


Fig. 10: acA2500-14gm Spectral Response (From Sensor Data Sheet)

1.3.2 Color Camera Spectral Response

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 700 ... 720 nm, and it should cut off from 700 ... 720 nm to 1100 nm. A suitable IR cut filter is built into the lens adapter on color models of the camera.

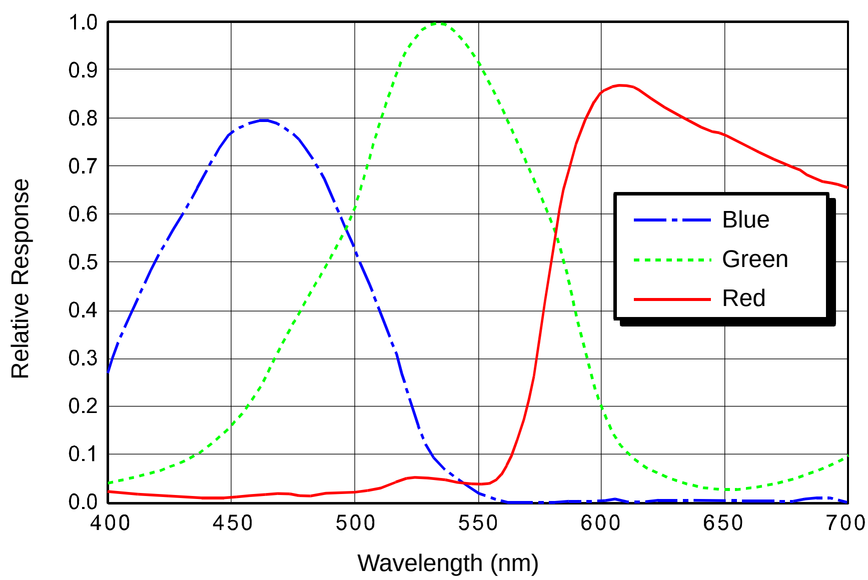


Fig. 11: acA640-90gc Spectral Response (From Sensor Data Sheet)

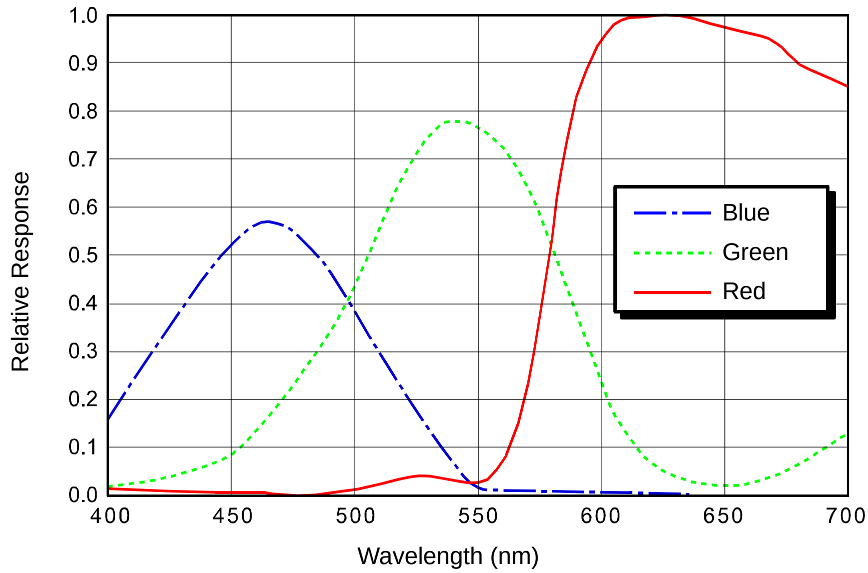


Fig. 12: acA640-100gc Spectral Response (From Sensor Data Sheet)

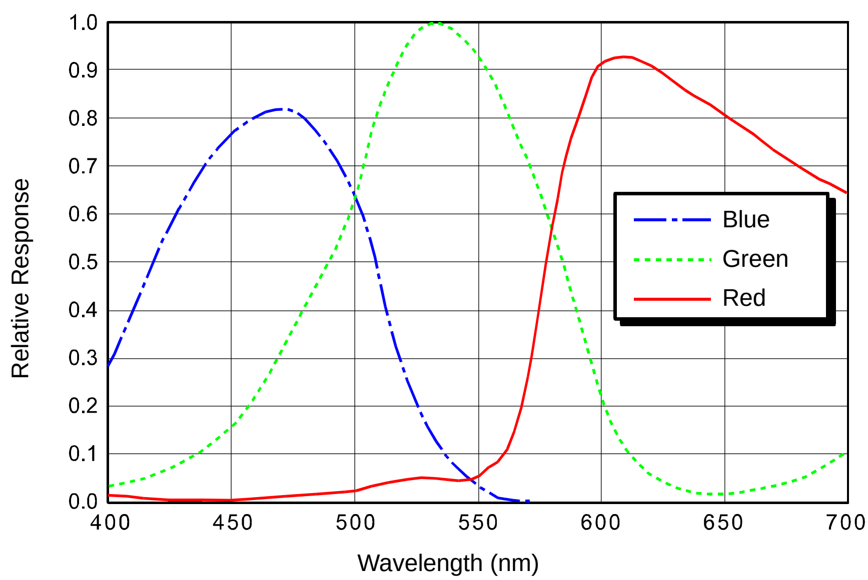


Fig. 13: acA645-100gc Spectral Response (From Sensor Data Sheet)

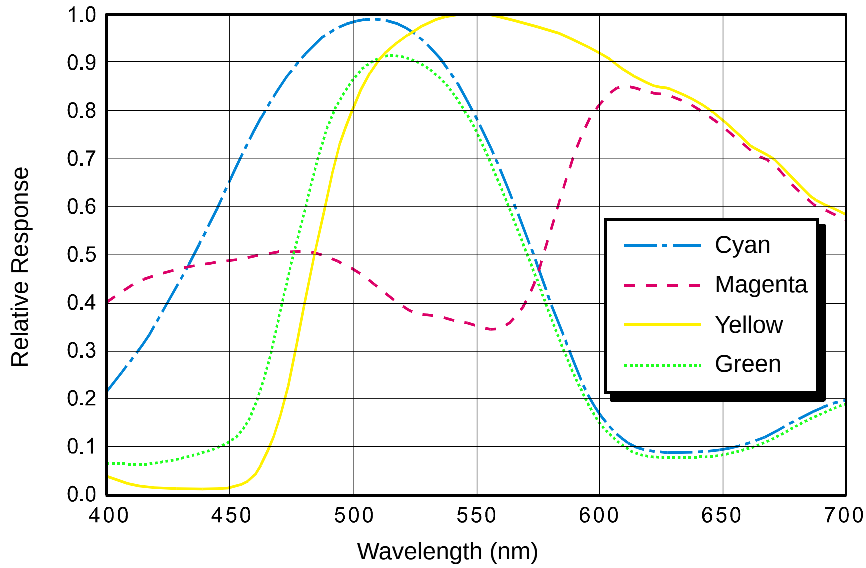


Fig. 14: acA750-30gc Spectral Response (From Sensor Data Sheet)

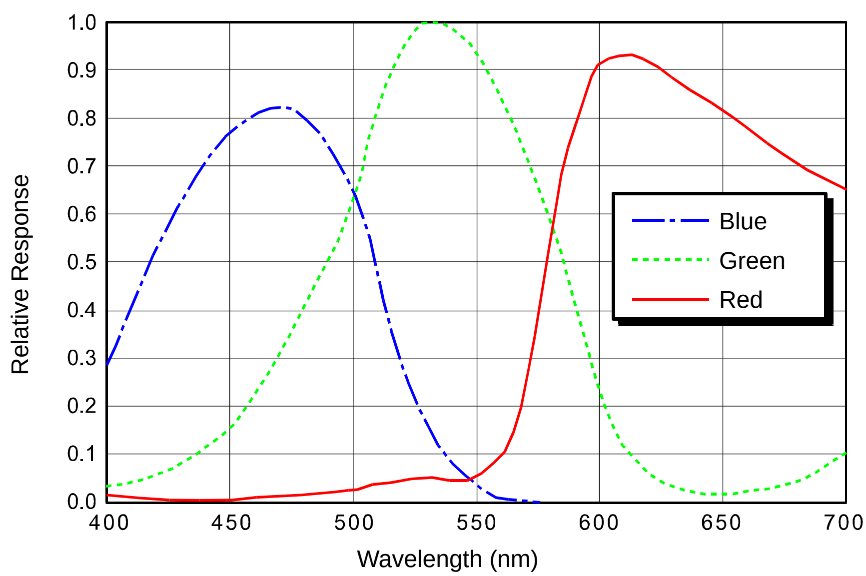


Fig. 15: acA780-75gc Spectral Response (From Sensor Data Sheet)

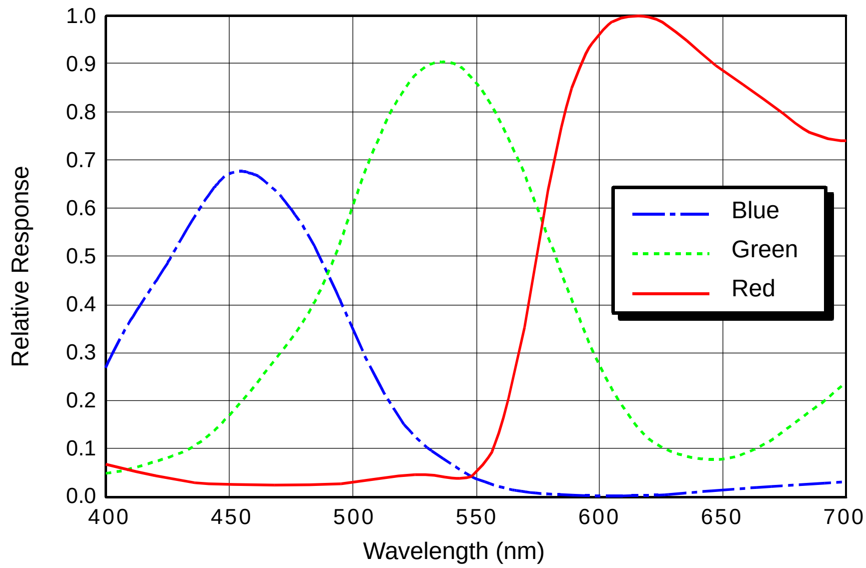


Fig. 16: acA1300-30gc Spectral Response (From Sensor Data Sheet)

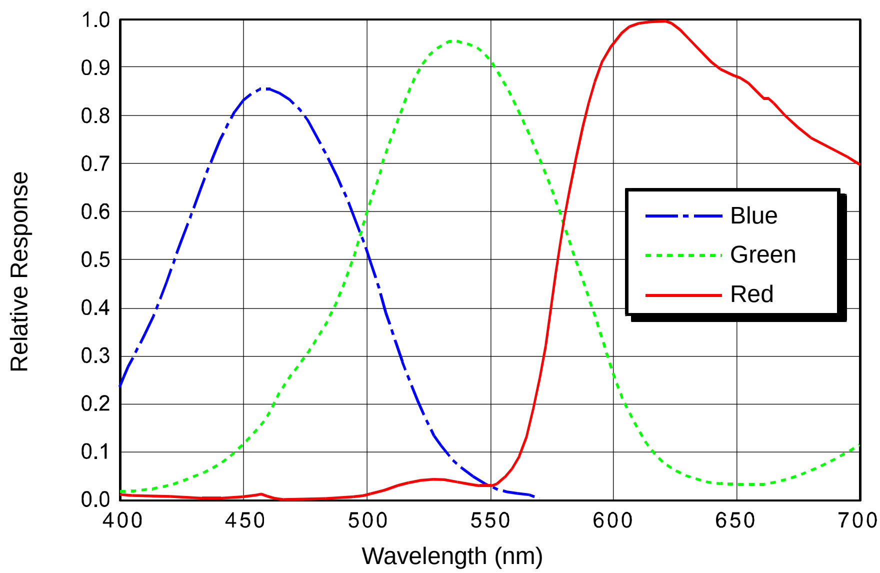


Fig. 17: acA1600-20gc Spectral Response (From Sensor Data Sheet)

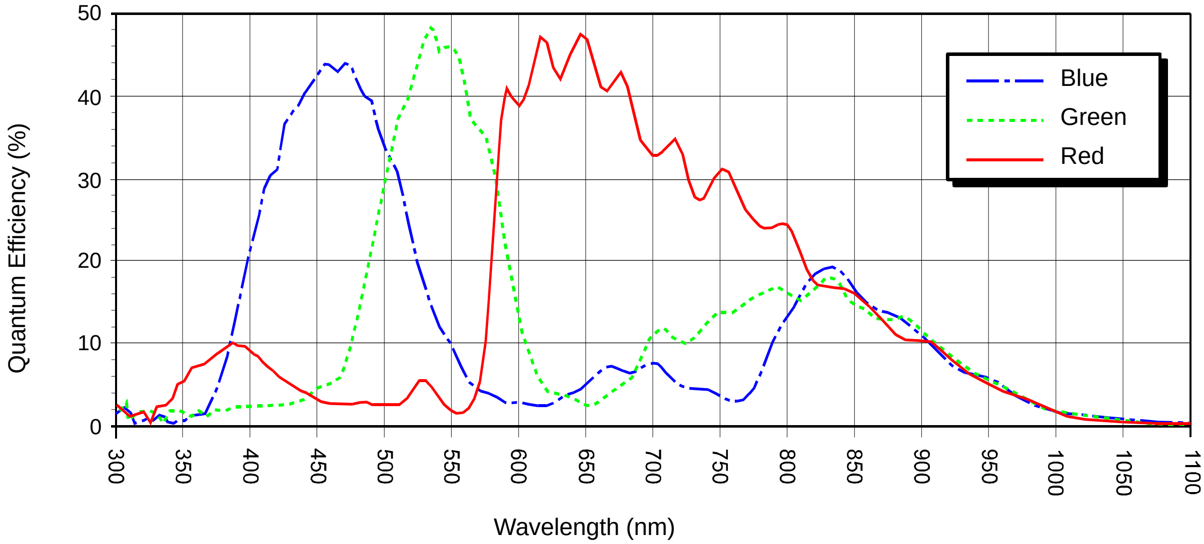


Fig. 18: acA2000-50gc, acA2040-25gc Spectral Response (From Sensor Data Sheet)

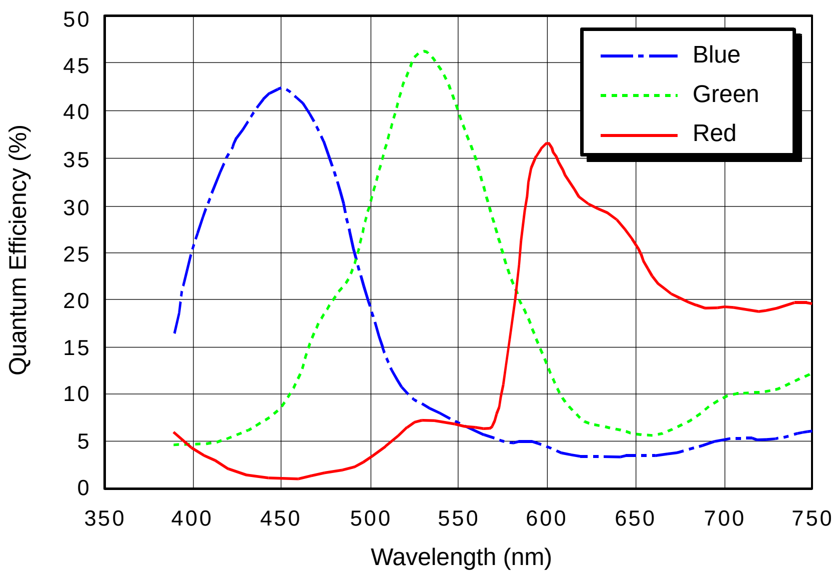


Fig. 19: acA2500-14gc Spectral Response (From Sensor Data Sheet)

1.4 Mechanical Specifications

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

1.4.1 Camera Dimensions and Mounting Points

The dimensions in millimeters for cameras equipped with a C-mount lens adapter are as shown in Figure 20. The dimensions for cameras equipped with a CS-mount lens adapter are as shown in Figure 21.

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

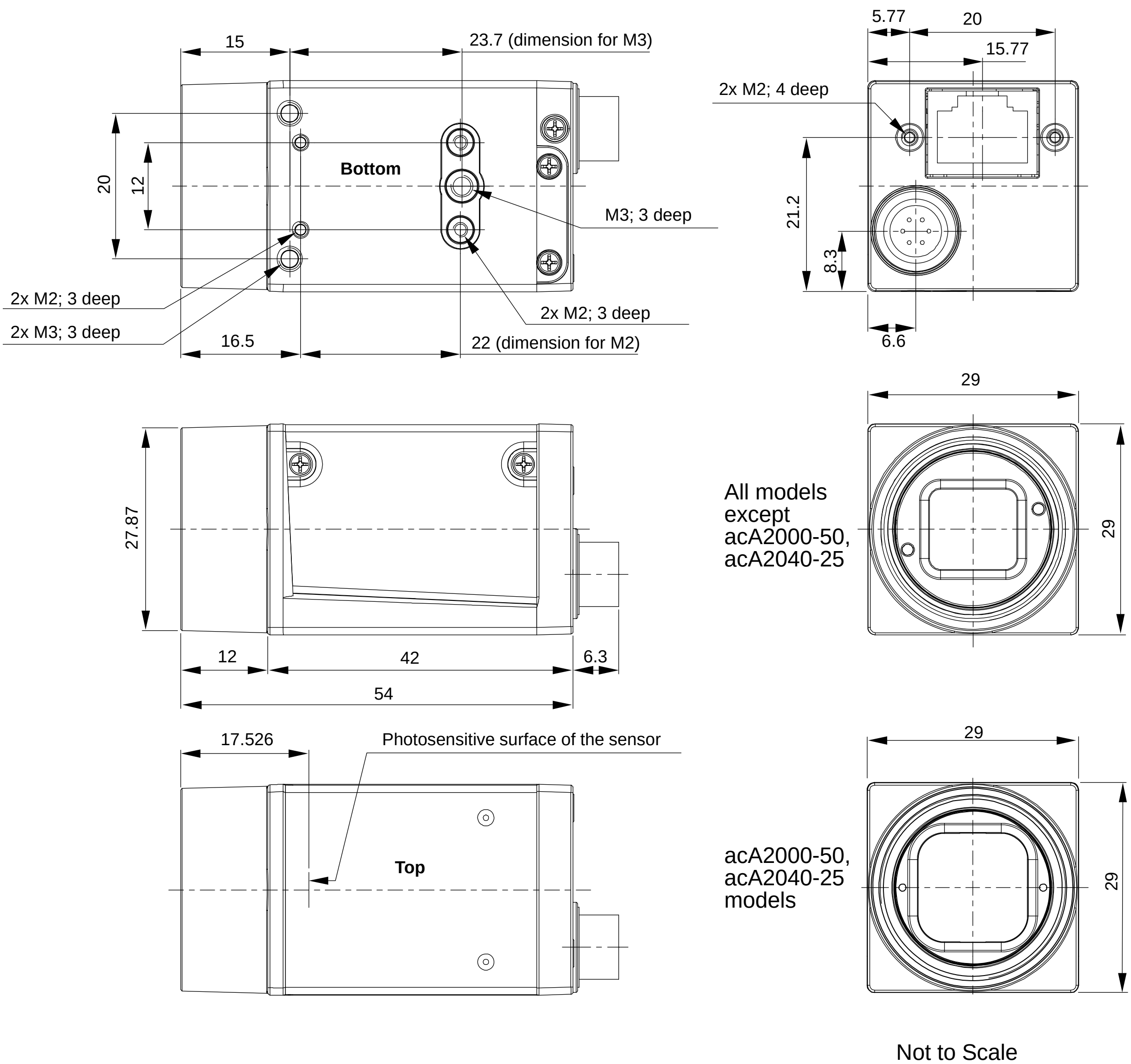
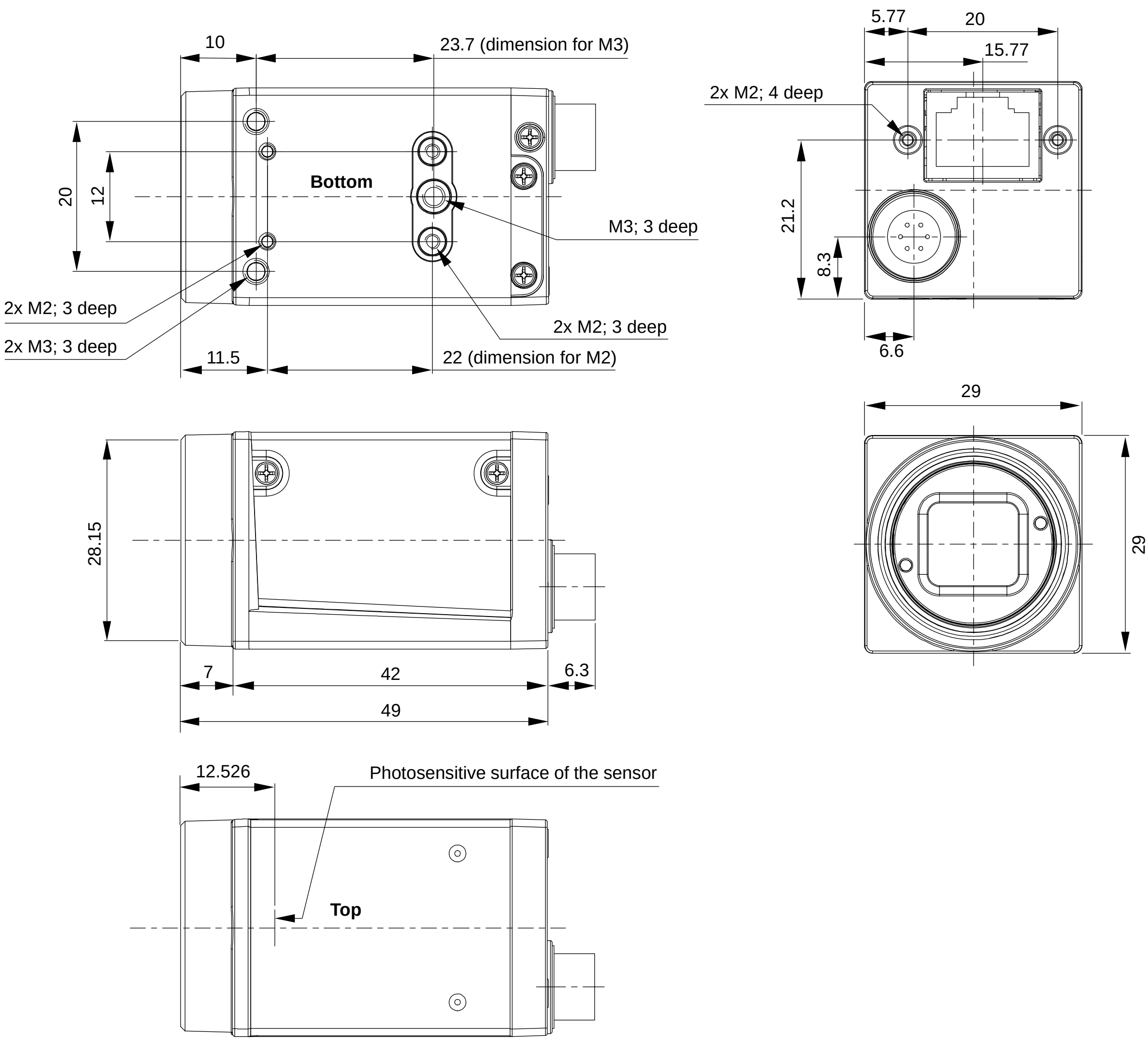


Fig. 20: Mechanical Dimensions (in mm) for Cameras with the C-mount Lens Adapter



Not to Scale

Fig. 21: Mechanical Dimensions (in mm) for Cameras with the CS-mount Lens Adapter

1.4.2 Maximum Allowed Lens Thread Length

The C-mount lens mount and the CS-mount lens mount on all cameras is normally equipped with a plastic filter holder. The length of the threads on any lens you use with the cameras depends on the lens adapter type you use with the camera:

- Camera with C-mount lens adapter (see Figure 22):

  The thread length can be a maximum of 9.6 mm, and the lens can intrude into the camera body a maximum of 10.8 mm.

- Camera with CS-mount lens adapter (see Figure 23):

  The thread length can be a maximum of 4.6 mm, and the lens can intrude into the camera body a maximum of 5.8 mm.

NOTICE

If either of these limits is exceeded, the lens mount or the filter holder will be damaged or destroyed and the camera will no longer operate properly.

Note that on color cameras, the filter holder will be populated with an IR cut filter. On monochrome cameras, the filter holder will be present, but will not be populated with an IR cut filter.

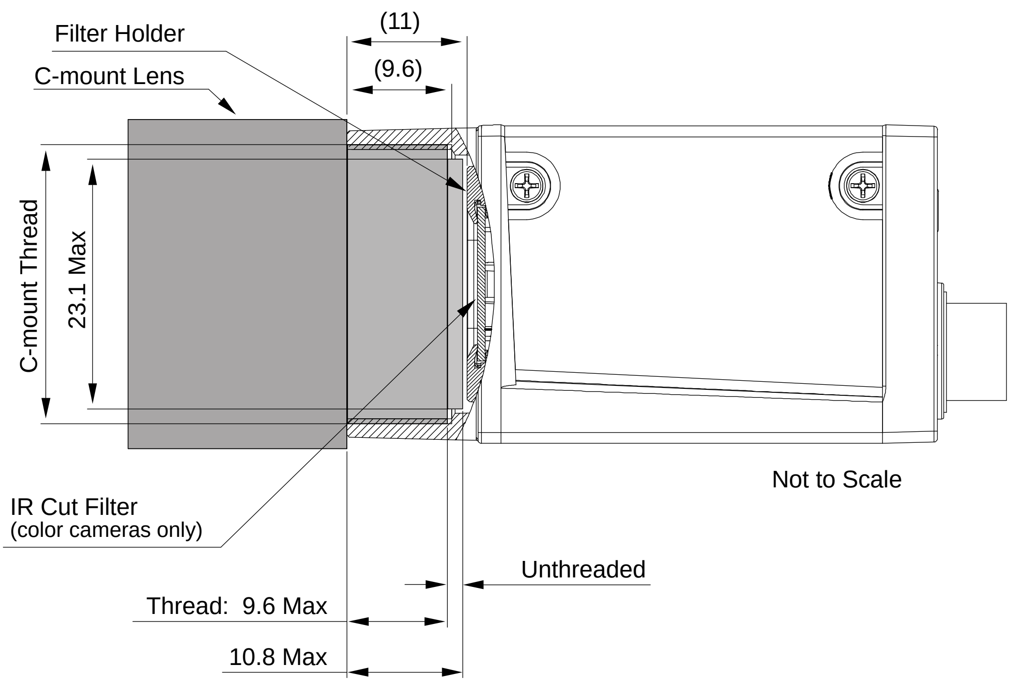


Fig. 22: Maximum Lens Thread Length (Dimensions in mm) for Cameras with the C-mount Lens Adapter

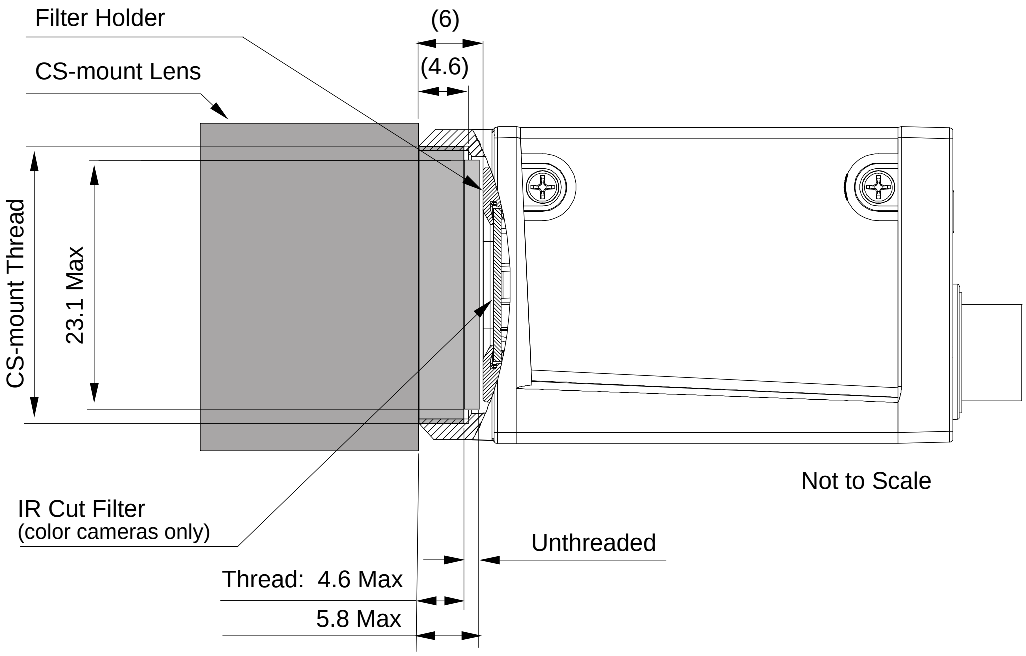


Fig. 23: Maximum Lens Thread Length (Dimensions in mm) for Cameras with the CS-mount Lens Adapter

1.4.3 Mechanical Stress Test Results

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. 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 7: 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.

1.5 Software Licensing Information

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

1.6 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.

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

1.7 Environmental Requirements

1.7.1 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

1.7.2 Heat Dissipation

You must provide sufficient heat dissipation to maintain the temperature of the camera housing at 50 °C or less. Since each installation is unique, Basler does not supply a strictly required technique for proper heat dissipation. Instead, we provide the following general guidelines:

- In all cases, you should monitor the temperature of the camera housing and make sure that the temperature does not exceed 50 °C. Keep in mind that the camera will gradually become warmer during the first hour of operation. After one hour, the housing temperature should stabilize and no longer increase.
- If your camera is mounted on a substantial metal component in your system, this may provide sufficient heat dissipation.
- The use of a fan to provide air flow over the camera is an extremely efficient method of heat dissipation. The use of a fan provides the best heat dissipation.

1.8 Precautions

NOTICE
Avoid dust on the sensor. The camera is shipped with a protective plastic seal 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 protective seal in place when there is no lens mounted on the camera.

NOTICE
On all cameras, the lens thread length is limited. All cameras (mono and color) are equipped with a plastic filter holder located in the lens mount. The location of the filter holder 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 filter holder or the lens mount will be damaged or destroyed and the camera will no longer operate properly.

For more specific information about the lens thread length, see Section 1.4.2 on page 27.

NOTICE
Voltage outside of the specified range can cause damage. 1. If you are supplying camera power via Power over Ethernet (PoE), the power must comply with the IEEE 802.3af specification. 2. If you are supplying camera power via the camera’s 6-pin connector and the voltage of the power is greater than +13.2 VDC, damage to the camera can result. If the voltage is less than +11.3 VDC, the camera may operate erratically.

NOTICE
An incorrect plug can damage the 6-pin connector. The plug on the cable that you attach to the camera’s 6-pin connector must have 6 female pins. Using a plug designed for a smaller or a larger number of pins can damage the connector.

NOTICE

Inappropriate code may cause unexpected camera behavior.

1. The code snippets provided in this manual are included as sample code only. Inappropriate code may cause your camera to function differently than expected and may compromise your application.
2. 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.
3. 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 package.

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 do a quick, simple installation of the camera is included in the ace Quick Installation Guide for GigE Cameras (AW000897xx000). You can download the Quick Installation Guide from the Downloads section of our website:
www.baslerweb.com

More extensive information about how to perform complicated installations is included in the Installation and Setup Guide for Cameras Used with Basler’s pylon API (AW000611xx000).

You can download the Installation and Setup Guide for Cameras Used with Basler’s pylon API from the Downloads section of our website:
www.baslerweb.com

The install and setup guide includes extensive information about how to install both hardware and software and how 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.

3 Camera Drivers and Tools for Changing Camera Parameters

This chapter provides an overview of the camera drivers and the options available for changing the camera's parameters.

The options available with the Basler pylon Driver Package let you change parameters and control the camera by using a stand-alone GUI (known as the pylon Viewer) or by accessing the camera from within your software application using the driver API.

3.1 The pylon Driver Package

The Basler pylon Driver Package is designed to operate all Basler cameras that have an IEEE 1394a interface, an IEEE 1394b interface, or a GigE interface. It will also operate some newer Basler camera models with a Camera Link interface. The pylon drivers offer reliable, real-time image data transport into the memory of your PC at a very low CPU load.

Features in the pylon driver package include:

- The Basler GigE Vision filter driver
- The Basler GigE Vision performance driver
- IEEE 1394a/b drivers
- A Camera Link configuration driver for some newer camera models
- A pylon camera API for use with a variety of programming languages
- A pylon DirectShow driver
- A pylon TWAIN driver
- A variety of adapters for third party software imaging processing libraries
- The Basler pylon Viewer and the Basler pylon IP Configuration Tool
- Source code samples
- A programming guide and API reference.

You can obtain the Basler pylon Driver Package from the Downloads section of our website: www.baslerweb.com

To help you install the drivers, you can also download the *Installation and Setup Guide for Cameras Used with Basler's pylon API* (AW000611xx000) from the website.

The pylon package includes several tools that you can use to change the parameters on your camera including the pylon Viewer, the pylon IP Configuration Tool, and the pylon API. The remaining sections in this chapter provide an introduction to the tools.

3.1.1 The pylon Viewer

The pylon Viewer is included in Basler's pylon Driver Package. The 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 Basler's pylon API* (AW000611xx000).

3.1.2 The pylon IP Configuration Tool

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

For more information about using the IP Configuration Tool, see the *Installation and Setup Guide for Cameras Used with Basler's pylon API* (AW000611xx000).

3.1.3 The pylon API

After the pylon Driver Package has been installed on your PC, you can access all of the camera's parameters and can control the camera's full functionality from within your application software by using the pylon API. The pylon Programmer's Guide and the pylon API Reference contain an introduction to the API and include information about all of the methods and objects included in the API. The programmer's guide and API reference are included in the pylon SDK.

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

4 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.

4.1 Overview (All Models Except acA2000-50, acA2040-25, acA2500-14)

The camera provides features such as a global 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 frame 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 24 on page 42 for cameras with a progressive scan sensor and Figure 25 on page 42 for cameras with an interlaced scan sensor). The charges from the bottom row 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 row-wise 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.

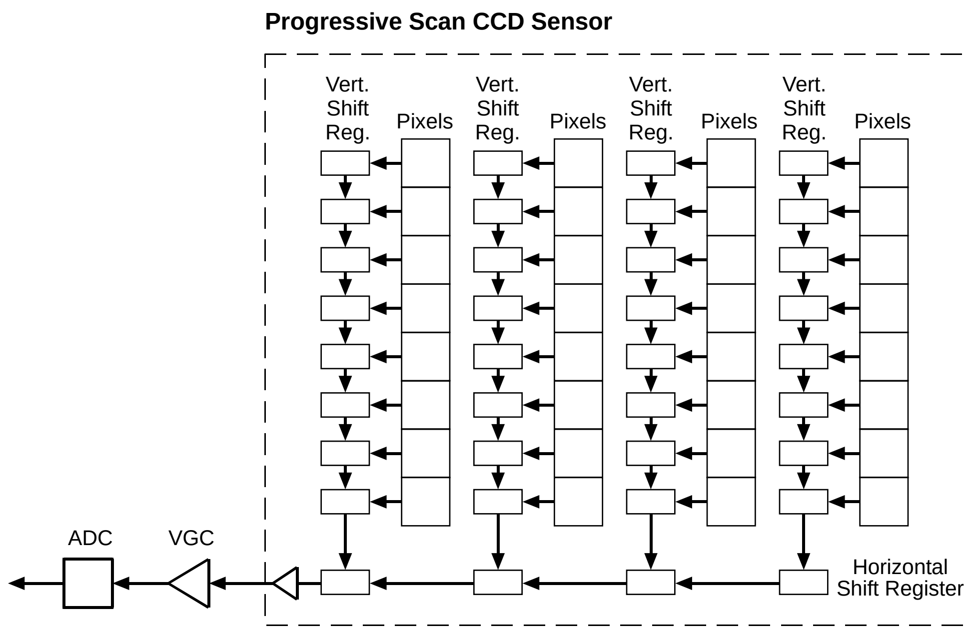


Fig. 24: CCD Sensor Architecture - Progressive Scan Sensors

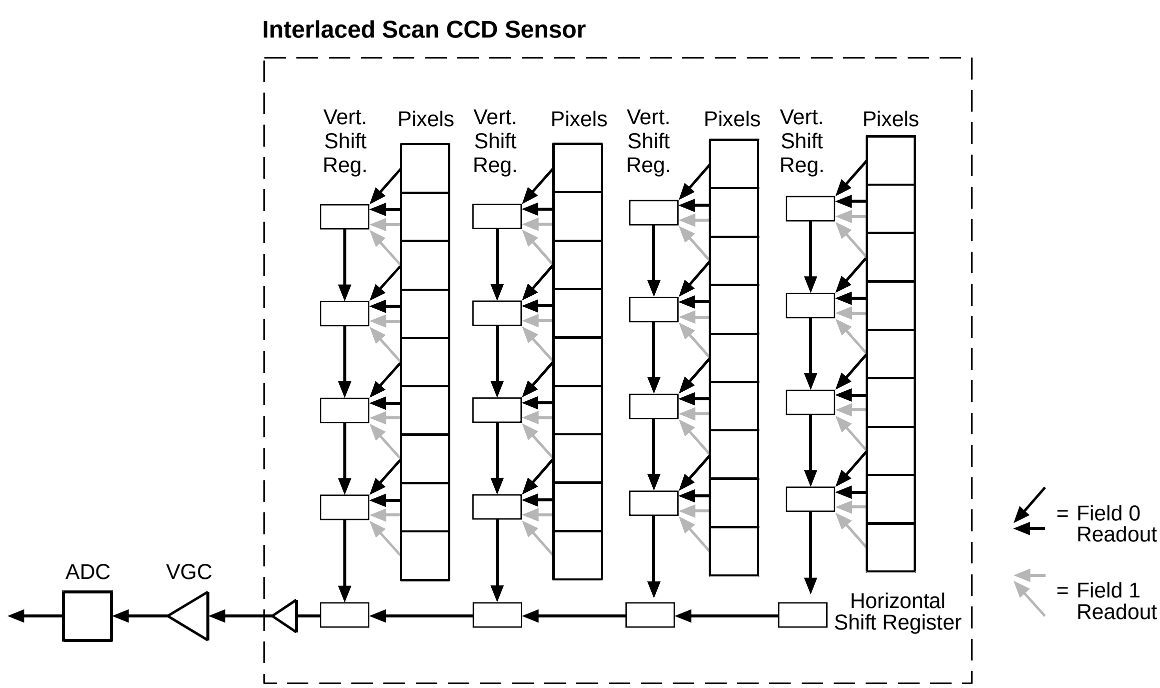


Fig. 25: CCD Sensor Architecture - Interlaced Scan Sensors

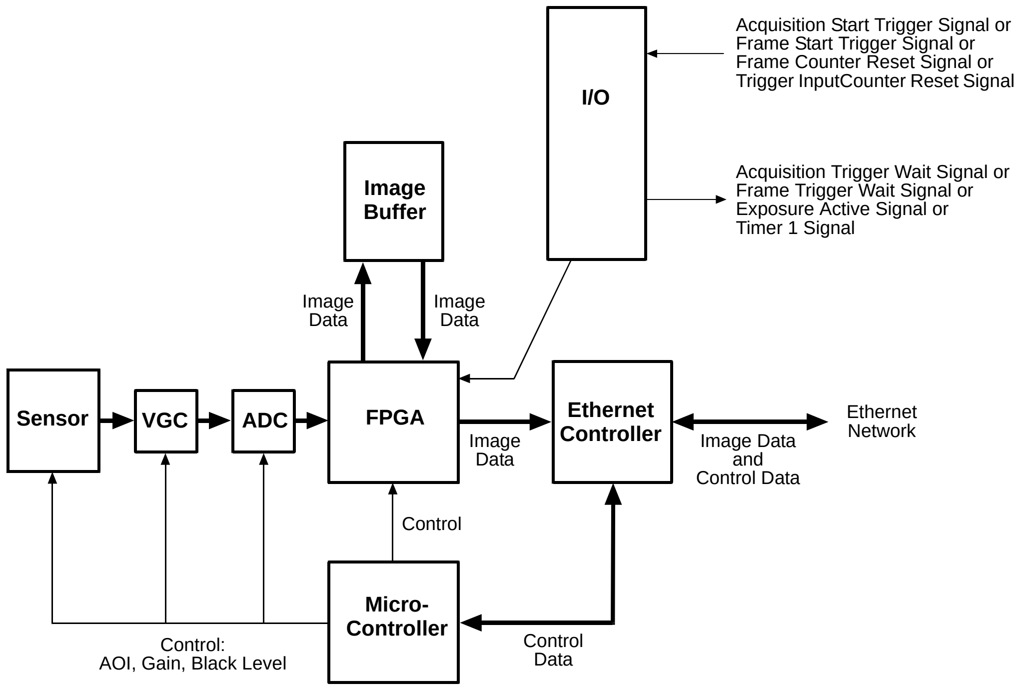


Fig. 26: Camera Block Diagram

4.2 Overview (acA2000-50, acA2040-25 Only)

The camera provides features such as a global 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. 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 each sensor row when exposure of the row ends. At readout, accumulated charges are transported from the row's light-sensitive elements (pixels) to the analog processing controls (see Figure 27 on page 44). As the charges move through the analog controls, they are converted to voltages proportional to the size of each charge. Each voltage is then amplified by a Variable Gain Control (VGC). Next the voltages are digitized by an Analog-to-Digital converter (ADC). After the voltages have been amplified and digitized, they are passed through the sensor's digital controls for additional signal processing. The digitized pixel data leaves the sensor, passes through an FPGA, and moves into an image buffer.

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.

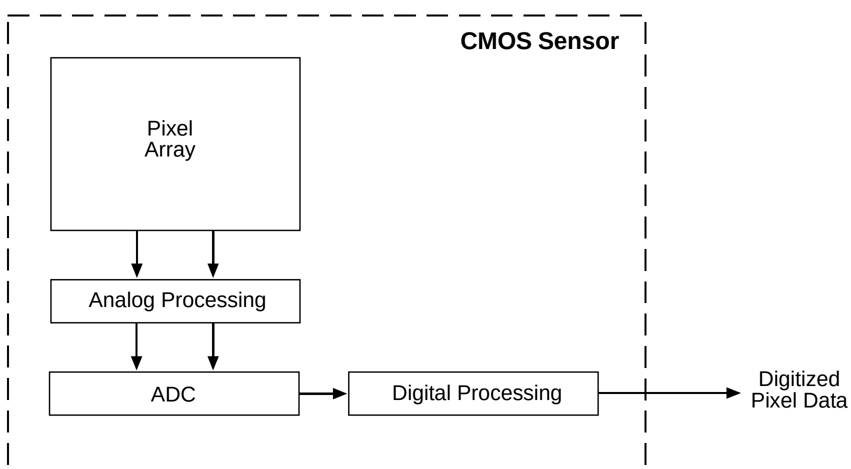


Fig. 27: CMOS Sensor Architecture

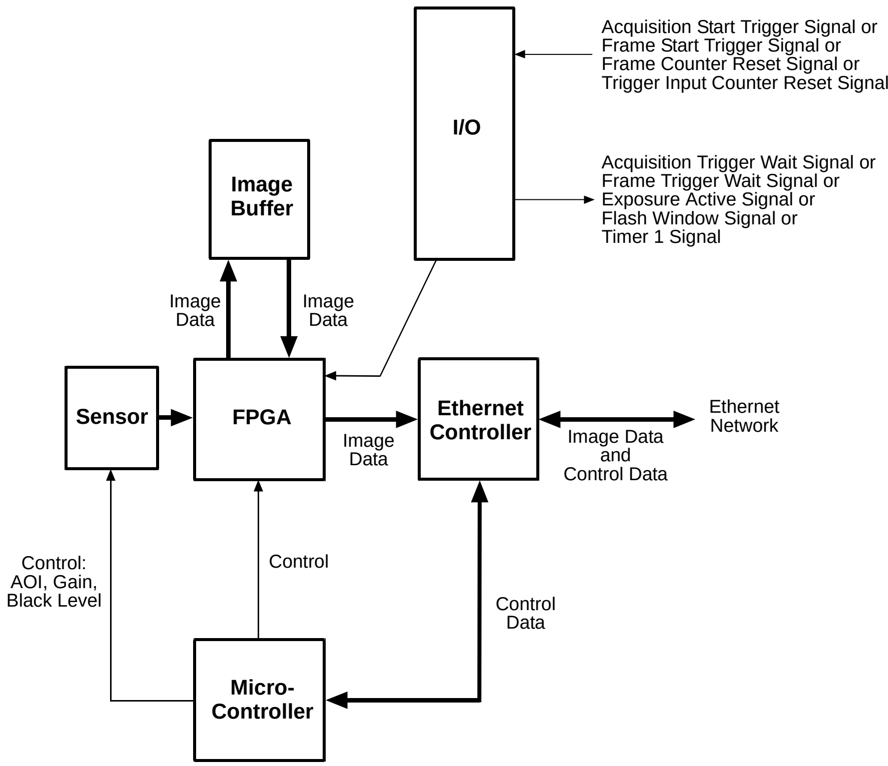


Fig. 28: Camera Block Diagram

4.3 Overview (acA2500-14 Only)

The camera provides features such as an electronic rolling 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 frame acquisition start.

Because the camera has a rolling shutter, the exposure start signal will only start exposure of the first row of pixels in the sensor. Exposure of each subsequent row will then automatically begin with an increasing temporal shift for each row. The exposure time will be equal for each row.

Accumulated charges are read out of each sensor when exposure ends. At readout, accumulated charges are transported from the row's light-sensitive elements (pixels) to the analog processing controls (see Figure 29 on page 47). As the charges move through the analog controls, they are converted to voltages proportional to the size of each charge. Each voltage is then amplified by a Variable Gain Control (VGC). Next the voltages are digitized by an Analog-to-Digital converter (ADC). After the voltages have been amplified and digitized, they are passed through the sensor's digital controls for additional signal processing. The digitized pixel data leaves the sensor, passes through an FPGA, and moves into an image buffer.

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.

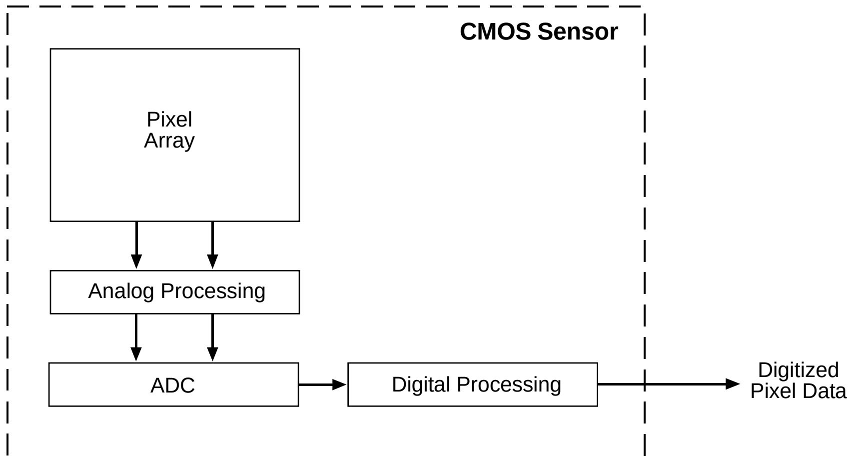


Fig. 29: CMOS Sensor Architecture

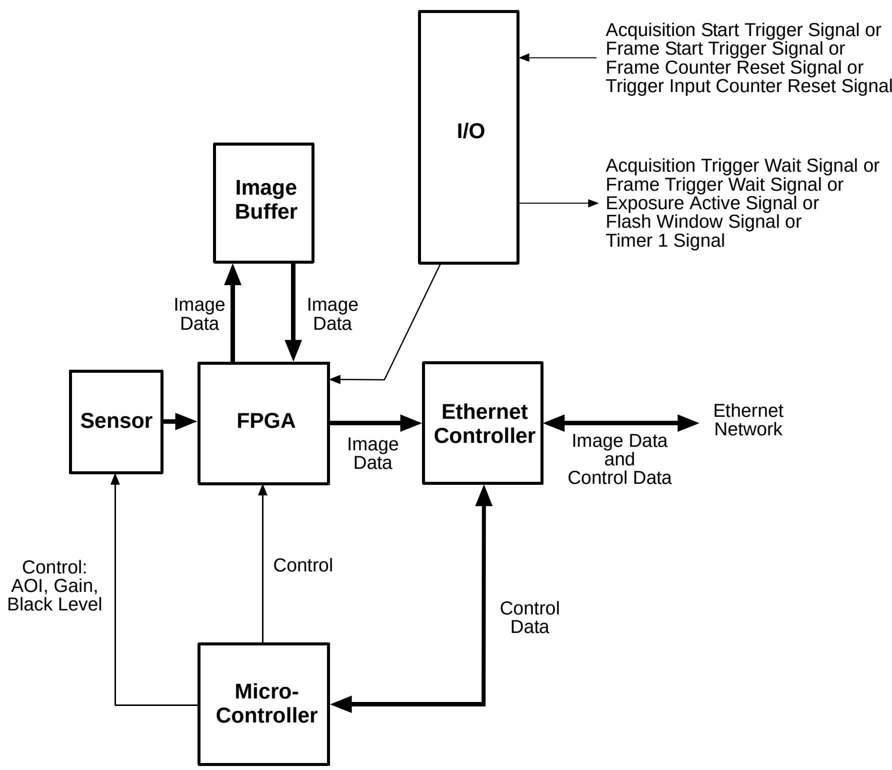


Fig. 30: Camera Block Diagram

5 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.

5.1 General Description of the Camera Connections

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. Since the camera is Power over Ethernet capable, the jack can also be used to provide power to the camera.
- A 6-pin receptacle used to provide access to the camera's I/O lines and to provide power to the camera (if PoE is not used).

Figure 31 shows the location of the two connectors.

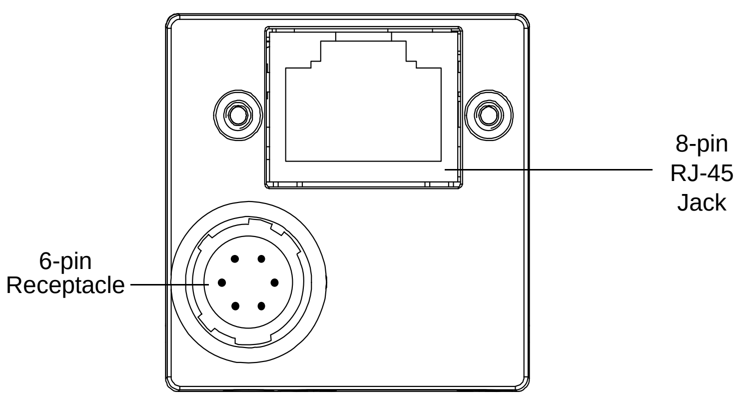


Fig. 31: Camera Connectors

5.2 Camera Connector Pin Assignments and Numbering

5.2.1 6-pin Connector Pin Assignments and Numbering

The 6-pin connector is used to access the physical input line and physical output line on the camera. It is also used to supply power to the camera (if PoE is not used).

The pin assignments for the connector are shown in Table 8.

Pin	Designation
1	+12 VDC Camera Power
2	I/O Input 1
3	Not Connected
4	I/O Out 1
5	I/O Ground
6	DC Camera Power Ground

Table 8: Pin Assignments for the 6-pin Connector

The pin numbering for the 6-pin connector is as shown in Figure 32.

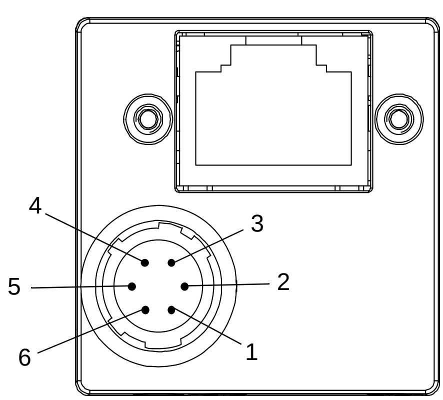


Fig. 32: Pin Numbering for the 6-pin Connector

5.2.2 8-pin RJ-45 Jack Pin Assignments and Numbering

The 8-pin RJ-45 jack provides a Gigabit Ethernet connection to the camera.

The jack can also be used to provide Power over Ethernet (IEEE 802.3af compliant) to the camera.

Pin assignments and pin numbering adhere to the Ethernet standard and IEEE 802.3af.

5.3 Camera Connector Types

5.3.1 6-pin Connector

The 6-pin connector on the camera is a Hirose micro receptacle (part number HR10A-7R-6PB) or the equivalent.

The recommended mating connector is the Hirose micro plug (part number HR10A-7P-6S) or the equivalent.

5.3.2 8-pin RJ-45 Jack

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

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

Suitable cable assemblies are also available from, for example, Components Express Inc. and from the Intercon 1 division of Nortech Systems, Inc.

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

5.4 Camera Cabling Requirements

5.4.1 Ethernet Cables

Use high-quality Ethernet cables. To avoid EMI, the cables must be shielded. Use of category 6 or category 7 cables with S/STP shielding is 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.

5.4.2 Standard Power and I/O Cable

The standard power and I/O cable is intended for use, if the camera is not connected to a PLC device. If the camera is connected to a PLC device, we recommend using a PLC power and I/O cable rather than the standard power and I/O cable.

If power for the I/O input is supplied at 24 VDC, you can use a PLC power and I/O cable when the camera is not connected to a PLC device.

See the following section for more information about PLC power and I/O cables.

A single "standard power and I/O cable" is used to supply power to the camera and to connect to the camera’s I/O lines as shown in Figure 33. If you are supplying power to the camera via Power over Ethernet, the cable still can be used to connect to the I/O lines.

We recommend that you supply power to the camera **either** via the camera’s RJ-45 jack **or** via the camera’s 6-pin connector.

The end of the standard power and I/O cable that connects to the camera must be terminated with a Hirose micro plug (part number HR10A-7P-6S) or the equivalent. The cable must be wired to conform with the pin assignments shown in the pin assignment table.

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

Close proximity to strong magnetic fields should be avoided.

The required 6-pin Hirose plug is available from Basler. Basler also offers a cable assembly that is terminated with a 6-pin Hirose plug on one end and unterminated on the other. Contact your Basler sales representative to order connectors or cables.

NOTICE
An incorrect plug can damage the 6-pin connector. The plug on the cable that you attach to the camera's 6-pin connector must have 6 female pins. Using a plug designed for a smaller or a larger number of pins can damage the connector.

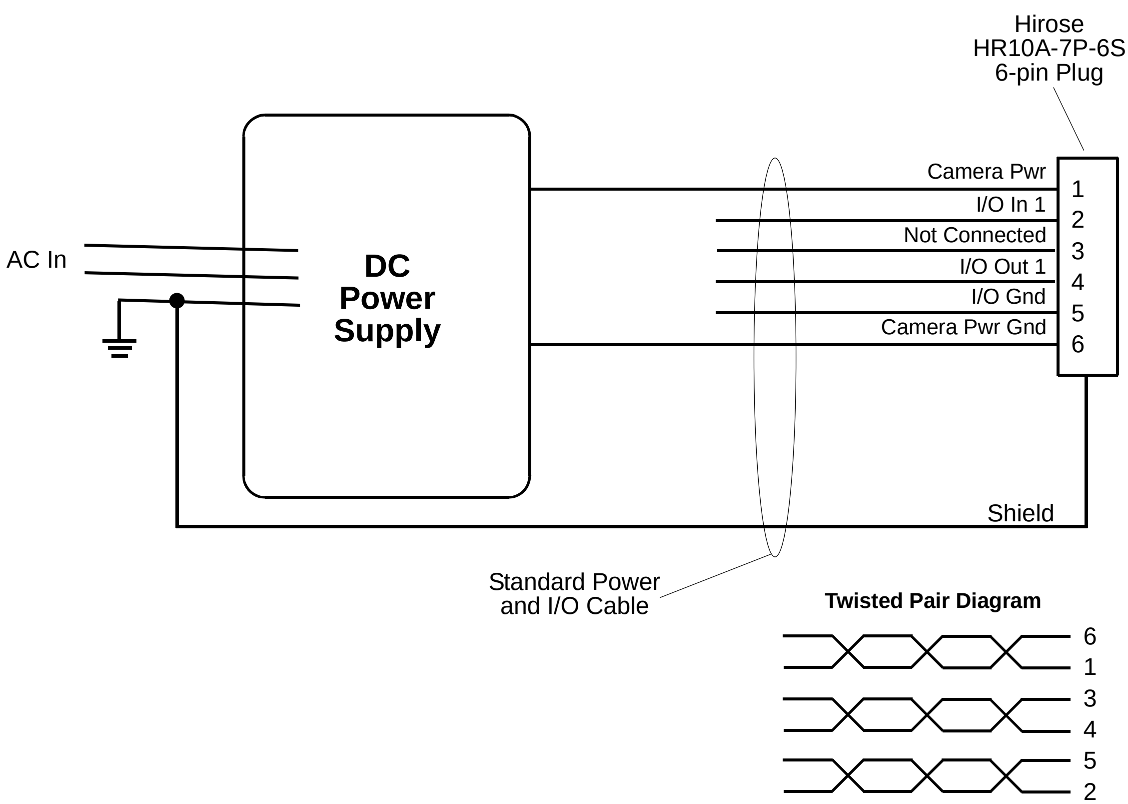


Fig. 33: Standard Power and I/O Cable

5.4.3 PLC Power and I/O Cable

We recommend using a PLC power and I/O cable, if the camera is connected to a PLC device.

If power for the I/O input is supplied at 24 VDC, you can use a PLC power and I/O cable when the camera is not connected to a PLC device.

As with the standard power and I/O cable described in the previous section, the PLC power and I/O cable is a single cable that both connects power to the camera and connects to the camera's I/O lines. 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.

We recommend that you supply power to the camera **either** via the camera's RJ-45 jack **or** via the camera's 6-pin connector.

Close proximity to strong magnetic fields should be avoided.

Basler offers a PLC power and I/O cable that is terminated with a 6-pin Hirose plug (HR10A-7P-6S) on the end that connects to the camera. The other end is unterminated. Contact your Basler sales representative to order the cable.

For information about the applicable voltage levels, see Section 5.7.1 on page 57.

5.5 Camera Power

Power can be supplied to the camera in either of two different ways:

- via Power over Ethernet (PoE), i.e., via the Ethernet cable plugged into the camera's RJ-45 connector.
- from a power supply via a power and I/O cable (either a standard cable or a PLC cable) plugged into the camera's 6-pin connector.

ⓘ	We recommend that you supply power to the camera **either** via the camera's RJ-45 jack **or** via the camera's 6-pin connector.

Via PoE

If you are supplying power via PoE, the power provided must adhere to the requirements specified in IEEE 802.3af.

Power consumption is as shown in the specification tables in Section 1 of this manual.

From a Power Supply to the 6-Pin Connector

Camera power can be provided from a power supply to the camera's 6-pin connector via a standard power and I/O cable or via a PLC power and I/O cable.

Nominal operating voltage is +12 VDC (± 10%) with less than one percent ripple. Power consumption is as shown in the specification tables in Section 1 of this manual.

Close proximity to strong magnetic fields should be avoided.

NOTICE
Voltage outside of the specified range can cause damage. If the voltage of the power to the camera is greater than +13.2 VDC, damage to the camera can result. If the voltage is less than +11.3 VDC, the camera may operate erratically.

NOTICE
Voltage outside of the specified range can cause damage. Note that the recommended voltage range for camera power (see above) differs from the recommended voltage ranges for the input and output lines (see Section 5.7.1 on page 57 and Section 5.8.1 on page 62). Note also that the recommended voltage range for camera power for Basler ace GigE cameras can differ from the recommended voltage range for camera power for other Basler cameras.

NOTICE
An incorrect plug can damage the 6-pin connector. The plug on the cable that you attach to the camera's 6-pin connector must have 6 female pins. Using a plug designed for a smaller or a larger number of pins can damage the connector.

For more information about the 6-pin connector and the power and I/O cables see Section 5.2 on page 50, Section 5.3 on page 51, and Section 5.4 on page 52.

5.6 Ethernet GigE Device Information

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

5.7 Input Line Description

5.7.1 Voltage Requirements

NOTICE

Voltage outside of the specified range can cause damage.

The recommended voltage range

- for the input line differs from the recommended voltage ranges for camera power (see Section 5.5 on page 55) and for the output line (see Section 5.8.1 on page 62).
- for the I/O input line of Basler ace GigE cameras can differ from the recommended voltage ranges for the I/O input lines of other Basler cameras.

You must supply power within the specified voltage range.

Different voltage levels apply, depending on whether the standard power and I/O cable or a PLC power and I/O cable is used (see below).

Voltage Levels When the Standard Power and I/O Cable is Used

The following voltage requirements apply to the camera's I/O input line (pin 2 of the 6-pin connector) when a standard power and I/O cable is used:

Voltage	Significance
+0 to +24 VDC	Recommended I/O input voltage.
+0 to +1.4 VDC	The voltage indicates a logical 0.
> +1.4 to +2.2 VDC	Region where the transition threshold occurs; the logical state is not defined in this region.
> +2.2 VDC	The voltage indicates a logical 1.
+30.0 VDC	Absolute maximum; the camera may be damaged when the absolute maximum is exceeded.

Table 9: Voltage Requirements When Using the Standard Power and I/O Cable

Voltage Levels When a PLC Power and I/O Cable is Used

The following requirements apply to the camera's I/O input (pin 2 of the 6-pin connector) when a PLC power and I/O cable is used. The PLC power and I/O cable will adjust the voltages to the levels required by the camera's I/O input (see Table 9).

Voltage	Significance
+0 to +24 VDC	Recommended I/O input voltage.
+0 to +8.4 VDC	The voltage indicates a logical 0.
> +8.4 to +10.4 VDC	Region where the transition threshold occurs; the logical state is not defined in this region.
> +10.4 VDC	The voltage indicates a logical 1.
+30.0 VDC	Absolute maximum; the camera may be damaged when the absolute maximum is exceeded.

Table 10: Voltage Requirements When Using a PLC Power and I/O Cable

5.7.2 Characteristics

The camera is equipped with one physical input line designated as Input Line 1. The input line is accessed via the 6-pin receptacle on the back of the camera.

As shown in Figure 34, the input line is opto-isolated. See the previous section for input voltages and their significances. The absolute maximum input voltage is +30.0 VDC. The current draw for each input line is between 5 mA and 15 mA.

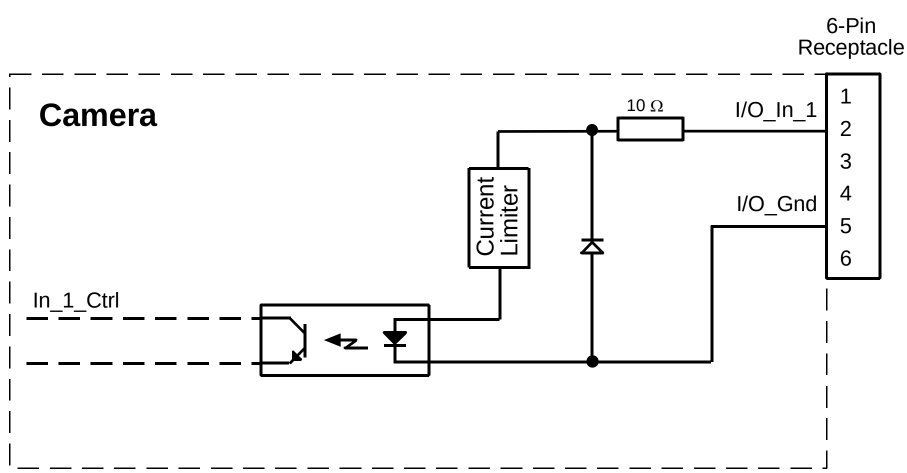


Fig. 34: Input Line Schematic (Simplified)

Figure 35 shows an example of a typical circuit you can use to input a signal into the camera.

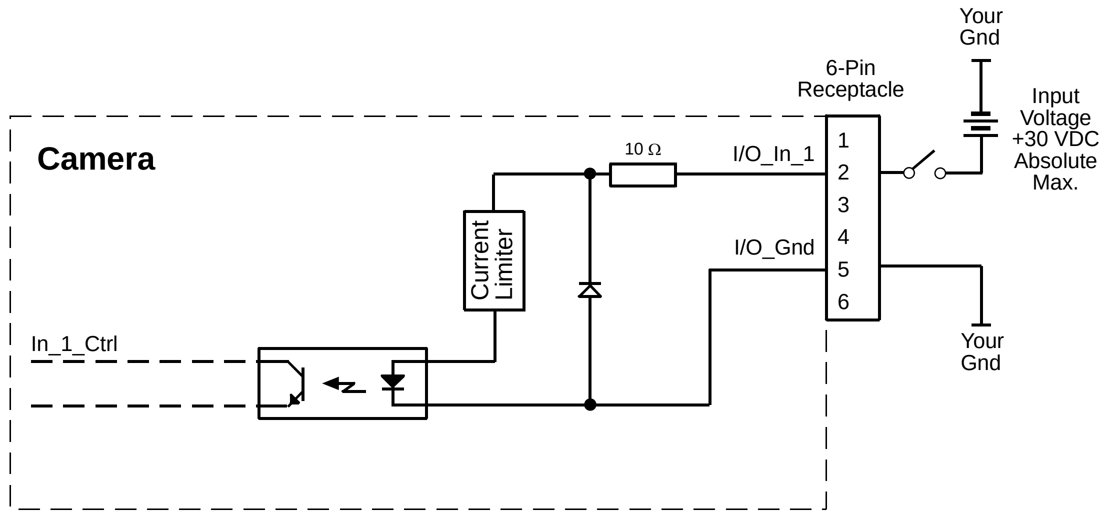


Fig. 35: Typical Input Circuit (Simplified)

For more information about input line pin assignments and pin numbering, see Section 5.2 on page 50.

For more information about how to use an externally generated frame start trigger (ExFSTrig) signal to control acquisition start, see Section 7.4.3 on page 98.

For more information about configuring the input line, see Section 6.1 on page 67.

5.7.3 Input Line Response Time

The response times for the input line on the camera are as shown in Figure 36.

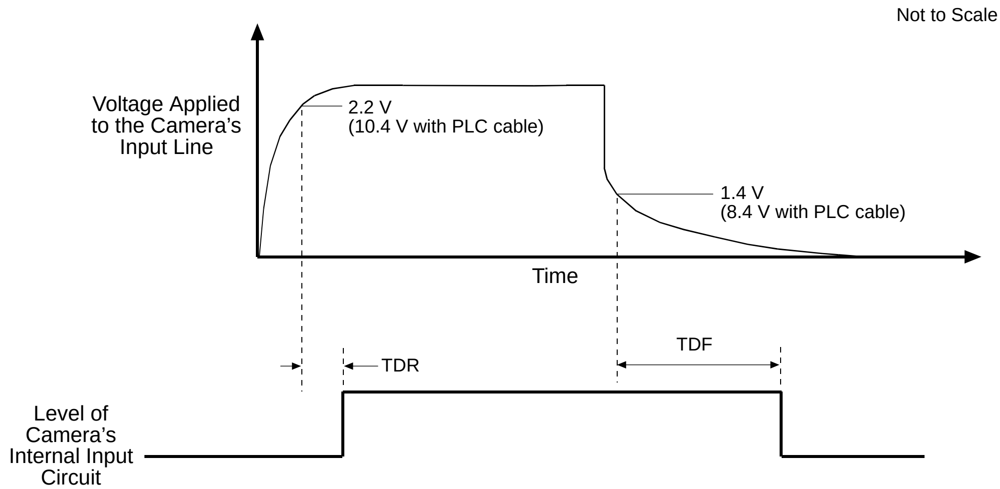


Fig. 36: Input Line Response Times

Time Delay Rise (TDR) = 1.3 µs to 1.6 µs

Time Delay Fall (TDF) = 40 µs to 60 µs

5.7.4 Selecting the Input Line as the Source Signal for a Camera Function

You can select input line 1 to act as the source signal for the following camera functions:

- the acquisition start trigger
- the frame start trigger
- the frame counter reset
- the trigger input counter reset

Note that when the input line has been selected as the source signal for a camera function, you must apply an electrical signal to the input line that is appropriately timed for the function.

For more information about selecting input line 1 as the source signal for a camera function, see Section 6.1 on page 67.

5.8 Output Line Description

5.8.1 Voltage Requirements

NOTICE

Voltage outside of the specified range can cause damage.

Note that the recommended voltage range for the output line differs from the recommended voltage ranges for camera power (see Section 5.5 on page 55) and for the input line (see Section 5.7.1 on page 57). You must supply power within the specific voltage range.

Note also that the recommended voltage range for the I/O output line of Basler ace GigE cameras can differ from the recommended voltage ranges for the I/O output lines of other Basler cameras. You must supply power within the specified voltage range.

The following voltage requirements apply to the I/O output line (pin 4 of the 6-pin connector):

Voltage	Significance
< +3.3 VDC	The I/O output may operate erratically.
+3.3 to +24 VDC	Recommended I/O output voltage.
+30.0 VDC	Absolute maximum; the camera may be damaged, if the absolute maximum is exceeded.

Table 11: Voltage Requirements for the I/O Output

5.8.2 Characteristics

The camera is equipped with one physical output line designated as Output Line 1. The output line is accessed via the 6-pin connector on the back of the camera.

As shown in Figure 37, the output line is opto-isolated. See the previous section for the recommended operating voltages. The absolute maximum voltage is +30.0 VDC. The **maximum current** allowed through the output circuit is **50 mA**.

A low output signal from the camera results in a non-conducting Q1 transistor in the output circuit.

A high output signal from the camera results in a conducting Q1 transistor in the output circuit.

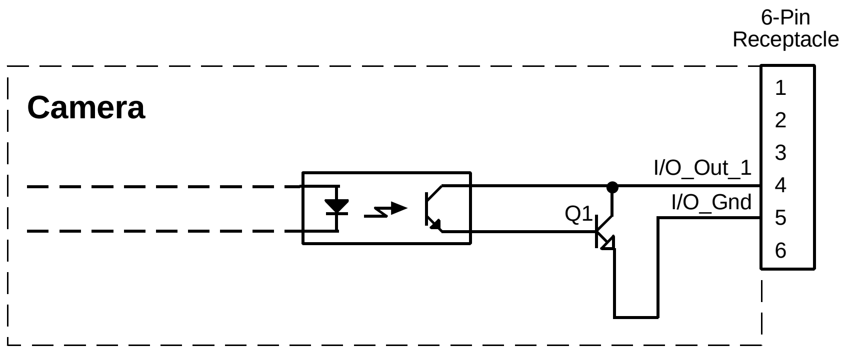


Fig. 37: Output Line Schematic (Simplified)

On early production cameras with firmware versions of V0.x-x, the logic for the output circuit was different. On these cameras:

- A low output signal from the camera on Out_1_Ctrl results in a conducting Q1 transistor.
- A high output signal from the camera results in a non-conducting Q1 transistor.

If you are using both older and newer cameras in your application, the difference in the behavior of the output may be a problem. One way that you can address the situation is to apply the invert function to the output on the older cameras. This will make the behavior of the output on the older cameras match the behavior on the newer cameras.

You could also choose to apply the invert function to the output on the newer cameras, and this would make the behavior of the newer cameras match the behavior of the older ones.

For more information about the invert function on the output, see Section 6.2.3 on page 72.

Figure 38 shows a typical circuit you can use to monitor the output line with a voltage signal.

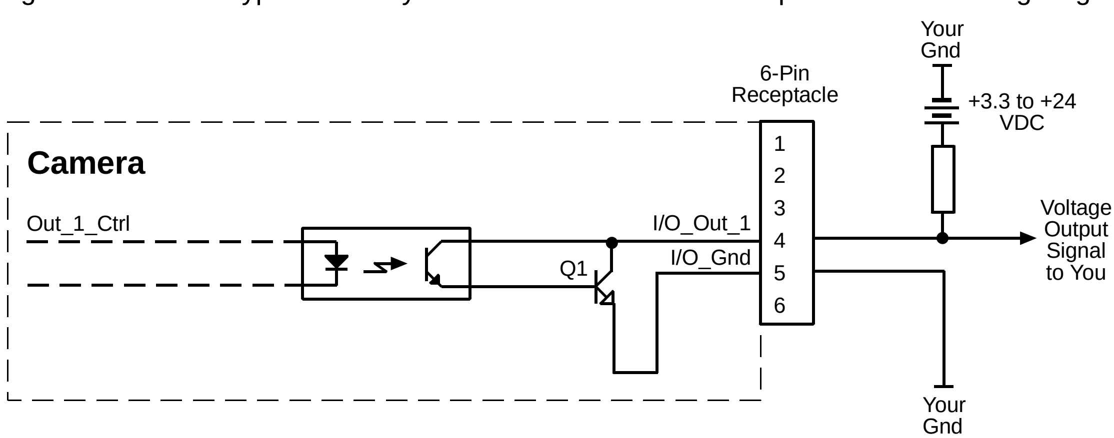


Fig. 38: Typical Voltage Output Circuit (Simplified Example)

Figure 39 shows a typical circuit you can use to monitor the output line with an LED or an opto-coupler. In this example, the voltage for the external circuit is +24 VDC. Current in the circuit is limited by an external resistor.

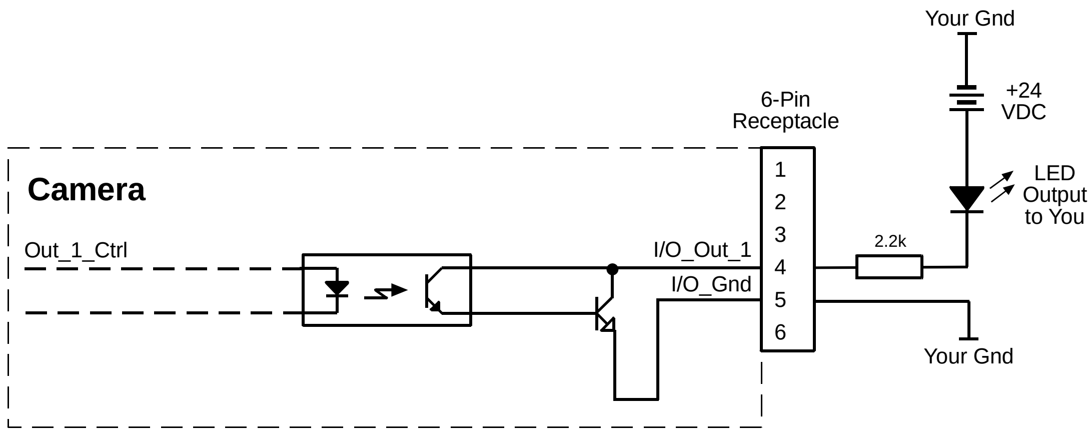


Fig. 39: Typical LED Output Signal at +24 VDC for the External Circuit (Simplified Example)

By default, the camera's Exposure Active signal is assigned to Output Line 1.

The assignment of a camera output signal to Output Line 1 can be changed by the user. For more information about assigning camera output signals to Output Line 1, see Section 6.2.1 on page 70.

For more information about output line pin assignments and pin numbering, see Section 5.2 on page 50.

For more information about the Exposure Active signal, see Section 7.10 on page 127.

5.8.3 Output Line Response Time

ⓘ	The information in this section assumes that the output circuit on your camera is designed as in the typical voltage output circuit shown in Section 5.8.2. The response time for the output line on your camera will typically fall into the ranges specified below. The exact response time for your specific application will depend on the external resistor and the applied voltage you use.

Response times for the output line on the camera are as shown in Figure 40.

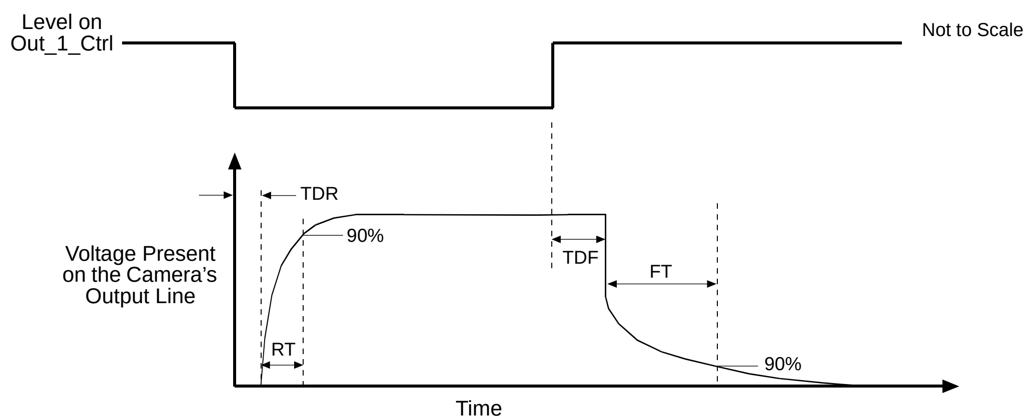


Fig. 40: Output Line Response Times

Time Delay Rise (TDR) = 40 µs

Rise Time (RT) = 20 µs to 70 µs

Time Delay Fall (TDF) = 0.6 µs

Fall Time (FT) = 0.7 µs to 1.4 µs

5.8.4 Selecting a Source Signal for the Output Line

To make the physical output line useful, you must select a source signal for the line. The camera has several standard output signals available and any one of them can be selected to act as the source signal for the output line.

For more information about selecting a source signal for the output line, see Section 6.2 on page 70.

6 I/O Control

This section describes how to configure the camera's physical input line and physical output line. It also provides information about monitoring the state of the input and output lines.

6.1 Configuring the Input Line

6.1.1 Selecting the Input Line as the Source Signal for a Camera Function

The camera is equipped with one physical input line designated as input line 1.

You can select the camera input line to act as the source signal for one of the following camera functions:

- Acquisition Start Trigger - If the input line is selected as the source signal for the acquisition start trigger, whenever a proper electrical signal is applied to the line, the camera will recognize the signal as an acquisition start trigger signal.
- Frame Start Trigger - If the input line is selected as the source signal for the frame start trigger, whenever a proper electrical signal is applied to the line, the camera will recognize the signal as a frame start trigger signal.
- Frame Counter Reset - If the input line is selected as the source signal for the frame counter reset, whenever a proper electrical signal is applied to the line, the counter value for the frame counter chunk feature will be reset.
- Trigger Input Counter Reset - If the input line is selected as the source signal for the trigger input counter reset, whenever a proper electrical signal is applied to the line, the counter value for the trigger reset counter chunk feature will be reset.

For detailed information about selecting input line 1 to act as the source signal for

- the acquisition start trigger and for details about how the acquisition start trigger operates, see Section 7.3 on page 85.
- the frame start trigger and for details about how the frame start trigger operates, see Section 7.4.3 on page 98.
- a frame counter reset and for details about how the frame counter chunk feature operates, see Section 11.3 on page 320.
- a trigger input counter reset and for details about how the trigger input counter chunk feature operates, see Section 11.3 on page 320.

For more information about the electrical characteristics of the input line, see Section 5.7 on page 57.

ⓘ	By default, input line 1 is selected as the source signal for the frame start trigger.

6.1.2 Input Line 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 41 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.

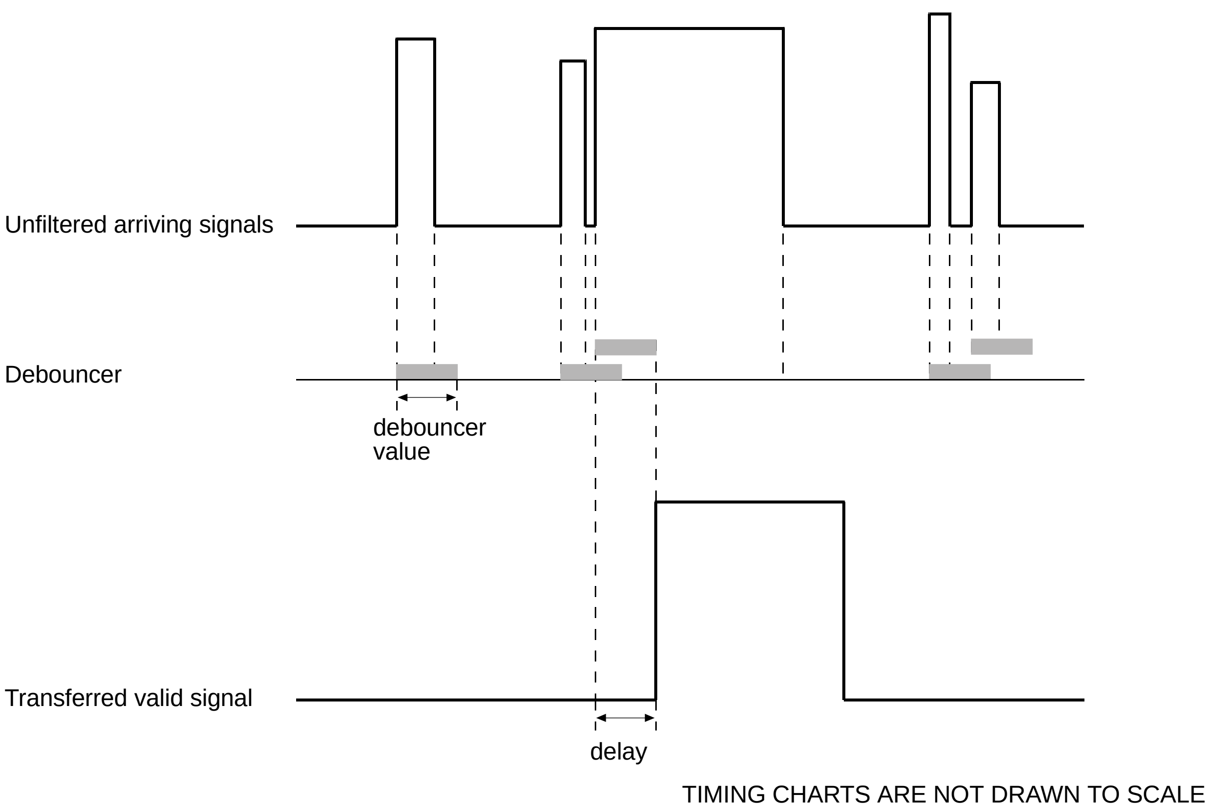


Fig. 41: Filtering of Input Signals by the Debouncer

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 20 µs.

To set the debouncer:

- Use the Line Selector to select input line1.
- 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 Basler 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 10 microseconds
Camera.LineDebouncerTimeAbs.SetValue( 10 );
```

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

For mor information about the pylon API and the pylon Viewer, see Section 3 on page 39.

6.1.3 Setting the Input Line for Invert

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

- Use the Line Selector to select the input line.
- Set the value of the Line Inverter parameter to true to enable inversion on the selected line or 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 );
```

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 3.1.1 on page 40.

6.2 Configuring the Output Line

6.2.1 Selecting a Source Signal for the Output Line

The camera is equipped with one physical output line designated as output line 1. You can select any one of the camera's standard output signals to act as the source signal for output line 1. The camera has five standard output signals available:

- Acquisition Trigger Wait
- Frame Trigger Wait
- Exposure Active
- Flash Window
- Timer Active

You can also designate the output line as "user settable". If the output line is designated as user settable, you can use the camera's API to set the state of the line as desired.

To select a camera output signal as the source signal for the output line or to designate the line as user settable:

- Use the Line Selector to select output line 1.
- 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 output line.

> By default, the Exposure Active signal is selected as the source signal for output line 1.

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

- the pylon API and the pylon Viewer, see Section 3.1.1 on page 40.
- the acquisition trigger and frame trigger wait signals, see Section 7.10.4 on page 132.
- the exposure active signal, see Section 7.10.1 on page 127.
- the flash window signal, see Section 7.7.2 on page 113 and Section 7.10.3 on page 131.
- working with a timer output signal, see Section 6.2.4 on page 73
- setting the state of a user settable output line, see Section 6.2.2 on page 71.
- the electrical characteristics of the output line, see Section 5.8 on page 62.

6.2.2 Setting the State of a User Settable Output Line

As mentioned in the previous section, you can designate the camera's output line as "user settable". If you have designated the output line as user settable, you can use camera parameters to set the state of the line.

Setting the State of a User Settable Output Line

To set the state of a user settable output line:

- Use the User Output Selector to select output line 1.
- Set the value of the User Output Value parameter to true (1) or false (0). This will set the state of the output line.

You can set the Output Selector and the User Output Value parameters from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to designate the output line as user settable and to set the state of the output line:

```
// Set output line 1 to user settable
Camera.LineSelector.SetValue( LineSelector_Out1 );
Camera.LineSource.SetValue( LineSource_UserOutput );
// Set the state of output line 1
Camera.UserOutputSelector.SetValue( UserOutputSelector_UserOutput1 );
Camera.UserOutputValue.SetValue( true );
bool currentUserOutput1State = Camera.UserOutputValue.GetValue( );
```

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

If you have the invert function enabled on the output line and the line is designated as user settable, the user setting sets the state of the line before the inverter.

For more information about the pylon API and the pylon Viewer, see Section 3.1.1 on page 40.

6.2.3 Setting the Output Line for Invert

You can set the output line to not invert or to invert.

When the output line is set to not invert:

- A logical zero on Out_1_Ctrl results in a non-conducting Q1 transistor in the output circuit (see Figure 42).
- A logical one on Out_1_Ctrl results in a conducting Q1 transistor in the output circuit.

When the output line is set to invert:

- A logical zero on Out_1_Ctrl results in a conducting Q1 transistor in the output circuit.
- A logical one on Out_1_Ctrl results in a non-conducting Q1 transistor in the output circuit.

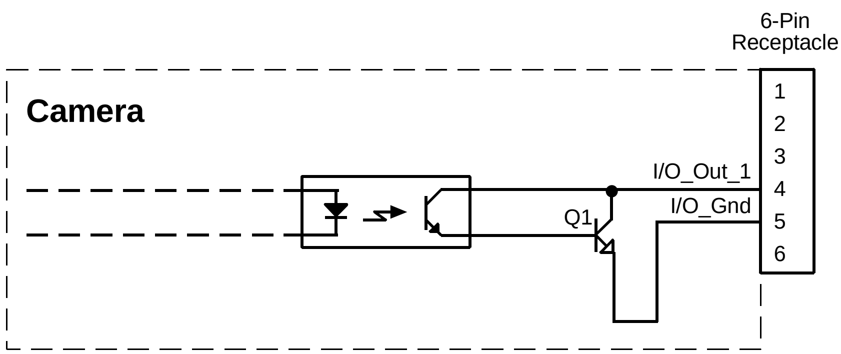


Fig. 42: Output Line Schematic (Simplified)

To set the invert function on the output line:

- Use the Line Selector to select output line 1.
- Set the value of the Line Inverter parameter to true to enable inversion on the selected line or to false to disable inversion.

You can set the Line Selector and the Line Inverter 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 value:

```
// Enable the inverter on output line 1
Camera.LineSelector.SetValue( LineSelector_Out1 );
Camera.LineInverter.SetValue( true );
```

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 3.1.1 on page 40.

6.2.4 Working with the Timer Output Signal

As mentioned in Section 6.2.1 on page 70, the source signal for the output line can be set to "timer active". The camera has one timer designated as "timer 1". When you set the source signal for the output line to "timer active", timer 1 will be used to supply the signal to the output line.

Timer 1 operates 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.

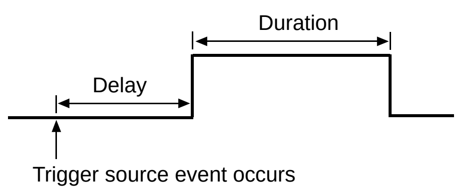


Fig. 43: Timer Signal

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 the timer.

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.

The timer signal can serve as the source signal for output line 1 on the camera. For information about selecting the timer 1 output signal as the source signal for output line 1, see Section 6.2.1 on page 70.

6.2.4.1 Setting the Trigger Source for the Timer

To set the trigger source for the timer:

- Use the Timer Selector to select timer 1.
- 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 Basler 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 );
```

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 3.1.1 on page 40.

6.2.4.2 Setting the Timer Delay Time

There are two ways to set the delay time for timer 1: 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 Time with Raw Values

When the delay time for timer 1 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 timer 1:

- Use the Timer Selector to select timer 1.
- 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 Basler 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 );
```

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.

You can set the Timer Delay Time Base 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 parameter value:

```
Camera.TimerDelayTimebaseAbs.SetValue( 5 );
```

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 3.1.1 on page 40.

Setting the Delay Time with an Absolute Value

You can also set the timer 1 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 timer 1 using an absolute value:

- Use the Timer Selector to select timer 1.
- 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 Basler 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( 100.00 );
```

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

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.

For more information about the pylon API and the pylon Viewer, see Section 3.1.1 on page 40.

6.2.4.3 Setting the Timer Duration Time

There are two ways to set the duration time for timer 1:

- 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 Time 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:

- Timer Duration Raw parameter, and
- 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.

  Range of Timer Duration Raw parameter value: 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 duration for a timer:

1. Use the Timer Selector to select a timer.
2. Set the value of the Timer Duration Raw parameter.

You can set the Timer Selector and the Timer Delay Raw 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.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDurationRaw.SetValue( 100 );
```

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.

You can set the Timer Duration Time Base 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 parameter value:

```
Camera.TimerDurationTimebaseAbs.SetValue( 5.0 );
```

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 3.1.1 on page 40.

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:

1. Use the Timer Selector to select timer 1.
2. 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 Basler 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 );
```

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

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 Time 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.

For more information about the pylon API and the pylon Viewer, see Section 3.1.1 on page 40.

6.3 Checking the State of the I/O Lines

6.3.1 Checking the State of the Output Line

You can determine the current state of the output line. To check the state of the output line:

- Use the Line Selector parameter to select output line 1.
- Read the value of the Line Status parameter to determine the current state of the 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 Basler pylon API. The following code snippet illustrates using the API to set the selector and read the parameter value:

```
// Select output line 1 and read the state
Camera.LineSelector.SetValue( LineSelector_Out1 );
bool outputLine1State = Camera.LineStatus.GetValue( );
```

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 3.1.1 on page 40.

6.3.2 Checking the State of All Lines

You can determine the current state of the input line and the output line with a single operation. To check the state of both 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 Basler 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.

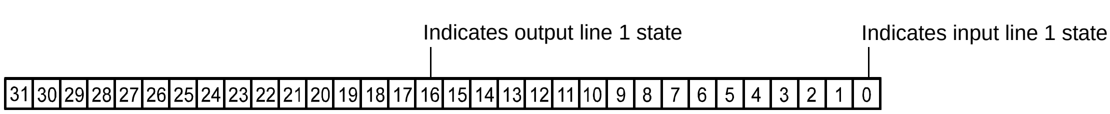


Fig. 44: Line Status All Parameter Bits

7 Image Acquisition Control

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

7.1 Overview

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 chapter, it is helpful to refer to Figure 45 on page 81 and to the use case diagrams in Section 7.11 on page 140. 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 81). 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 81). 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.

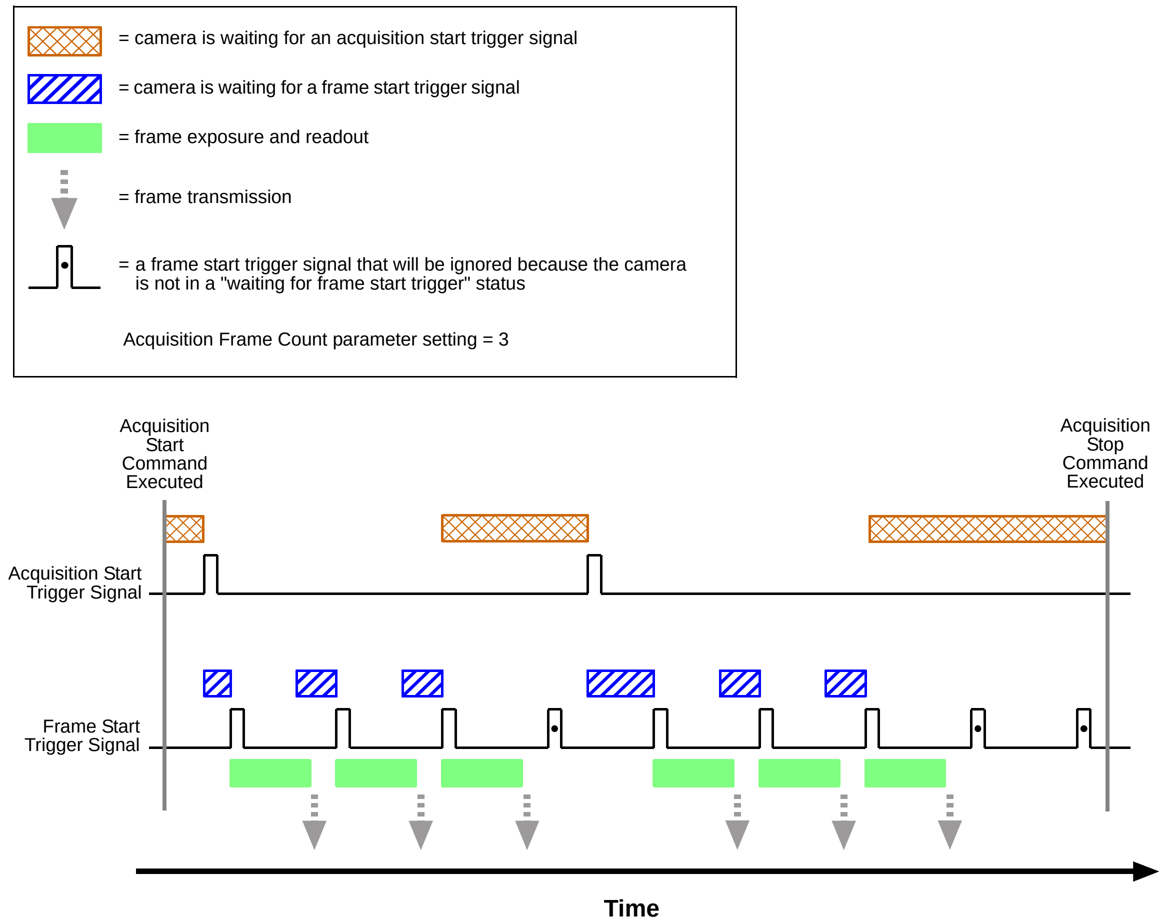


Fig. 45: Acquisition Start and Frame Start Triggering

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 occurance 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, a parameter called the Exposure Time Abs 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 Abs parameter 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.

7.2 Acquisition Start and Stop Commands and the Acquisition Mode

Executing an Acquisition Start commmand 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.

> 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.
>
> To achieve the maximum possible possible acquisition frame rate, set the camera for the continuous acquisition mode and use "overlapped" exposure.
>
> For more information about overlapped exposure, see Section 7.11 on page 140.

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 API and the pylon Viewer, see Section 3 on page 39.

7.3 The Acquisition Start Trigger

(When reading this section, it is helpful to refer to Figure 45 on page 81.)

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.

7.3.1 Acquisition Start Trigger 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.

7.3.1.1 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.

7.3.1.2 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.

7.3.2 Acquisition Frame Count

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.

7.3.3 Setting the Acquisition Start Trigger Mode and Related Parameters

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 API and the pylon Viewer, see Section 3 on page 39.

7.3.4 Using a Software Acquisition Start Trigger

7.3.4.1 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.)

7.3.4.2 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.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

7.3.5 Using a Hardware Acquisition Start Trigger

7.3.5.1 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 7.3.5.2.

For more information about the electrical requirements for Line 1, see Section 5.7 on page 57.

7.3.5.2 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 API and the pylon Viewer, see Section 3 on page 39.

7.4 The Frame Start Trigger

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 7.1 on page 79 and Section 7.2 on page 83.

For more information about the acquisition start trigger, and about the acquisition status, see Section 7.1 on page 79 and Section 7.3 on page 85.

Referring to the use case diagrams that appear in Section 7.11 on page 140 can help you understand the explanations of the frame start trigger.

7.4.1 Frame Start Trigger 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.

7.4.1.1 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.

ⓘ	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 7.1 on page 79 and Section 7.3 on page 85.

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 7.5 on page 103.

7.4.1.2 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. 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 7.4.2 on page 96.

For more information about using a hardware trigger to control frame acquisition start, see Section 7.4.3 on page 98.

> 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 trigger signals when it is in a "waiting for frame start trigger" acquisition status. For more information about the acquisition status, see Section 7.1 on page 79 and Section 7.3 on page 85.

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 value of the camera's Exposure Time Abs parameter.

When the Trigger Mode parameter is set to on and the Trigger Source parameter is set to input line 1, the exposure time for each frame acquisition can be controlled with the Exposure Time Abs parameter 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 7.4.2 on page 96.

For more information about controlling exposure time when using a hardware trigger, see Section 7.4.3 on page 98.

7.4.1.3 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 API and the pylon Viewer, see Section 3 on page 39.

7.4.2 Using a Software Frame Start Trigger

7.4.2.1 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 Abs parameter.

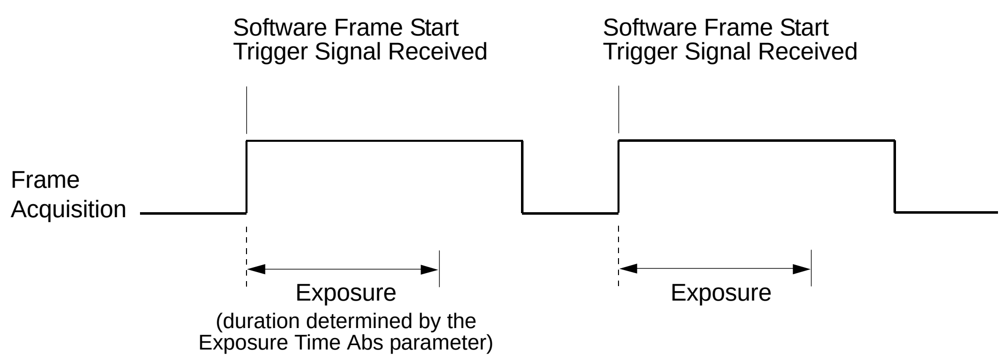


Fig. 46: Frame Acquisition with a Software Frame Start Trigger

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 frame start trigger signals that are applied to the camera when it is not ready to receive them will be ignored.

Section 7.4.2.2 on page 97 includes more detailed information about applying a software frame start trigger signal to the camera using Basler pylon.

For more information about determining the maximum allowed frame rate, see Section 7.12 on page 143.

7.4.2.2 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)
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 API and the pylon Viewer, see Section 3 on page 39.

7.4.3 Using a Hardware Frame Start Trigger

7.4.3.1 Introduction

If the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to line 1, an externally generated electrical signal injected into physical 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.

If you are triggering frame acquisition with an ExFSTrig signal and you attempt to acquire frames at too high a rate, some of the frame trigger signals that you apply will be received by the camera when it is not in a "waiting for frame start trigger" acquisition status. The camera will ignore any frame start trigger signals that it receives when it is not "waiting for frame 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 frame start triggering and selecting the input line to receive the ExFSTrig signal, see Section 7.4.3.4 on page 101.

For more information about the electrical requirements for line 1, see Section 5.7 on page 57.

For more information about determining the maximum allowed frame rate, see Section 7.12 on page 143.

7.4.3.2 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 value of the camera's Exposure Time Abs parameter. 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. Figure 47 illustrates timed exposure with the camera set for rising edge triggering.

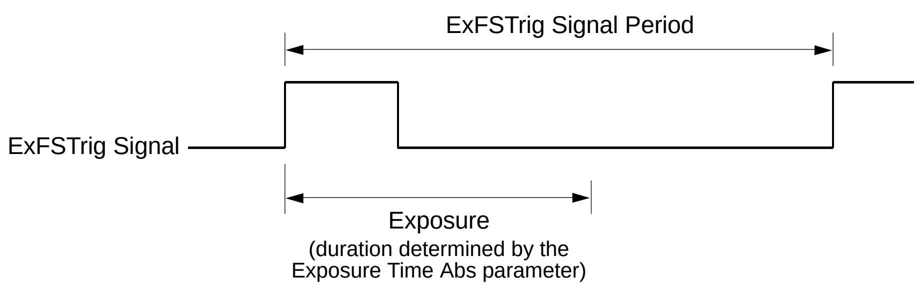


Fig. 47: Timed Exposure with Rising Edge Triggering

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 in Figure 48 for rising edge triggering.

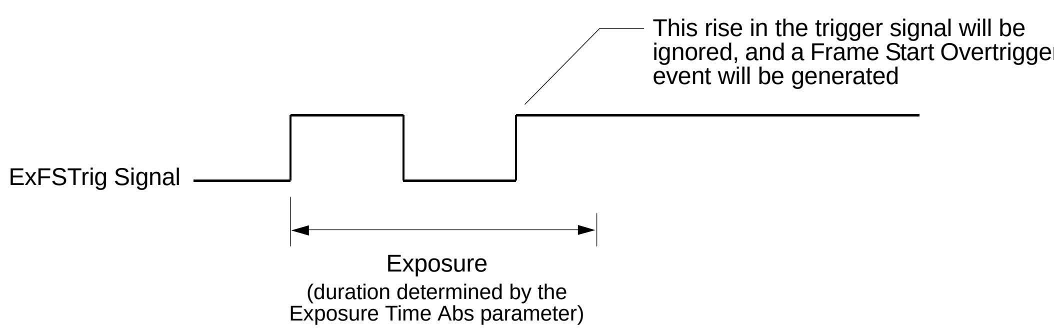


Fig. 48: Overtriggering with Timed Exposure

For more information about the Frame Start Overtrigger event, see Section 10.14 on page 301.

For more information about the camera's Exposure Time Abs parameter, see Section 7.5 on page 103.

Trigger Width Exposure Mode

ⓘ	Trigger width exposure mode is not available on acA750-30gm/gc cameras and is not available on acA2500-14gm/gc cameras.

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.

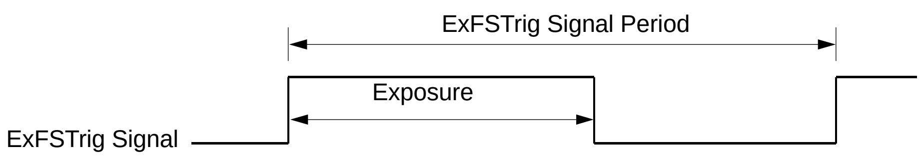


Fig. 49: Trigger Width Exposure with Rising Edge Triggering

When you operate the camera in trigger width exposure mode, you must also set the camera's Exposure Overlap Time Max Abs parameter. This parameter setting will be used by the camera to operate the Frame Trigger Wait signal.

You should set the Exposure Overlap Time Max Abs parameter value 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 set the camera's Exposure Overlap Time Max Abs parameter to 3000 µs.

For more information about the Frame Trigger Wait signal and the Exposure Overlap Time Max Abs parameter, see Section 7.10.4 on page 132.

7.4.3.3 Frame Start Trigger Delay

The frame start 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 frame start trigger delay can be specified in the range from 0 to 1000000 µs (equivalent to 1 s). When the delay is set to 0 µs, no delay will be applied.

To set the frame start trigger delay:

- Set the camera's Trigger Selector parameter to frame start.
- Set the value of the Trigger Delay Abs parameter.

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

7.4.3.4 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 exposure overlap time max abs - the shortest exposure time
// we plan to use is 1500 us
Camera.ExposureOverlapTimeMaxAbs.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 API and pylon Viewer, see Section 3 on page 39.

7.5 acA-750 Acquisition Control Differences

7.5.1 Overview

In almost all respects, acquisition triggering on acA750 model cameras adheres to the acquisition control description provided throughout in this chapter. But because the acA750 models have an interlaced sensor (rather than the standard progressive scan sensor used on the other camera models), there are some significant differences.

With the architecture of the acA750 sensor, there is only one vertical shift register for each two physical pixels in the sensor. This leads to what is commonly known as a "field" readout scheme for the sensor. There are two fields that can be read out of the sensor "Field 0" and "Field 1". The main difference between Field 0 and Field 1 is that they combine the pixels in the sensor rows in different ways.

As shown in Figure 50, with Field 0 readout the pixel values from row 0 are binned with the pixel values from row 1, the pixel values from row 2 are binned with the pixel values from row 3, the pixel values from row 4 are binned with the pixel values from row 5, and so on.

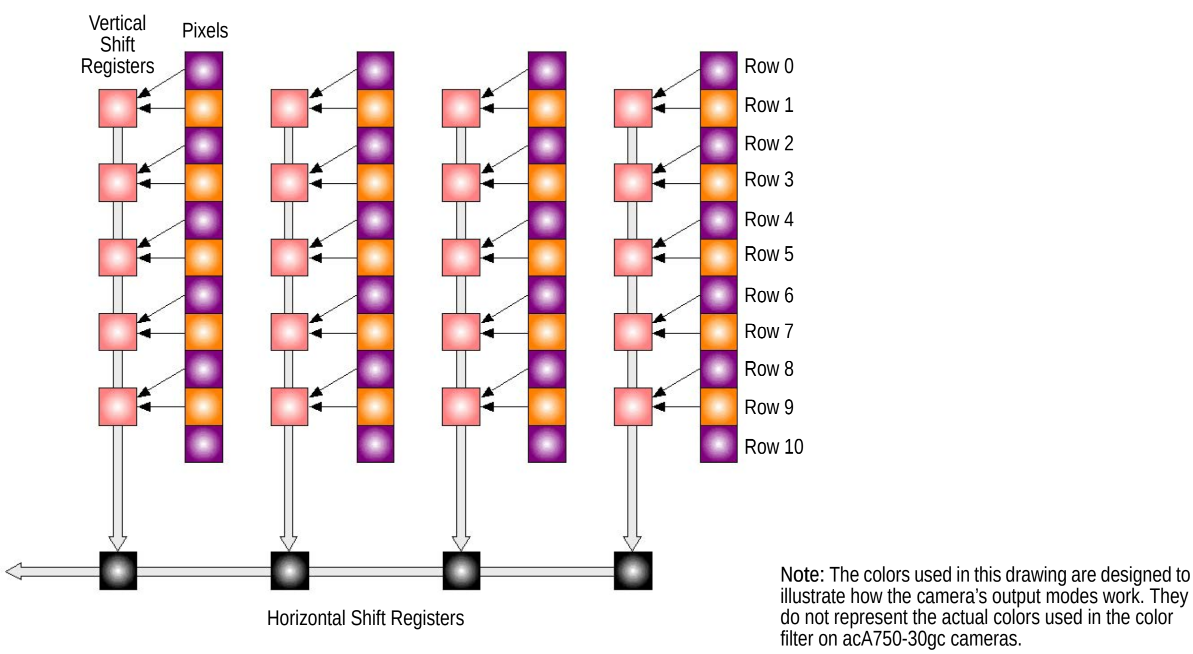


Fig. 50: Field 0 Readout

As shown in Figure 51, with Field 1 readout the pixel values from row 1 are binned with the pixel values from row 2, the pixel values from row 3 are binned with the pixel values from row 4, the pixel values from row 5 are binned with the pixel values from row 6, and so on

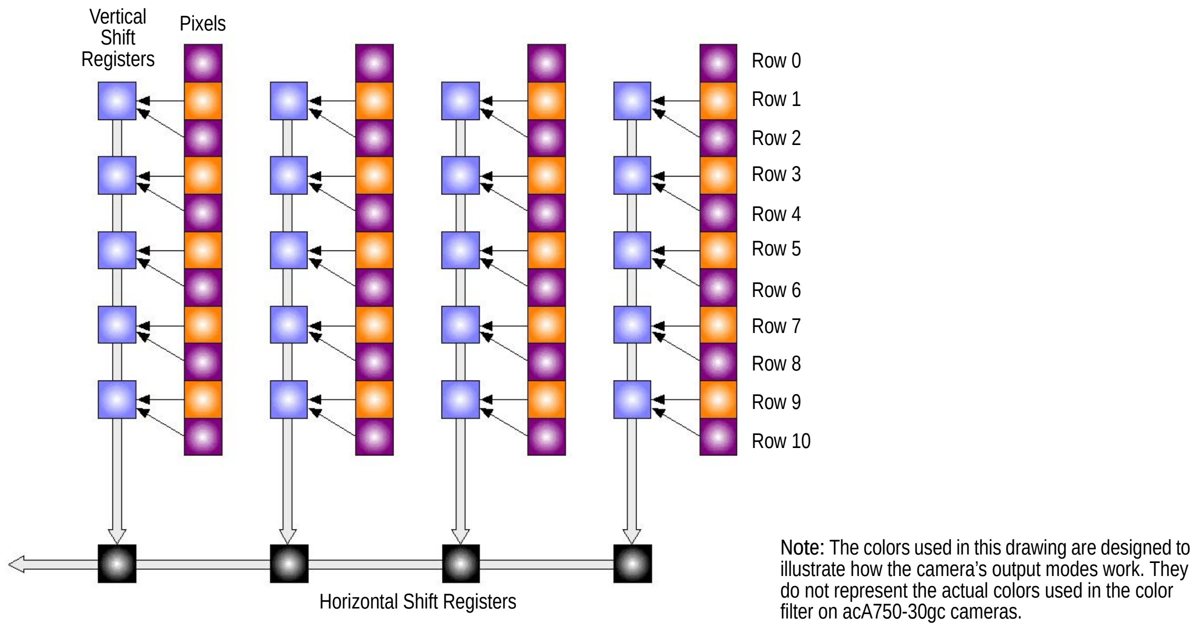


Fig. 51: Field 1 Readout

7.5.2 Field Output Modes

On acA750 cameras, four "field output modes" are available: field 0, field 1, concatenated new fields, and deinterlaced new fields.

Field 0 Output Mode: Each time the camera receives a frame trigger signal, it acquires, reads out, and transmits a frame using the field 0 scheme described in Section 7.5.1 on page 103. Because pairs of rows are combined, the transmitted image is commonly referred to as "half height", i.e., the number of vertical pixels in the transmitted image will be one half of the number of physical pixels in the sensor.

In Field 0 output mode, the pixel data from field 0 is considered to be a frame. Each time the camera receives a frame trigger signal, it will acquire field 0 and will transmit the field 0 pixel data as a frame.

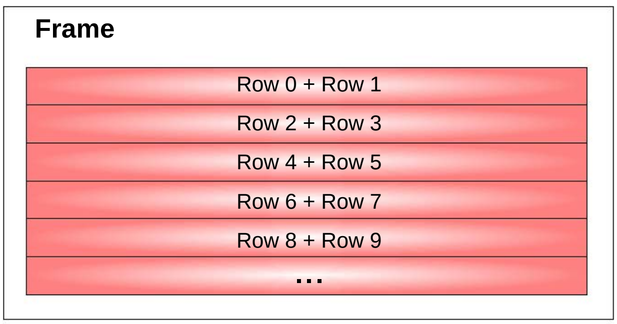


Fig. 52: Field 0 Output Mode

Field 1 Output Mode: Each time the camera receives a frame trigger signal, it acquires, reads out, and transmits a frame using the field 1 scheme described in Section 7.5.1 on page 103. Because pairs of rows are combined, the transmitted image is commonly referred to as "half height", i.e., the number of vertical pixels in the transmitted image will be one half of the number of physical pixels in the sensor.

In Field 1 output mode, the pixel data from field 1 is considered to be a frame. Each time the camera receives a frame trigger signal, it will acquire field 1 and will transmit the field 1 pixel data as a frame.

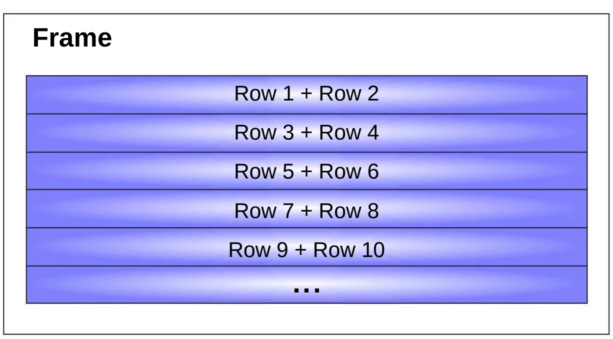


Fig. 53: Field 1 Output Mode

Concatenated New Fields Output Mode: Each time the camera receives a frame trigger signal it acquires two fields, combines them into a single frame, and transmits the frame.

After receiving a frame trigger signal, the camera first acquires and reads out an image using the field 0 scheme and it places this image into the camera's memory. The camera then automatically acquires and reads out a second image using the field 1 scheme. The data from the two acquired images is concatenated as shown in Figure 54, and the concatenated image data is transmitted as a single frame.

In concatenated new fields output mode, the concatenated pixel data from field 0 plus field 1 is considered to be a frame. It is not necessary to issue a separate frame trigger signal to acquire each field. When a frame trigger signal is issued to the camera, it will first acquire field 0 and will then automatically acquire field 1 without the need for a second frame trigger signal. When acquiring each field, the camera will use the full exposure time indicated by the camera's exposure time parameter setting.

If a camera is operating in concatenated new fields output mode and is set, for example, for 30 frames per second, it will acquire 60 fields per second. Since two fields are combined to produce one frame, the camera will end up transmitting 30 frames per second. When set for a 30 frames per second rate, the camera will begin acquiring field 0 each time it receives a frame trigger signal and will automatically begin acquiring field one 1/60th of a second later.

The main advantages of using the concatenated new fields output mode are that it provides pixel data for a "full height" image and that it provides much more image information about a given scene.

The disadvantages of using the concatenated new fields output mode is that the image data must be deinterlaced in order to use it effectively and that, if the object being imaged is moving, there can be significant temporal distortion in the transmitted frame.

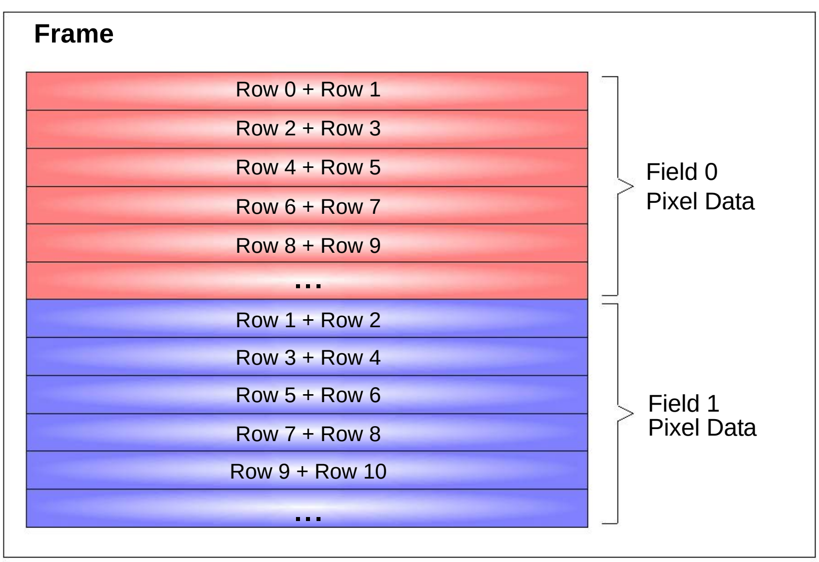


Fig. 54: Concatenated New Fields Output Mode

Deinterlaced New Fields Output Mode: Each time the camera receives a frame trigger signal it acquires two fields, combines them into a single frame, and transmits the frame.

After receiving a frame trigger signal, the camera first acquires and reads out an image using the field 0 scheme and it places this image into the camera's memory. The camera then acquires and reads out a second image using the field 1 scheme. The data from the two acquired images is deinterlaced as shown in Figure 55, and the deinterlaced image data is transmitted as a single frame.

In deinterlaced new fields output mode, the deinterlaced pixel data from field 0 plus field 1 is considered to be a frame. It is not necessary to issue a separate frame trigger signal to acquire each field. When a frame trigger signal is issued to the camera, it will first acquire field 0 and will then automatically acquire field 1 without the need for a second frame trigger signal. When acquiring each field, the camera will use the full exposure time indicated by the camera's exposure time parameter setting.

If a camera is operating in deinterlaced new fields output mode and is set, for example, for 30 frames per second, it will acquire 60 fields per second. Since two fields are combined to produce one frame, the camera will end up transmitting 30 frames per second. When set for a 30 frames per second rate, the camera will begin acquiring field 0 each time it receives a frame trigger signal and will automatically begin acquiring field one 1/60th of a second later.

The main advantages of using the deinterlaced new fields output mode are that it provides pixel data for a "full height" image and that it provides much more image information about a given scene.

The disadvantage of using the deinterlaced new fields output mode is that, if the object being imaged is moving, there can be significant temporal distortion in the transmitted frame.

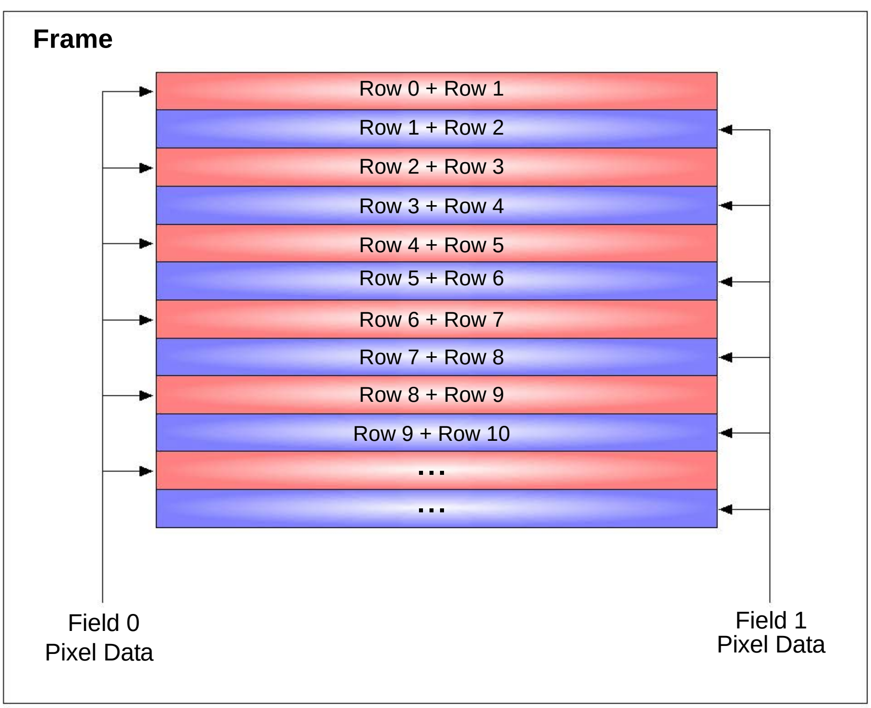


Fig. 55: Deinterlaced New Fields Output Mode

7.5.3 Setting the Field Output Mode

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

```
// Set the field output mode to Field 0
Camera.FieldOutputMode.SetValue( Field0 );

// Set the field output mode to Field 1
Camera.FieldOutputMode.SetValue( Field1 );

// Set the field output mode to Concatenated New Fields
Camera.FieldOutputMode.SetValue( ConcatenatedNewFields );

// Set the field output mode to Deinterlaced New Fields
Camera.FieldOutputMode.SetValue( DeinterlacedNewFields );
```

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 page 39.

7.6 Setting the Exposure Time

This section (Section 7.6) describes how the exposure time can be adjusted "manually", i.e., by setting the value of the exposure time parameter.

The camera also has an Exposure Auto function that can automatically adjust the exposure time. **Manual adjustment of the exposure time parameter will only work correctly, if the Exposure Auto function is disabled.**

For more information about auto functions in general, see Section 10.12.1 on page 282.

For more information about the Exposure Auto function in particular, see Section 10.12.5 on page 293.

All Models Except the acA2500gm/gc

If you are operating the camera in any one of the following ways, you must specify an exposure time by setting the camera's Exposure Time Abs parameter:

- the frame start trigger mode is set to off
- the frame start trigger mode is set to on and the trigger source is set to software
- the frame start trigger mode is set to on, the trigger source is set to line 1, and the exposure mode is set to timed.

The Exposure Time Abs parameter must not be set below a minimum specified value. The minimum setting for each camera model is shown in Table 12.

The maximum possible exposure time that can be set is also shown in Table 12.

Camera Model	Minimum Allowed Exposure Time	Maximum Possible Exposure Time
acA640-90gm/gc	17 µs	1000000 µs
acA640-100gm/gc	4 µs	1000000 µs
acA645-100gm/gc	20 µs	10000000 µs
acA750-30gm/gc	30 µs	1000000 µs
acA780-75gm/gc	20 µs	10000000 µs
acA1300-30gm/gc	16 µs	10000000 µs
acA1600-20gm/gc	25 µs	1000000 µs
acA2000-50gm/gc	24 µs	1000000 µs
acA2000-50gmNIR	24 µs	1000000 µs
acA2040-25gm/gc	24 µs	1000000 µs
acA2040-25gmNIR	24 µs	1000000 µs

Table 12: Minimum Allowed Exposure Time Setting and Maximum Possible Exposure Time Setting

The Exposure Time Abs parameter sets the exposure time in µs. The parameter can be set in increments of 1 µs.

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

```
// Set the exposure time to 3000 µs
Camera.ExposureTimeAbs.SetValue( 3000 );
```

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

For more information about the pylon API and pylon Viewer, see Section 3.1.1 on page 40.

acA2500gm/gc Only

You must specify an exposure time by setting the camera's Exposure Time Abs parameter: The Exposure Time Abs parameter can be set in a range from 35 µs to 999985 µs and can be set in increments of 35 µs.

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

```
// Set the exposure time to 3500 µs
Camera.ExposureTimeAbs.SetValue( 3500 );
```

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

For more information about the pylon API and pylon Viewer, see Section 3.1.1 on page 40.

7.7 Electronic Shutter Operation

All ace cameras are equipped with imaging sensors that have an electronic shutter. There are two types of electronic shutters used in the sensors: global and rolling.

All ace models except the acA2500-14gm/gc use sensors with global shutters. The acA2500-14gm/gc models use a sensor with a rolling shutter.

The following sections describe the differences between a global shutter and a rolling shutter.

7.7.1 Global Shutter (All Cameras Except acA2500-14)

All camera models other than the acA2500-14gm/gc are equipped with an electronic global shutter. On cameras equipped with a global shutter, when frame acquisition is triggered, exposure begins for all lines in the sensor as shown in Figure 56. Exposure continues for all lines in the sensor until the programmed exposure time ends (or when the frame start trigger signal ends the exposure time, if the camera is using the trigger width exposure mode). At the end of the exposure time, exposure ends for all lines in the sensor. Immediately after the end of exposure, pixel data readout begins and proceeds in a linewise fashion until all pixel data is read out of the sensor.

A main characteristic of a global shutter is that for each frame acquisition, all of the pixels in the sensor start exposing at the same time and all stop exposing at the same time. This means that image brightness tends to be more uniform over the entire area of each acquired image, and it helps to minimize problems with acquiring images of objects in motion.

The cameras can provide an exposure active output signal that will go high when the exposure time for a frame acquisition begins and will go low when the exposure time ends.

You can determine the readout time for a frame by checking the value of the camera’s Readout Time Abs parameter.

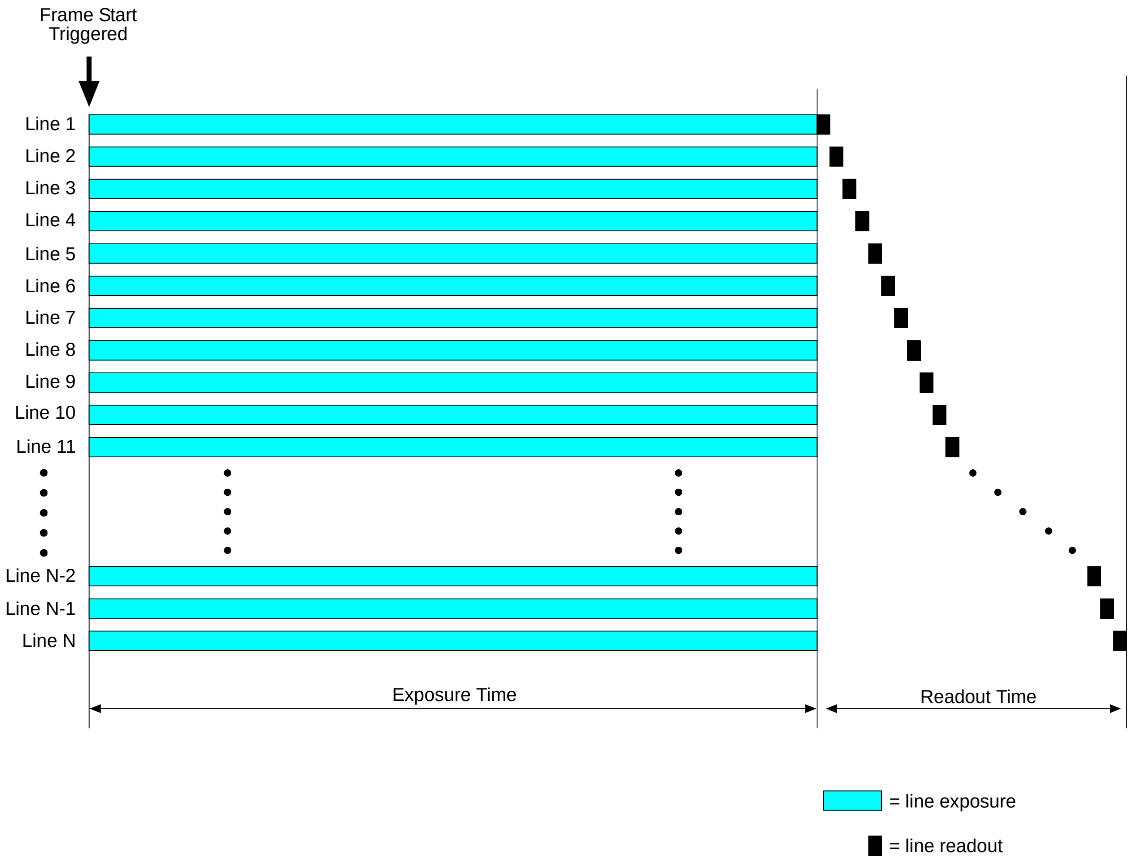


Fig. 56: Global Shutter

For more information about the exposure active output signal, see Section 7.10.1 on page 127.

For more information about the Readout Time Abs parameter, see Section 7.11 on page 140.

7.7.2 Rolling Shutter (acA2500-14 Only)

All acA2500-14gm/gc cameras are equipped with an electronic rolling shutter. The rolling shutter is used to control the start and stop of sensor exposure. The rolling shutter used in these cameras has two operating modes: electronic rolling shutter mode and global reset release mode.

Electronic Rolling Shutter Mode

When the shutter is in the electronic rolling shutter (ERS) operating mode, it exposes and reads out the pixel lines with a temporal offset (designated as tRow) from one line to the next. When frame start is triggered, the camera resets the top line of pixels of the AOI (line one) and begins exposing that line. The camera resets line two tRow later and begins exposing the line. The camera resets line three tRow later and begins exposing the line. And so on until the bottom line of pixels is reached (see Figure 57).

The exposure time is the same for all lines and is determined by the Exposure Time Abs parameter setting.

The pixel values for each line are read out at the end of exposure for the line. Because the readout time for each line is also tRow, the temporal shift for the end of readout is identical to the temporal shift for the start of exposure.

For the acA2500-14gm/gc, tRow = 35 µs.

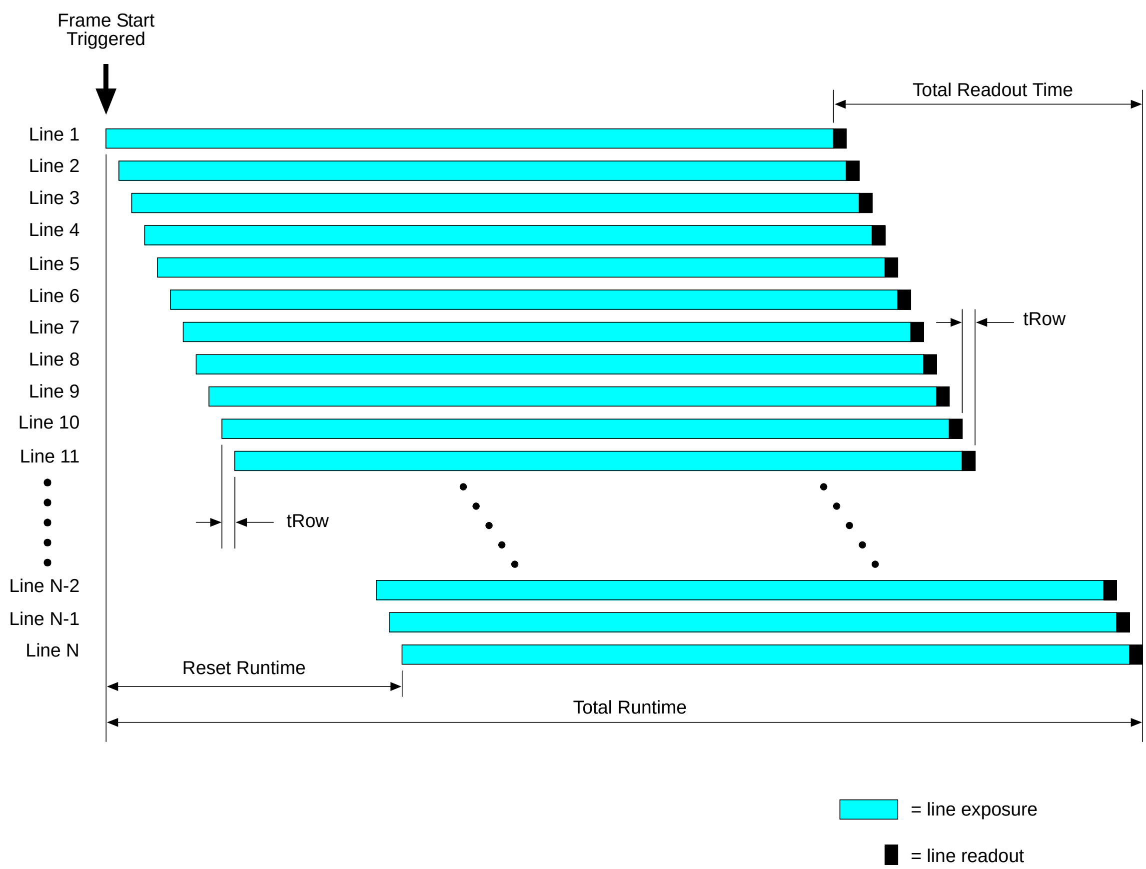


Fig. 57: Rolling Shutter in the ERS Mode

You can calculate the reset runtime using this formula:

Reset Runtime = tRow x (AOI Height -1)

You can calculate the total readout time using this formula:

Total Readout Time = [tRow x (AOI Height)] + 490 µs

You can calculate the total runtime using this formula:

Total Runtime = Exposure Time Abs Parameter Setting + Total Readout Time

The cameras can provide an exposure active output signal that will go high when the exposure time for line one begins and will go low when the exposure time for line one ends.

If the camera is operating with the rolling shutter in ERS mode and you are using the camera to capture images of moving objects, the use of flash lighting is most strongly recommended. The camera supplies a flash window output signal to facilitate the use of flash lighting.

For more information about the exposure active output signal, see Section 7.10.1 on page 127.

For more information about the Exposure Time Abs parameter, see Section 7.6 on page 109.

For more information about the flash window, see Section 7.7.2.1 on page 116.

Global Reset Release Mode

When the shutter is operating in global reset release mode, all of the lines in the sensor reset and begin exposing when frame start is triggered. However, there is a temporal offset (designated as tRow) from one line to the next in the end of exposure. The exposure time for line one is determined by the Exposure Time Abs parameter setting. The exposure for line two will end tRow after the exposure ends for line one. The exposure for line three will end tRow after the exposure ends for line two. And so on until the bottom line of pixels is reached (see Figure 58).

The pixel values for each line are read out at the end of exposure time for the line. The readout time for each line is also equal to tRow.

For the acA2500-14gm/gc, tRow = 35 µs.

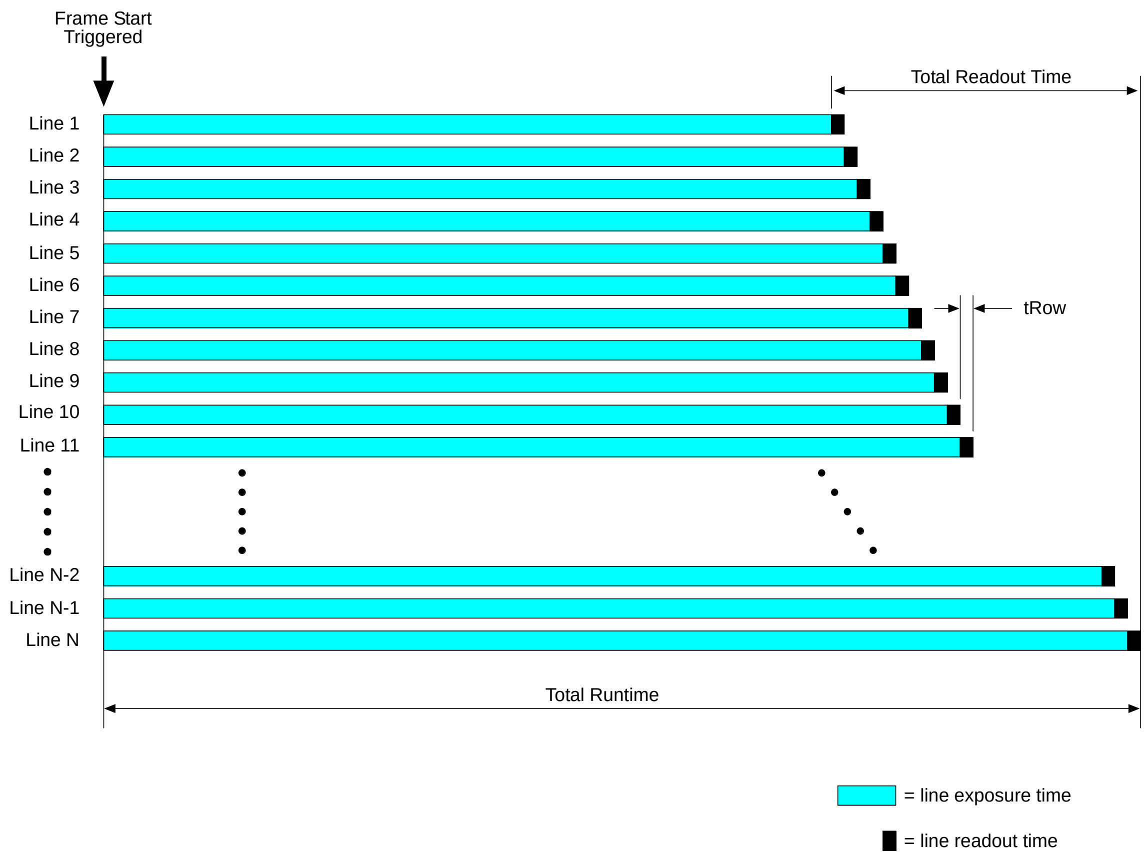


Fig. 58: Rolling Shutter in the Global Reset Release Mode

You can calculate the total readout time using this formula:

Total Readout Time = [tRow x (AOI Height)] + 810 µs

You can calculate the total runtime using the following formula:

Total Runtime = Exposure Time Abs Parameter Setting + Readout Time

The cameras can provide an exposure active output signal that will go high when the exposure time for line one begins and will go low when the exposure time for line one ends.

When the camera is operating with the rolling shutter in the global release mode, the use of flash lighting is most strongly recommended. The camera supplies a flash window output signal to facilitate the use of flash lighting.

For more information about the exposure active output signal, see Section 7.10.1 on page 127.

For more information about the Exposure Time Abs parameter, see Section 7.6 on page 109.

For more information about the flash window, see Section 7.7.2.1 on page 116.

Setting the Shutter Mode

The camera’s shutter has two operating modes: electronic rolling shutter mode and global reset release mode. The shutter will operate in the electronic rolling shutter mode whenever the global reset release mode is disabled. When the global reset release mode is enabled, the shutter will operate in global reset release mode.

You can enable and disable the global reset release mode for the rolling shutter from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to enable and disable the global reset release mode:

```
// Enable the global reset release mode
Camera.GlobalResetReleaseModeEnable.SetValue( true );

// Disable the global reset release mode
Camera.GlobalResetReleaseModeEnable.SetValue( false );
```

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

7.7.2.1 The Flash Window

Flash Window in Electronic Rolling Shutter Mode

If you are using the electronic rolling shutter mode, capturing images of moving objects requires the use of flash exposure. If you don’t use flash exposure when capturing images of moving objects, the images will be distorted due to the temporal shift between the start of exposure for each line.

You can avoid distortion problems by using flash lighting and by applying the flash during the "flash window" for each frame. The flash window is the period of time during a frame acquisition when all of the lines in the sensor are open for exposure.

Figure 59 illustrates the flash window for the electronic rolling shutter mode.

You can calculate when the flash window will open (i.e., the time from the point where the frame is triggered until the point where the window opens) using this formula:

Time to Flash Window Open = tRow x (AOI Height -1)

You can calculate the flash window width (i.e., how long the flash window will remain open) using this formula:

Flash Window Width = Exposure Time Abs Parameter Setting - [(tRow x (AOI Height -1)]

For the acA2500-14gm/gc, tRow = 35 µs.

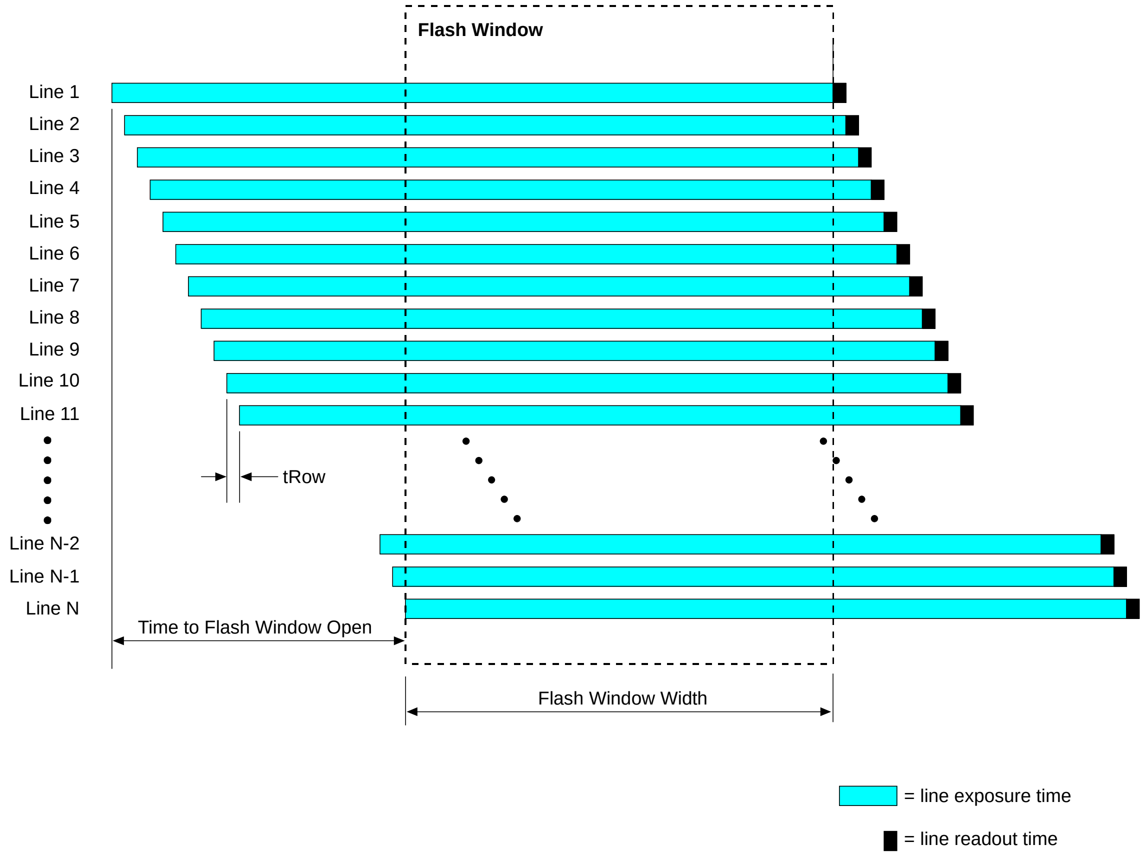


Fig. 59: Flash Window for Rolling Shutter in the ERS Mode

For more information about the Exposure Time Abs parameter, see Section 7.6 on page 109.

Flash Window in Global Reset Release Operating mode

If you are using the global reset release mode, you should use flash exposure for capturing images of both stationary and moving objects. If you don't use flash exposure when capturing images of stationary objects, the brightness in each acquired image will vary significantly from top to bottom due to the differences in the exposure times of the lines. If you don't use flash exposure when capturing images of moving objects, the brightness in each acquired image will vary significantly from top to bottom due to the differences in the exposure times of the lines and the images will be distorted due to the temporal shift between the end of exposure for each line.

You can avoid these problems by using flash lighting and by applying the flash during the "flash window" for each frame. The flash window is the period of time during a frame acquisition when all of the lines in the sensor are open for exposure.

Figure 60 illustrates the flash window for the global reset release mode.

In global reset release mode, the flash window opens when the frame is triggered and closes after a time period equal to the Exposure Time Abs parameter setting. Thus, the flash window width (i.e., how long the flash window will remain open) is equal to the Exposure Time Abs parameter setting.

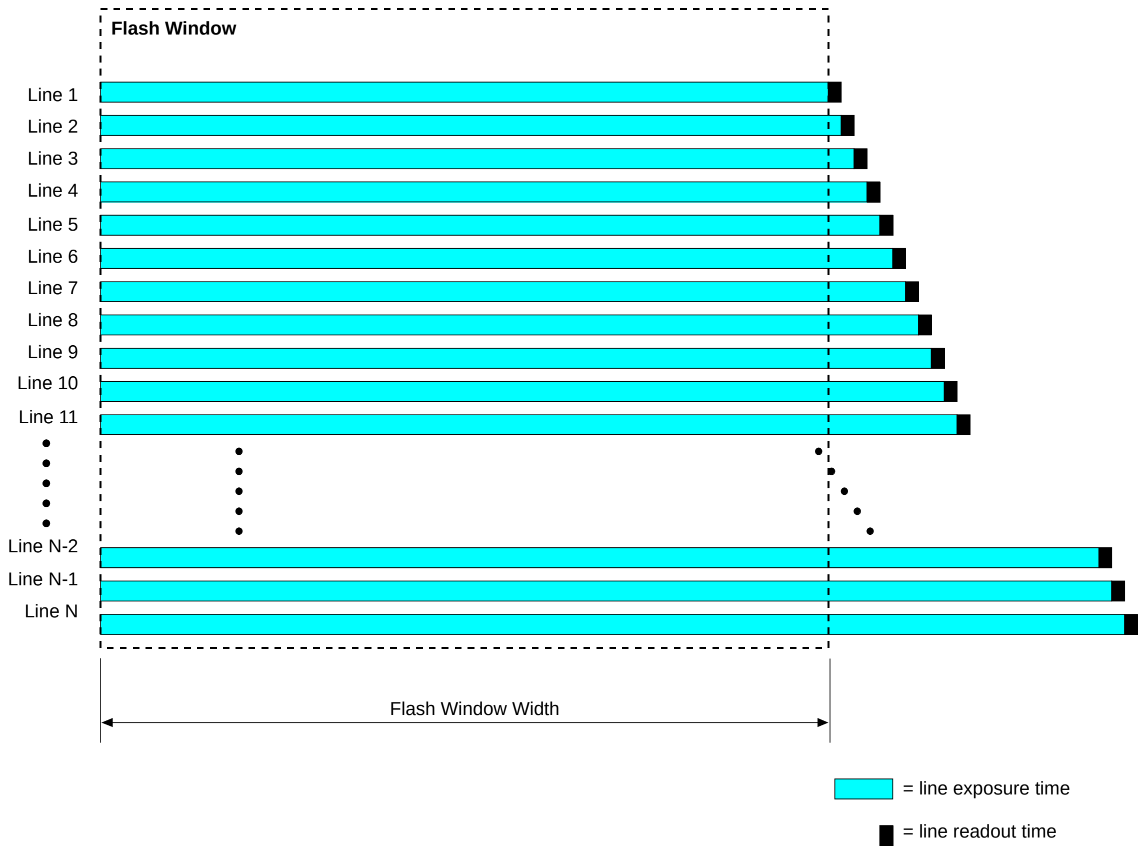


Fig. 60: Flash Window for Rolling Shutter in the Global Reset Release Mode

For more information about the Exposure Time Abs parameter, see Section 7.6 on page 109.

The Flash Window Signal

Cameras with a rolling shutter imaging sensor (e.g., acA2500-14 models) can provide a flash window output signal to aid you in the use of flash lighting. The flash window signal will go high when the flash window for each image acquisition opens and will go low when the flash window closes. Figure 70 illustrates the flash window signal on a camera with the shutter operating in the electronic rolling shutter mode.

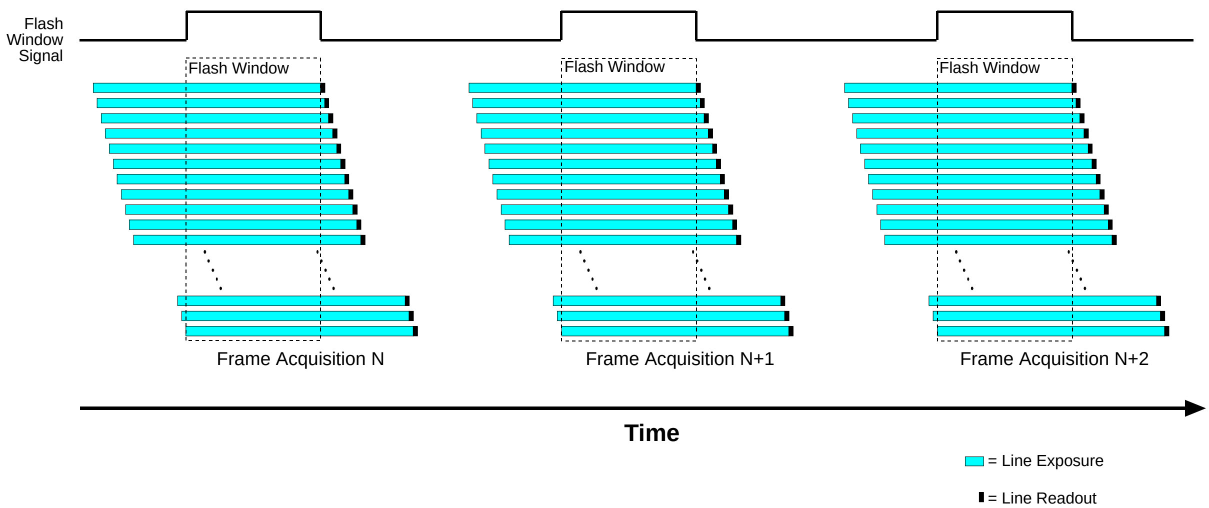


Fig. 61: Flash Window Signal on Cameras with a Rolling Shutter

ⓘ	The flash window signal is also available on cameras with a global shutter imaging sensor. On global shutter cameras, the flash window signal is simply the equivalent of the exposure active signal.

For more information about the flash window signal, see Section 7.10.2 on page 129.

7.8 Overlapping Exposure with Sensor Readout
(All Models Except acA2500-14)

The frame 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 frame 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 a frame is acquired the camera completes the entire exposure/readout process before acquisition of the next frame is started. The exposure for a new frame does not overlap the sensor readout for the previous frame. This situation is illustrated in Figure 62 with the camera set for the trigger width exposure mode.

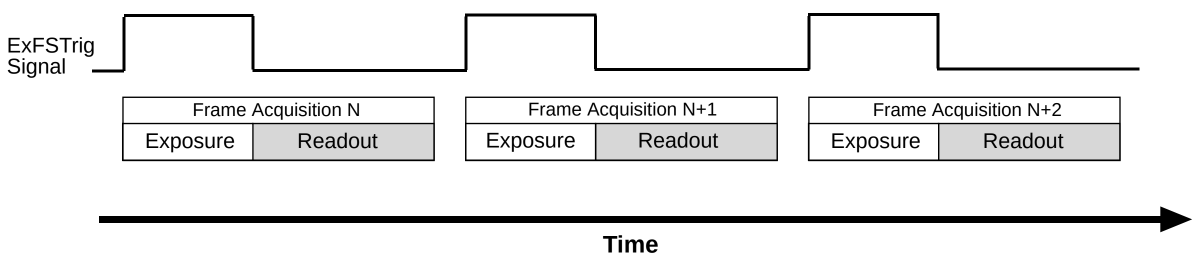


Fig. 62: Non-overlapped Exposure and Readout

In the overlapped mode of operation, the exposure of a new frame begins while the camera is still reading out the sensor data for the previously acquired frame. This situation is illustrated in Figure 63 with the camera set for the trigger width exposure mode.

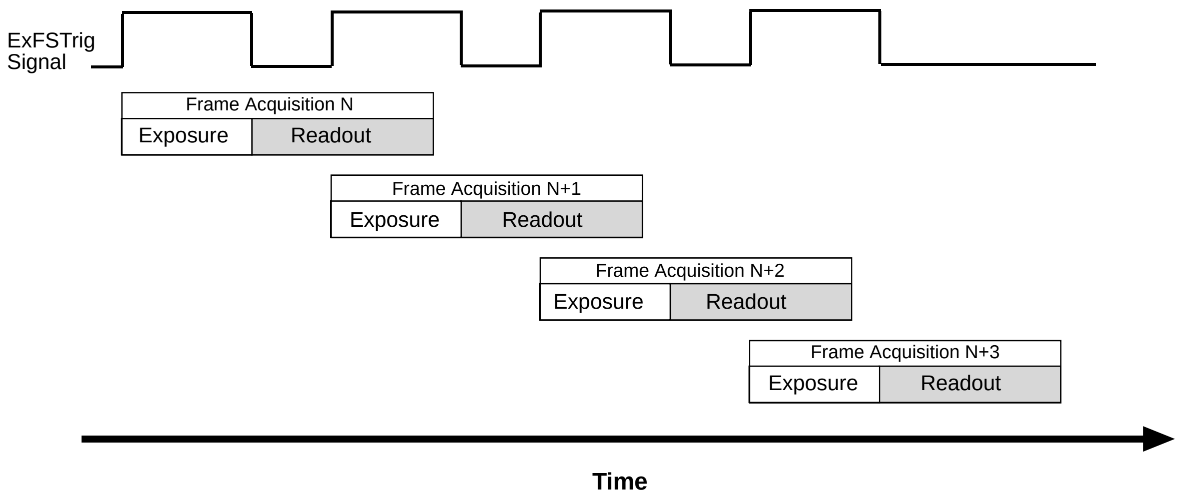


Fig. 63: Overlapped Exposure and Readout

Determining whether your camera is operating with overlapped or non-overlapped exposure and readout 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 and readouts are overlapped or not. If we define the "frame period" as the time from the start of exposure for one frame acquisition to the start of exposure for the next frame acquisition, then:

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

You can determine the readout time by reading the value of the Readout Time Abs parameter. The parameter indicates what the readout time will be in microseconds given the camera's current settings. You can read the Readout Time Abs parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to get the parameter value:

```
double ReadoutTime = Camera.ReadoutTimeAbs.GetValue( );
```

You can also use the Basler pylon Viewer application to easily get the parameter value.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

Guideline for Overlapped Operation with Trigger Width Exposure

If the camera is set for the trigger width exposure mode and you are operating the camera in a way that readout and exposure will be overlapped, there is an important guideline you must keep in mind:

> You must not end the exposure time of the current frame acquisition until readout of the previously acquired frame is complete.

If this guideline is violated, the camera will drop the frame for which the exposure was just ended and will declare a Frame Start Overtrigger event. This situation is illustrated in Figure 64 with the camera set for the trigger width exposure mode with rising edge triggering.

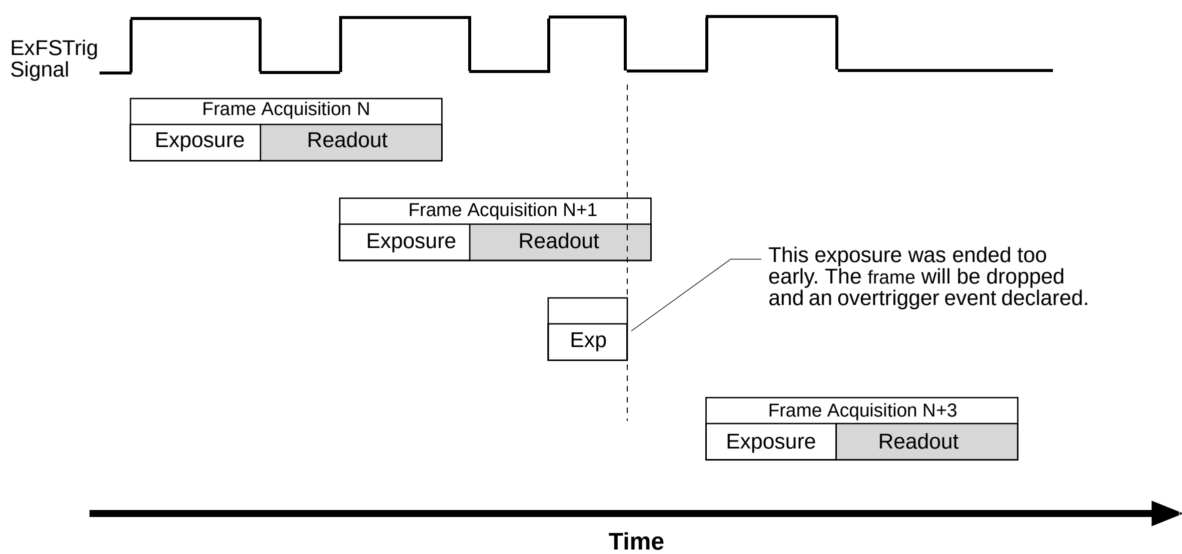


Fig. 64: Overtriggering Caused by an Early End of Exposure

You can avoid violating this guideline by using the camera's Frame Trigger Wait signal to determine when exposure can safely begin and by properly setting the camera's Exposure Overlap Time Max Abs parameter.

For more information about the Frame Trigger Wait signal and the Exposure Overlap Time Max Abs parameter, see Section 7.10.4 on page 132.

For more information about trigger width exposure, see Section 7.4.3.2 on page 99.

7.9 Overlapping Image Acquisitions (acA2500-14 Only)

When using a camera with a rolling shutter, there are two common ways for the camera to operate: with "non-overlapped" acquisition and with "overlapped" acquisition.

In the non-overlapped mode of operation, each time a frame is acquired the camera completes the entire exposure/readout process before acquisition of the next frame is started. The acquisition of a new frame does not overlap any part of the acquisition process for the previous frame. This situation is illustrated in Figure 65 with the camera using an external frame start trigger.

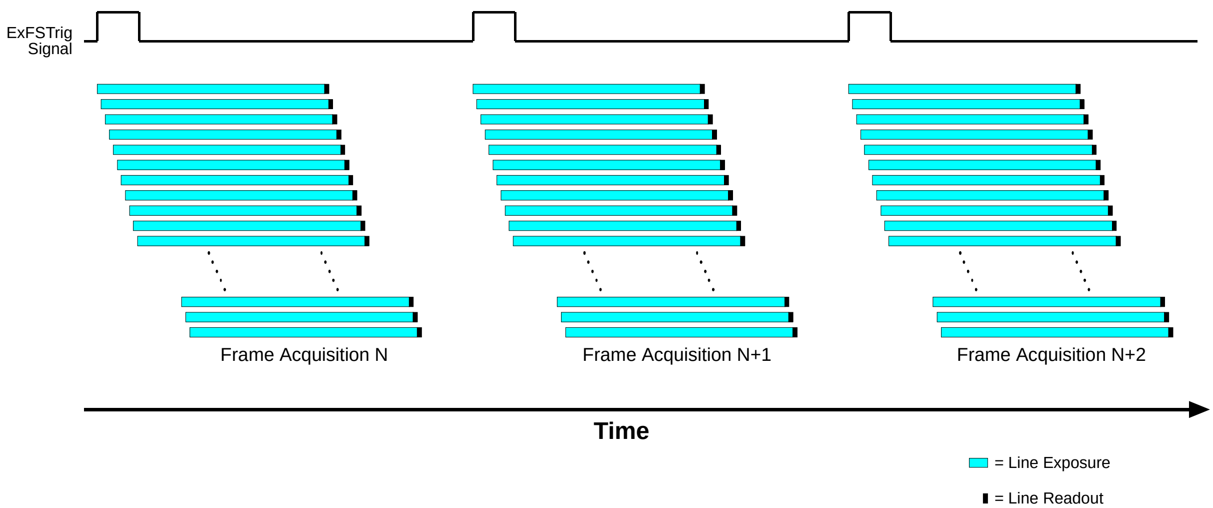


Fig. 65: Non-overlapped Acquisition

In the overlapped mode of operation, the acquisition for a new frame begins while the camera is still completing the acquisition process for the previous frame. This situation is illustrated in Figure 66.

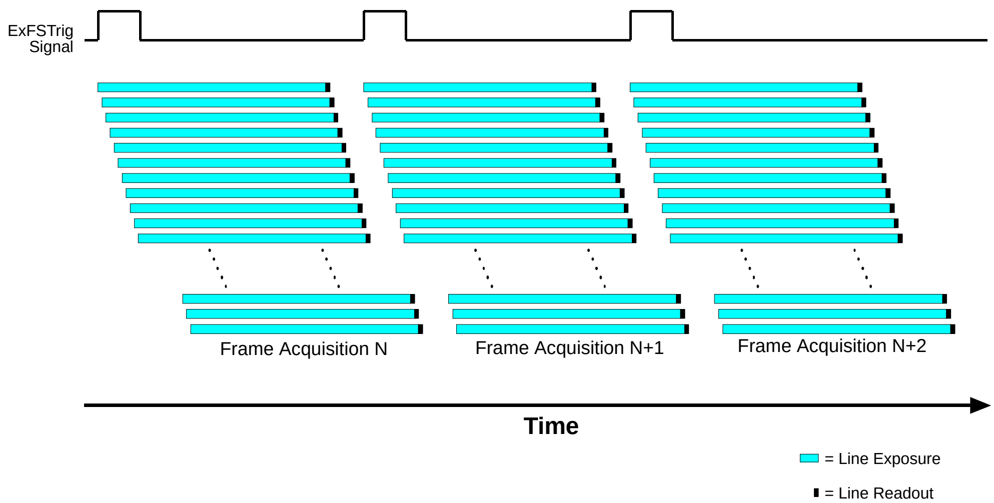


Fig. 66: Overlapped Exposure and Readout

Determining whether your camera is operating with overlapped or with non-overlapped acquisition 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 frame acquisitions are overlapped or not. If we define the “frame period” as the time from the start of exposure for line one in the frame N acquisition to the start of exposure for line one in frame N+1 acquisition, then:

- Exposure will not overlap when:

  Frame Period > Exposure Time Abs Parameter Setting + Total Readout Time

- Exposure will overlap when:

  Frame Period ≤ Exposure Time Abs Parameter Setting + Total Readout Time

ⓘ	Overlapped frame acquisition cannot be performed when the camera is set for global reset release rolling shutter mode. Overlapped frame acquisition can only performed when the camera is in the electronic rolling shutter mode.

ⓘ	If you use the acA2500-14 in the overlapped mode of operation, and you activate the sequencer feature, it depends on the way you use the sequencer, whether the sequencer feature has an effect on the frame rate or not: If the camera takes multiple images ■ with the **same** sequence set, overlapped operation is possible and the sequencer feature has no effect on the camera's frame rate. ■ with **alternating** sequence sets, overlapped operation is not possible. The camera must complete the entire exposure/readout process before a new sequence set can be loaded. In this case the initial overlapped operation turns out to work as non-overlapped operation. As a consequence the frame rate can be significantly reduced.

You can determine the total readout time for a frame by reading the value of the Readout Time Abs parameter. This parameter indicates the time in microseconds from the beginning of readout for line one to the end of readout for line N (the last line). You can read the Readout Time Abs parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to get the parameter value:

```
double ReadoutTime = Camera.ReadoutTimeAbs.GetValue( );
```

You can also use the Basler pylon Viewer application to easily get the parameter value.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

Guideline for Overlapped Acquisition

If you are operating the camera in such a way that frame acquisitions will be overlapped, there is an important guideline you must keep in mind:

> You must wait a minimum of 400 µs after the end of exposure for line one in frame N before you can trigger acquisition of frame N+1. This requirement is illustrated in Figure 67

If this guideline is violated, the camera will ignore the frame start trigger signal and will declare a Frame Start Overtrigger event.

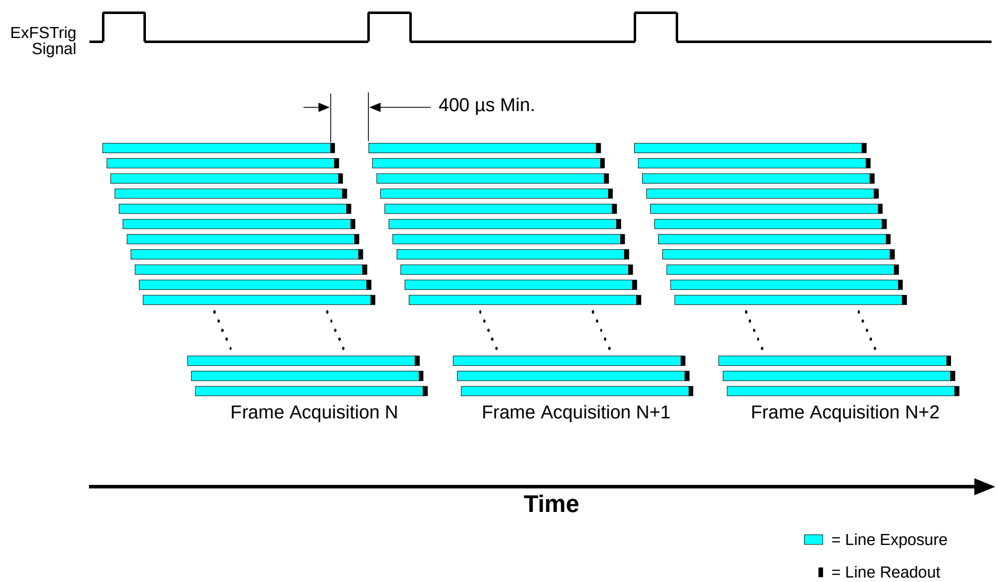


Fig. 67: Acquisition Overlap Guideline

You can avoid violating this guideline by using the camera's Frame Trigger Wait signal to determine when exposure can safely begin.

7.10 Acquisition Monitoring Tools

7.10.1 Exposure Active Signal

Exposure Active on Global Shutter Cameras (All Models Except the acA2500-14)

Cameras with a global shutter imaging sensor can provide an "exposure active" (ExpAc) output signal. On these cameras, the signal goes high when the exposure time for each frame acquisition begins and goes low when the exposure time ends as shown in Figure 68. 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.

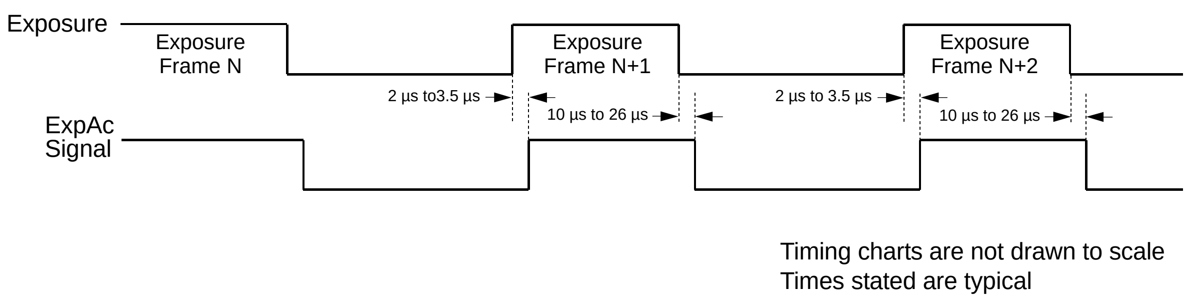


Fig. 68: Exposure Active Signal on Cameras with a Global shutter

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 68 for details.

Exposure Active on Rolling Shutter Cameras (acA2500-14 Only)

Cameras with a rolling shutter imaging sensor can provide an "exposure active" (ExpAc) output signal. On these cameras, the signal goes high when exposure for the first line in a frame begins and goes low when exposure for the first line ends as shown in Figure 69.

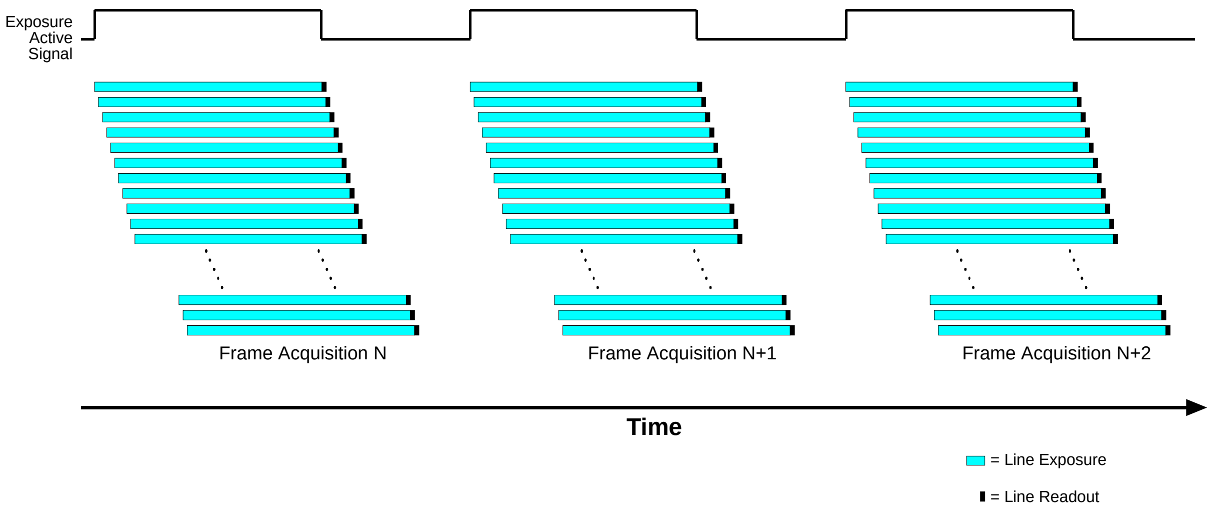


Fig. 69: Exposure Active Signal on Cameras with a Rolling shutter

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

The exposure active output signal can be selected to act as the source signal for 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 the pylon API and the pylon Viewer, see Section 3 on page 39.

For more information about changing which camera output signal is selected as the source signal for the output line, see Section 6.2.1 on page 70.

For more information about the electrical characteristics of the camera's output line, see Section 5.8 on page 62.

7.10.2 Flash Window Signal

Cameras with a rolling shutter imaging sensor (e.g., acA2500-14 models) can provide a flash window output signal to aid you in the use of flash lighting. The flash window signal will go high when the flash window for each image acquisition opens and will go low when the flash window closes. Figure 70 illustrates the flash window signal on a camera with the shutter operating in the electronic rolling shutter mode.

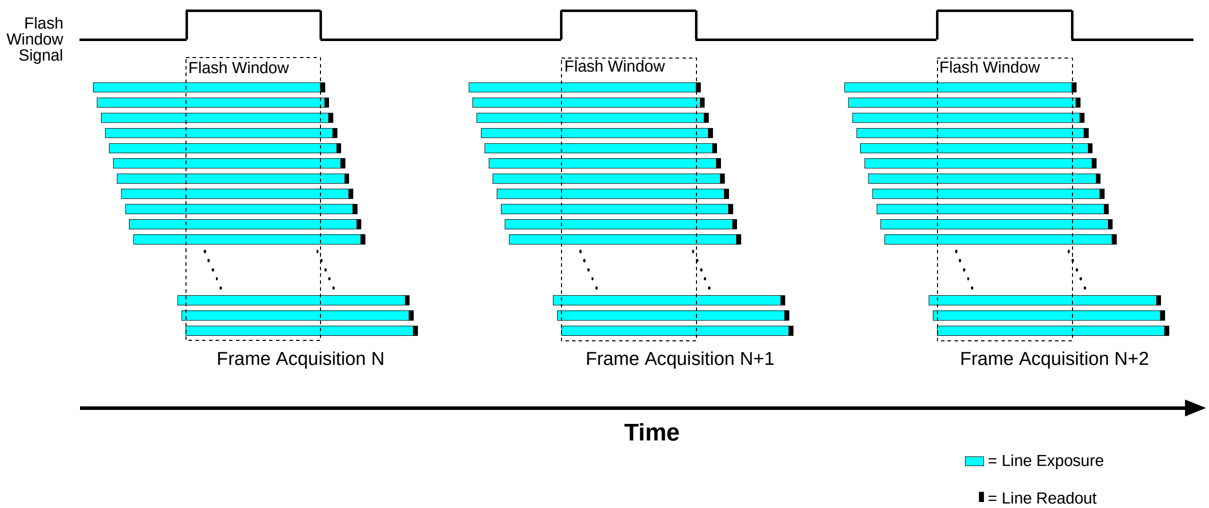


Fig. 70: Flash Window Signal on Cameras with a Rolling Shutter

ⓘ	The flash window signal is also available on cameras with a global shutter imaging sensor. On global shutter cameras, the flash window signal is simply the equivalent of the exposure active signal.

For more information about the rolling shutter and the flash window, see Section 7.7.2 on page 113.

Selecting the Flash Window Signal as the Source Signal for the Output Line

The flash window output signal can be selected to act as the source signal for camera 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 flash window 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_FlashWindow );
```

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 page 39.

For more information about changing which camera output signal is selected as the source signal for the output line, see Section 6.2.1 on page 70.

For more information about the electrical characteristics of the camera's output line, see Section 5.8 on page 62.

7.10.3 Acquisition Status Indicator

If a camera receives a software acquisition start trigger signal when it is not in a "waiting for acquisition start trigger" acquisition status, it will simply ignore the trigger signal and will generate an acquisition start overtrigger event.

If a camera receives a software frame start trigger signal when it is not in a "waiting for frame start trigger" acquisition status, it will simply ignore the trigger signal and will generate a frame start overtrigger event.

The camera's acquisition status indicator gives you the ability to check whether the camera is in a "waiting for acquisition start trigger" acquisition status or in a "waiting for frame start trigger" acquisition status. If you check the acquisition status before you apply each software acquisition start trigger signal or each software frame start trigger signal, you can avoid applying trigger signals to the camera that will be ignored.

The acquisition status indicator is designed for use when you are using host control of image acquisition, i.e., when you are using software acquisition start and frame start trigger signals.

To determine the acquisition status of the camera via the Basler pylon API:

- Use the Acquisition Status Selector to select the Acquisition Trigger Wait status or the Frame Trigger Wait status.
- Read the value of the Acquisition Status parameter.

  If the value is set to "false", the camera is not waiting for the trigger signal.

  If the value is set to "true", the camera is waiting for the trigger signal.

You can check the acquisition status from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to check the acquisition status:

```
// Check the acquisition start trigger acquisition status
// Set the acquisition status selector
Camera.AcquisitionStatusSelector.SetValue
( AcquisitionStatusSelector_AcquisitionTriggerWait );
// Read the acquisition status
bool IsWaitingForAcquisitionTrigger = Camera.AcquisitionStatus.GetValue();

// Check the frame start trigger acquisition status
// Set the acquisition status selector
Camera.AcquisitionStatusSelector.SetValue
( AcquisitionStatusSelector_FrameTriggerWait );
// Read the acquisition status
bool IsWaitingForFrameTrigger = Camera.AcquisitionStatus.GetValue();
```

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

For more information about the pylon API and pylon Viewer, see Section 3 on page 39.

7.10.4 Trigger Wait Signals

If a camera receives a hardware acquisition start trigger signal when it is not in a "waiting for acquisition start trigger" acquisition status, it will simply ignore the trigger signal and will generate an acquisition start overtrigger event.

If a camera receives a hardware frame start trigger signal when it is not in a "waiting for frame start trigger" acquisition status, it will simply ignore the trigger signal and will generate a frame start overtrigger event.

The camera's acquisition trigger wait signal gives you the ability to check whether the camera is in a "waiting for acquisition start trigger" acquisition status. If you check the acquisition trigger wait signal before you apply each hardware acquisition start trigger signal, you can avoid applying acquisition start trigger signals to the camera that will be ignored.

The camera's frame trigger wait signal gives you the ability to check whether the camera is in a "waiting for frame start trigger" acquisition status. If you check the frame trigger wait signal before you apply each hardware frame start trigger signal, you can avoid applying frame start trigger signals to the camera that will be ignored.

These signals are designed to be used when you are triggering acquisition start or frame start via a hardware trigger signal.

7.10.4.1 Acquisition Trigger Wait Signal

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 71 illustrates the Acquisition Trigger Wait signal with the Acquisition Frame Count parameter set to 3 and with exposure and readout overlapped on a camera with a global shutter. 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.

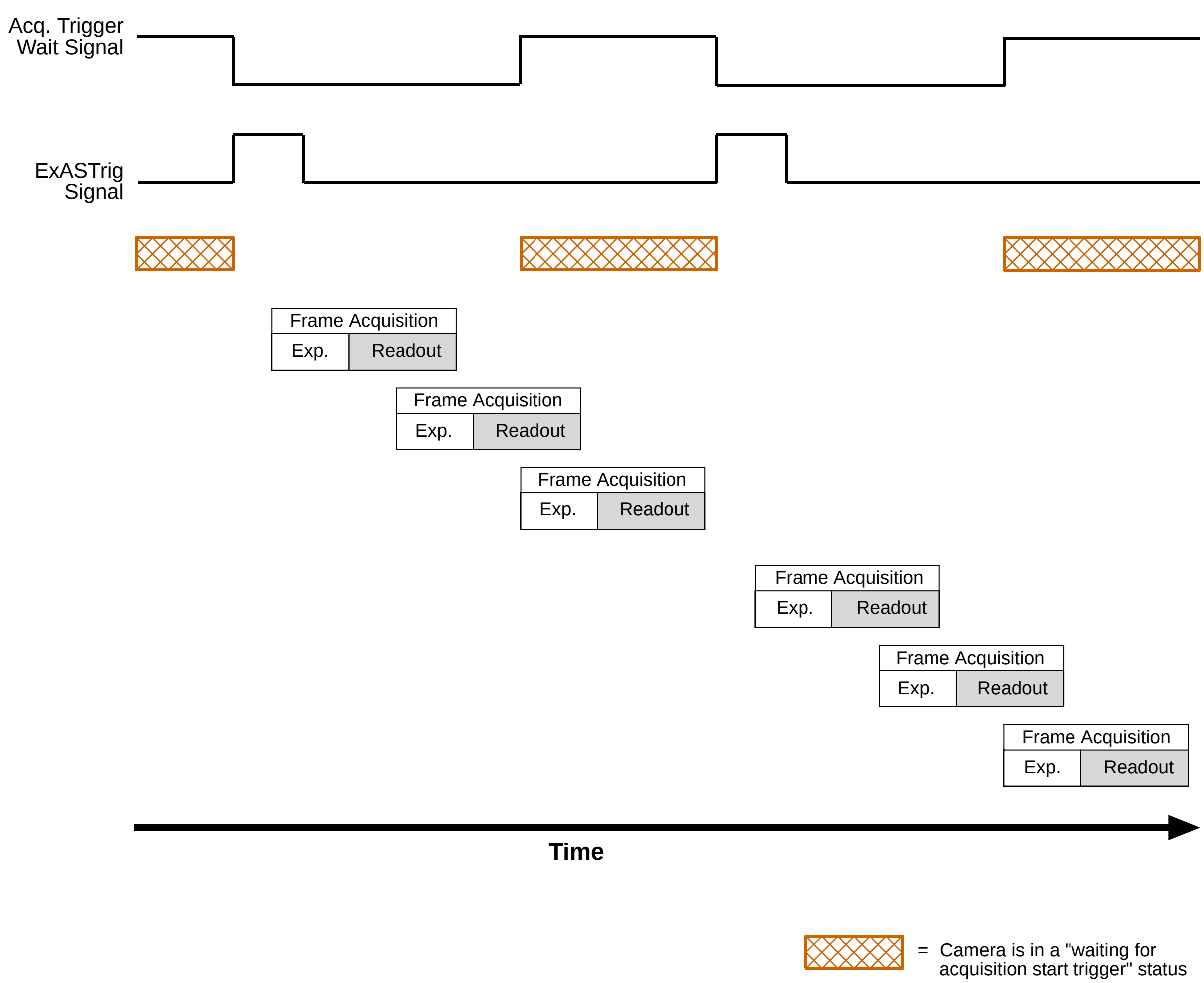


Fig. 71: Acquisition Trigger Wait Signal

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

For more information about event reporting, see Section 10.14 on page 301.

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

The acquisition trigger wait signal can be selected to act as the source signal for camera 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 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.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

For more information about changing which camera output signal is selected as the source signal for the output line, see Section 6.2.1 on page 70.

For more information about the electrical characteristics of the camera's output line, see Section 5.8 on page 62.

7.10.4.2 The Frame Trigger Wait Signal

Overview

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

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

Figure 72 illustrates the Frame Trigger Wait signal on a camera with a global shutter. The camera is set for the trigger width exposure mode with rising edge triggering and with exposure and readout overlapped.

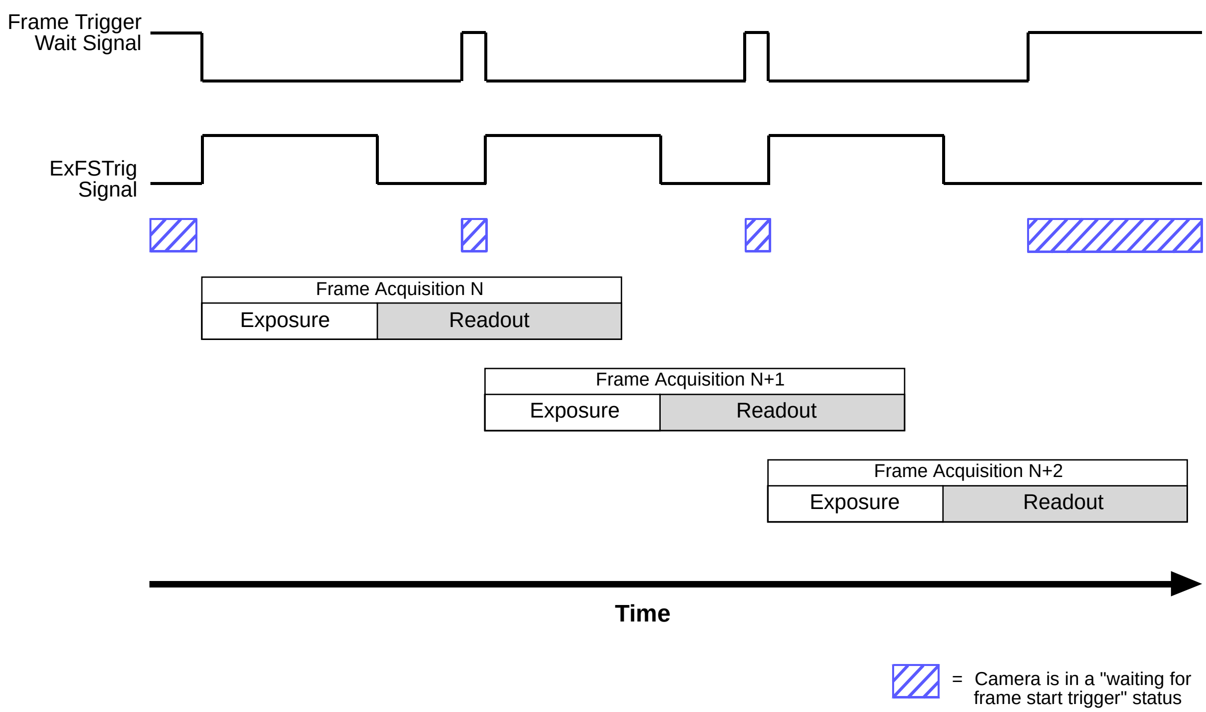


Fig. 72: Frame Trigger Wait Signal

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

For more information about event reporting, see Section 10.14 on page 301.

For more information about hardware triggering, see Section 7.4.3 on page 98.

Frame Trigger Wait Signal Details (All Models Except acA2500-14gm/gc)

When the camera is set for the timed exposure mode, the rise of the Frame Trigger Wait signal is based on the current Exposure Time Abs parameter setting and on when readout of the current frame will end. This functionality is illustrated in Figure 73.

If you are operating the camera in the timed exposure mode, you can avoid overtriggering by always making sure that the Frame Trigger Wait signal is high before you trigger the start of frame capture.

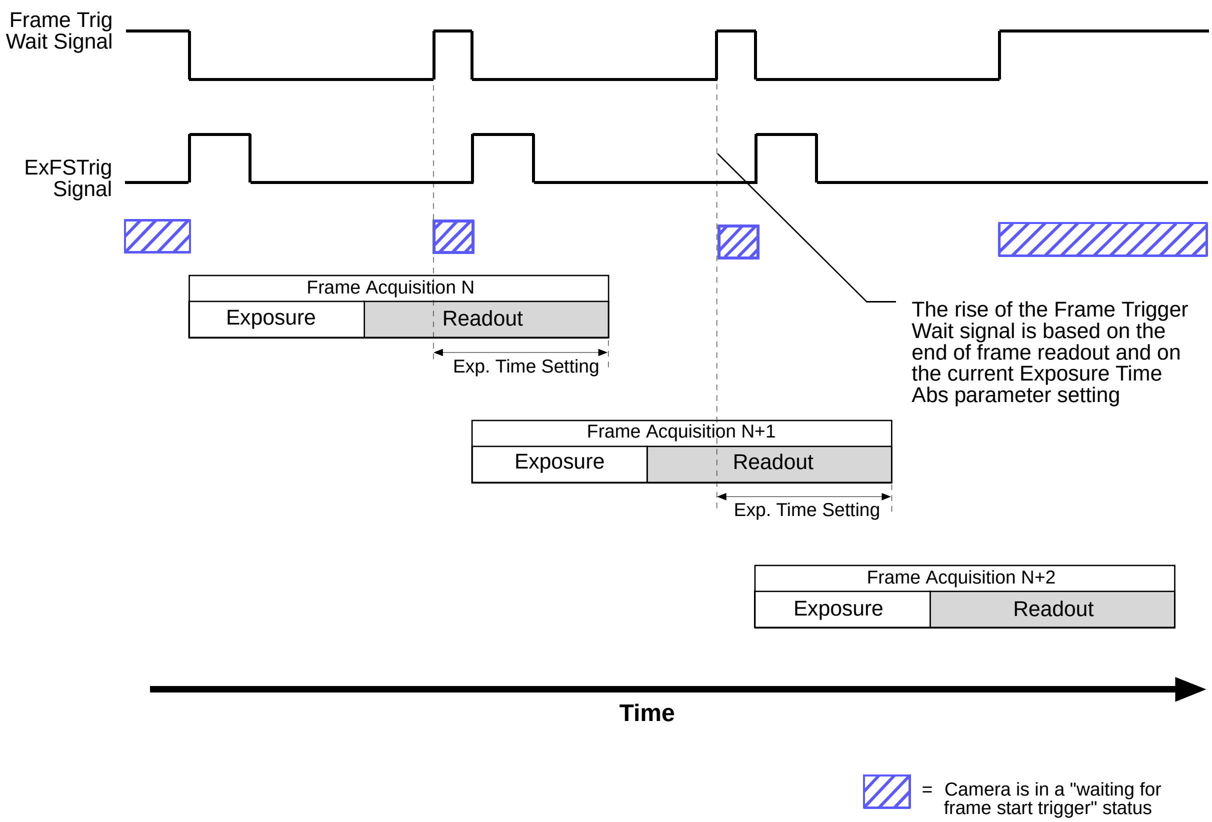


Fig. 73: Frame Trigger Wait Signal with the Timed Exposure Mode

When the camera is set for the trigger width exposure mode, the rise of the Frame Trigger Wait signal is based on the Exposure Overlap Time Max Abs parameter setting and on when readout of the current frame will end. This functionality is illustrated in Figure 74.

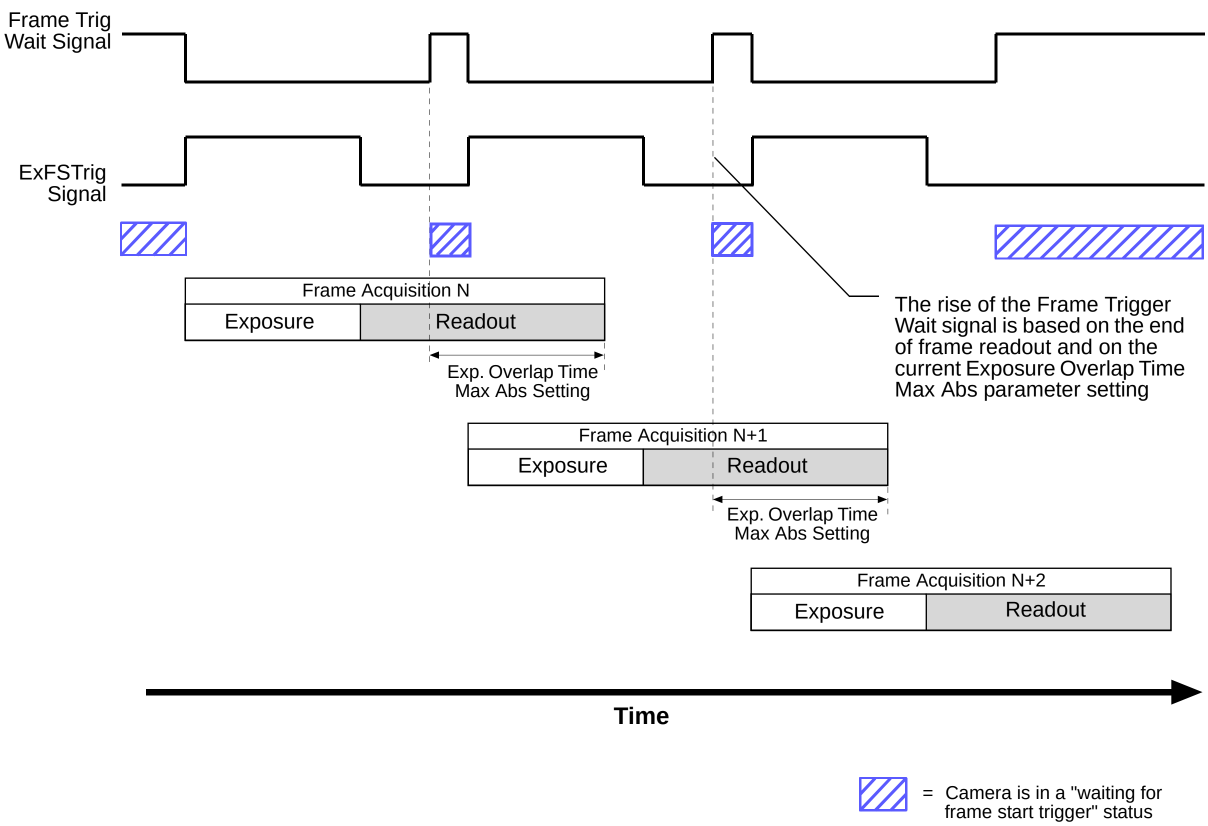


Fig. 74: Frame Trigger Wait Signal with the Trigger Width Exposure Mode

If you are operating the camera in the trigger width exposure mode, you can avoid overtriggering the camera by always doing the following:

- Setting the camera's Exposure Overlap Time Max Abs parameter so that it represents the smallest exposure time you intend to use.
- Making sure that your exposure time is always equal to or greater than the setting for the Exposure Overlap Time Max Abs parameter.
- Monitoring the camera's Frame Trigger Wait signal and only using the ExFSTrig signal to start exposure when the Frame Trigger Wait signal is high.

You should set the Exposure Overlap Time Max Abs parameter value 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 set the camera's Exposure Overlap Time Max Abs parameter to 3000 µs.

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

```
Camera.ExposureOverlapTimeMaxAbs.SetValue( 3000 );
```

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 page 39.

For more information about the electrical characteristics of the camera's output line, see Section 5.8 on page 62.

Frame Trigger Wait Signal Details (acA2500-14 gm/gc Only)

For cameras with a rolling shutter, the rise of the Frame Trigger Wait signal is based on the minimum time required between the end of exposure of the first line in a frame and the start of exposure for the first line in the following frame. This functionality is illustrated in Figure 75.

If you are operating a camera with a rolling shutter, you can avoid overtriggering by always making sure that the Frame Trigger Wait signal is high before you trigger the start of frame capture.

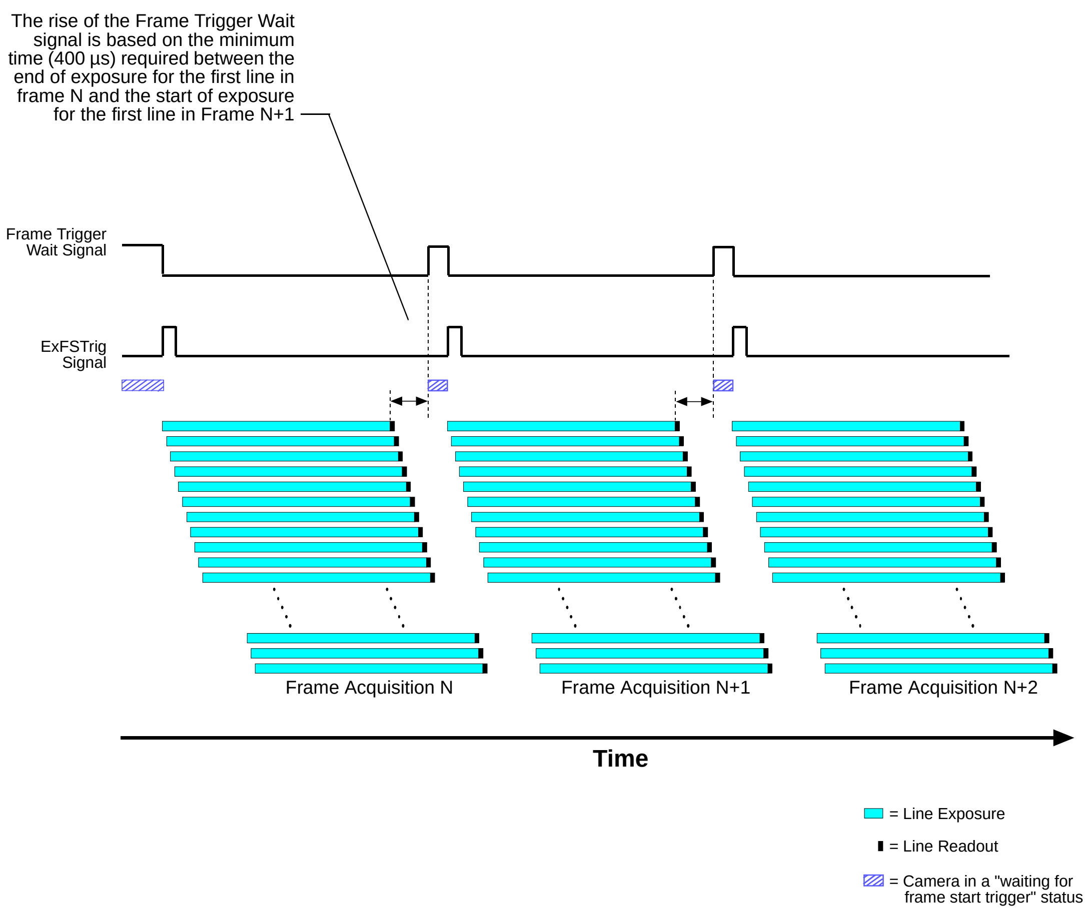


Fig. 75: Frame Trigger Wait Signal on a Rolling Shutter Camera

Selecting the Frame Trigger Wait Signal as the Source Signal for the Output Line

The frame trigger wait signal can be selected to act as the source signal for camera 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 frame 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_FrameTriggerWait );
```

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 page 39.

For more information about changing which camera output signal is selected as the source signal for the output line, see Section 6.2.1 on page 70.

For more information about the electrical characteristics of the camera's output line, see Section 5.8 on page 62.

7.10.5 Camera Events

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

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

For more information about the camera events and event reporting, see Section 10.14 on page 301.

7.11 Acquisition Timing Chart

Figure 76 shows a timing chart for frame acquisition and transmission. The chart assumes that exposure is triggered by an externally generated frame start trigger (ExFSTrig) signal with rising edge activation and that the camera is set for the timed exposure mode.

As Figure 76 shows, there is a slight delay between the rise of the ExFSTrig signal and the start of exposure. After the exposure time for a frame acquisition is complete, the camera begins reading out the acquired frame data from the imaging sensor into a buffer in the camera. When the camera has determined that a sufficient amount of frame 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 frame (or for the acA750, an acquired field) from the imaging sensor into the frame buffer.

The **frame transmission time** is the amount of time it takes to transmit an acquired frame 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 frame data from the sensor to the point where it begins transmitting the data for the acquired frame 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
acA640-90gm/gc	21.48 µs
acA640-100gm/gc	17.62 µs
acA645-100gm/gc	21.84 µs
acA750-30gm/gc	48.97 µs
acA780-75gm/gc	24.50 µs
acA1300-30gm/gc	34.50 µs
acA1600-20gm/gc	45.54 µs
acA2000-50gm/gc	0.2 to 17.2 µs (with frame acquisitions overlapped) 0.2 µs (with frame acquisitions not overlapped)
acA2000-50gmNIR	
acA2040-25gm/gc	
acA2040-25gmNIR	

Table 13: Exposure Start Delays

Camera Model	Exposure Start Delay
acA2500-14gm/gc	848 to 883 µs (with frame acquisitions overlapped) 848 µs (with frame acquisitions not overlapped, or in global reset release mode)

Table 13: Exposure Start Delays

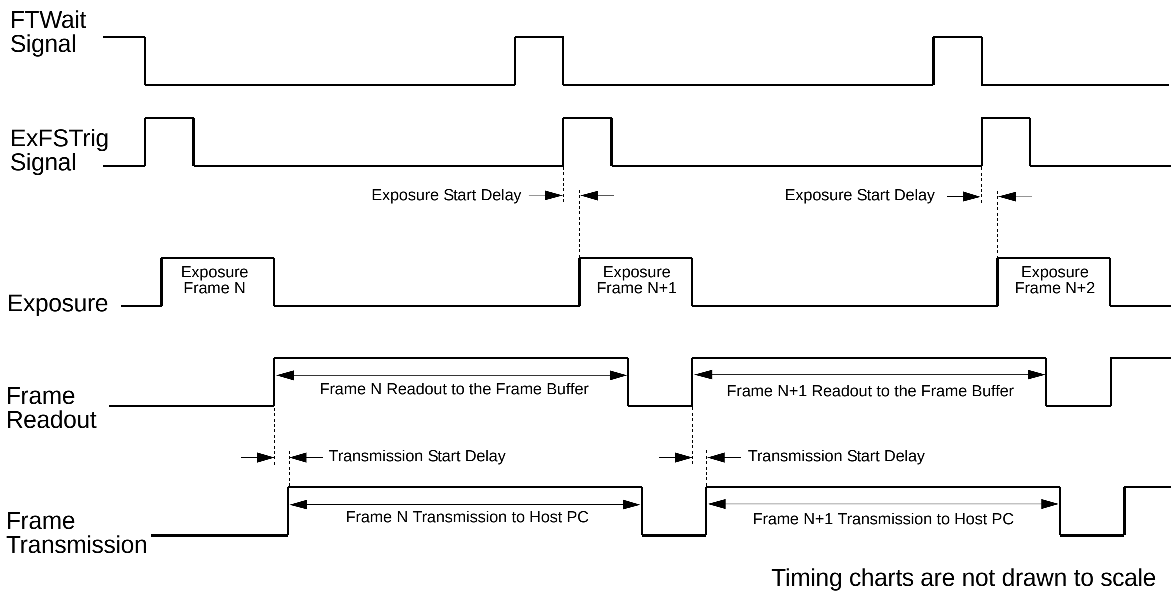


Fig. 76: Exposure Start Controlled with an ExFSTrig Signal

Note that you may have to add additional delays to the exposure start delay:

- If you use a hardware signal to trigger image acquisition, you must add a delay due to the input line response time. Note that such delays are associated with the acquisition start trigger signal and the frame start trigger signal.
  For more information about the input line response time, see Section 5.7.3 on page 60.
- If you use the debouncer feature, you must add the delay due to the debouncer setting.
  For more information about the debouncer feature, see Section 6.1.2 on page 68.
- If you have set a frame start trigger delay, you must add the delay due to the frame start trigger delay setting.
  For more information about the frame start trigger delay, see Section 7.4.3.3 on page 101.

For example, assume that you are using an acA640-100 camera and that you have set the camera for hardware triggering. Also assume that you have selected input line 1 to accept the hardware trigger signal, that the input line response time is 1.5 µs, that the delay due to the debouncer setting for input line 1 is 5 µs, and that you set the frame start trigger delay to 200 µs.

In this case:

Total Start Delay =

= Exposure Start Delay (Table 13) + Input Line Response time + Debouncer Setting + Frame Start Trigger Delay

Total Start Delay = 17.62 µs + 1.5 µs + 5 µs + 200 µs = 224.12 µs

You can determine the readout time by reading the value of the Readout Time Abs parameter. The parameter indicates what the readout time will be in microseconds given the camera's current settings. You can read the Readout Time Abs parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to get the parameter value:

```
double ReadoutTime = Camera.ReadoutTimeAbs.GetValue( );
```

You can also use the Basler pylon Viewer application to easily get the parameter value.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

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 Appendix B on page 349.

7.12 Maximum Allowed Frame Rate

In general, the maximum allowed acquisition frame rate on any ace camera 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.

On acA750-30 cameras, an additional factor is involved:

- The Field Output Mode parameter setting. If a camera is set for the Field 0 or the Field 1 mode, it can output approximately twice as many frames as it can with the camera set for the Concatenated New Fields or the Deinterlaced New Fields output mode.

On acA2000-50, acA2040-25 cameras, an additional factor is involved:

The stacked zone imaging feature:
Using the stacked zone imaging feature increases the camera’s frame rate.
For more information on the stacked zone imaging feature, see Section 10.6 on page 228.

There are two ways that you can determine the maximum allowed acquisition frame rate with your current camera settings:

- you can use the online frame rate calculator found in the Support section of our website: www.baslerweb.com
- You can use the Basler pylon API to read the value of the camera’s Resulting Frame Rate Abs parameter (see the next page).

For more information about AOI Height settings, see Section 10.5 on page 225.

For more information about the field output modes on acA750-30 cameras, see Section 7.5 on page 103.

ⓘ	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. To achieve the maximum possible acquisition frame rate, set the camera for the continuous acquisition mode and use "overlapped" exposure. For more information about overlapped exposure, see Section 7.11 on page 140.

7.12.1 Using Basler pylon to Check the Maximum Allowed Frame Rate

You can use the Basler pylon API to read the current value of the Resulting Frame Rate Abs parameter from within your application software using the Basler pylon API. The following code snippet illustrates using the API to get the parameter value:

```
// Get the resulting frame rate
double resultingFps = Camera.ResultingFrameRateAbs.GetValue();
```

The Resulting Frame Rate Abs parameter takes all camera settings that can influence the frame rate into account and indicates the maximum allowed frame rate given the current settings.

You can also use the Basler pylon Viewer application to easily read the parameter.

For more information about the pylon API and pylon Viewer, see Section 3 on page 39.

7.12.2 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 adjust one or more of the factors that can influence the maximum allowed rate and then check to see if the maximum allowed rate has increased:

- Decreasing the height of the AOI can have a significant impact on the maximum allowed frame rate. If possible in your application, decrease the height of the AOI.
- If you are using normal exposure times and you are using the camera at it's maximum resolution, your exposure time will not normally restrict the frame rate. However, if you are using long exposure times or small areas of interest, it is possible that your exposure time is limiting the maximum allowed frame rate. If you are using a long exposure time or a small AOI, try using a shorter exposure time and see if the maximum allowed frame rate increases. (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 restrict the frame rate. 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.

If you are working with an **acA2500-14** camera:

Use the normal shutter mode rather than the global reset release shutter mode. Because the normal shutter mode allows frame acquisitions to be overlapped and the global reset release mode does not allow overlapping, you will be able to achieve a higher frame rate when using the normal shutter mode.

If you are working with an **acA750-30** camera:

Use the Field 0 or the Field 1 field output mode instead of the Concatenated New Fields or the Deinterlaced New Fields field output mode. With the Field 0 or the Field 1 modes, you can get approximately twice the frame rate, but you will be getting half height frames.

If you are working with an **acA2000-50** or **acA2040-25** camera:

Using the stacked zone imaging feature increases the camera's frame rate.

For more information on the stacked zone imaging feature, see Section 10.6 on page 228.

> An important thing to keep in mind is a common mistake new camera users frequently make when they are working with exposure time. They will often use a very long exposure time without realizing that this can severely limit the camera's maximum allowed frame rate. As an example, assume that your camera is set to use a 1/2 second exposure time. In this case, because each frame acquisition will take at least 1/2 second to be completed, the camera will only be able to acquire a maximum of two frames per second. Even if the camera's nominal maximum frame rate is, for example, 100 frames per second, it will only be able to acquire two frames per second because the exposure time is set much higher than normal.

For more information about AOI settings, see Section 10.5 on page 225.

For more information about the packet size and inter-packet delay settings and about the settings that determine the bandwidth assigned to the camera, see Appendix B on page 349.

7.12.3 Removing the Frame Rate Limit (acA640-100 Only)

Normally, the maximum frame rate that an acA640-100 camera can achieve with a given group of parameter settings is as described in the previous section. In this normal situation, the maximum frame rate is limited by the standard operating ranges of several of the electronic components used in the camera. The goal of remaining within these standard operating ranges is to ensure that the camera provides optimum image quality.

If you desire, you can use the remove parameter limits feature to remove the maximum frame rate limit on your acA640-100 camera. If you remove the frame rate limit, the electronic components will be allowed to operate outside of their normal operating ranges. With the limit removed, you will find that the maximum allowed frame rate at full resolution will increase and that the maximum allowed frame rate with smaller AOI settings will also increase proportionately.

If you do remove the maximum frame rate limit, you may see some degradation in the overall image quality. In many applications, however, the benefits of an increase in the maximum allowed frame rate will outweigh the drawbacks of a marginal decrease in image quality.

To determine how much removing the frame rate limit will affect the maximum allowed frame rate with your current camera settings:

- Read the value of the Resulting Frame rate parameter with the maximum frame rate limit enabled.
- Use the remove parameter limits feature to remove the limit.
- Read the value of the Resulting Frame rate parameter with the limit removed.

For more information about using the remove parameter limits feature, see Section 10.14 on page 301.

For more information about the Resulting Frame Rate parameter, see page 143.

7.13 Use Case Descriptions and Diagrams

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.

> 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)

Use case one is illustrated on page 148.

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.

For more information about the Acquisition Frame Rate Abs parameter, see Section 7.3.1.1 on page 85.

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

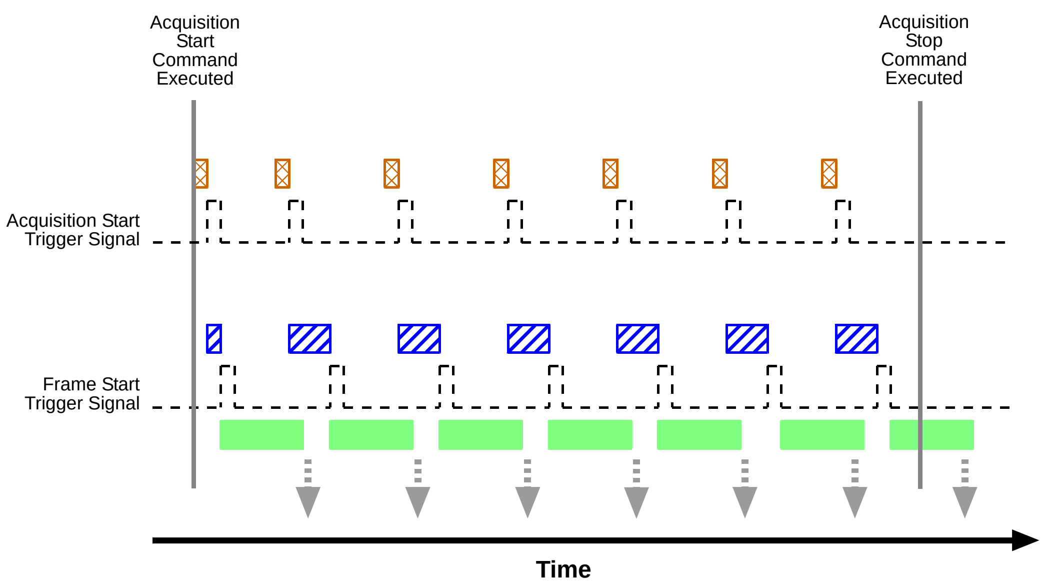


Fig. 77: Use Case 1 - Acquisition Start Trigger Off and Frame Start Trigger Off

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

Use case two is illustrated on page 150.

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

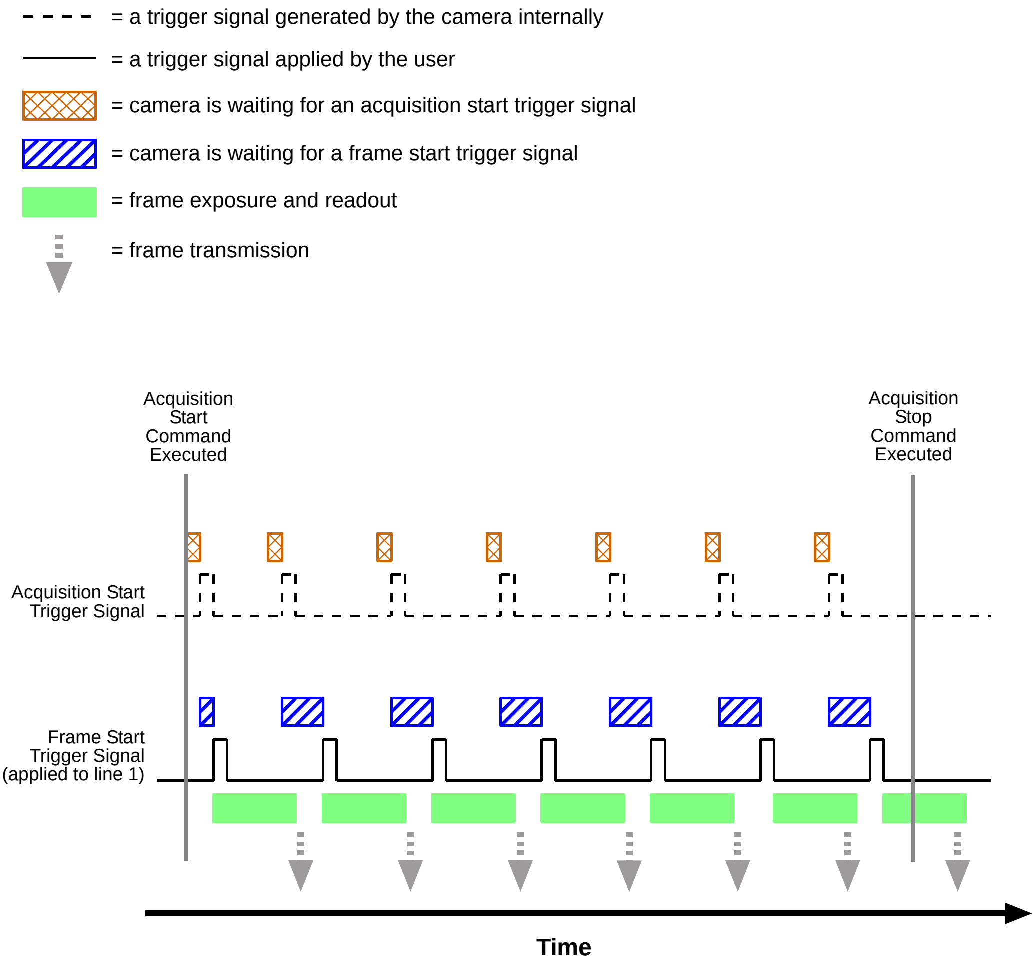


Fig. 78: Use Case 2 - Acquisition Start Trigger Off and Frame Start Trigger On

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

Use case three is illustrated on page 152.

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 7.3.1.1 on page 85.

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

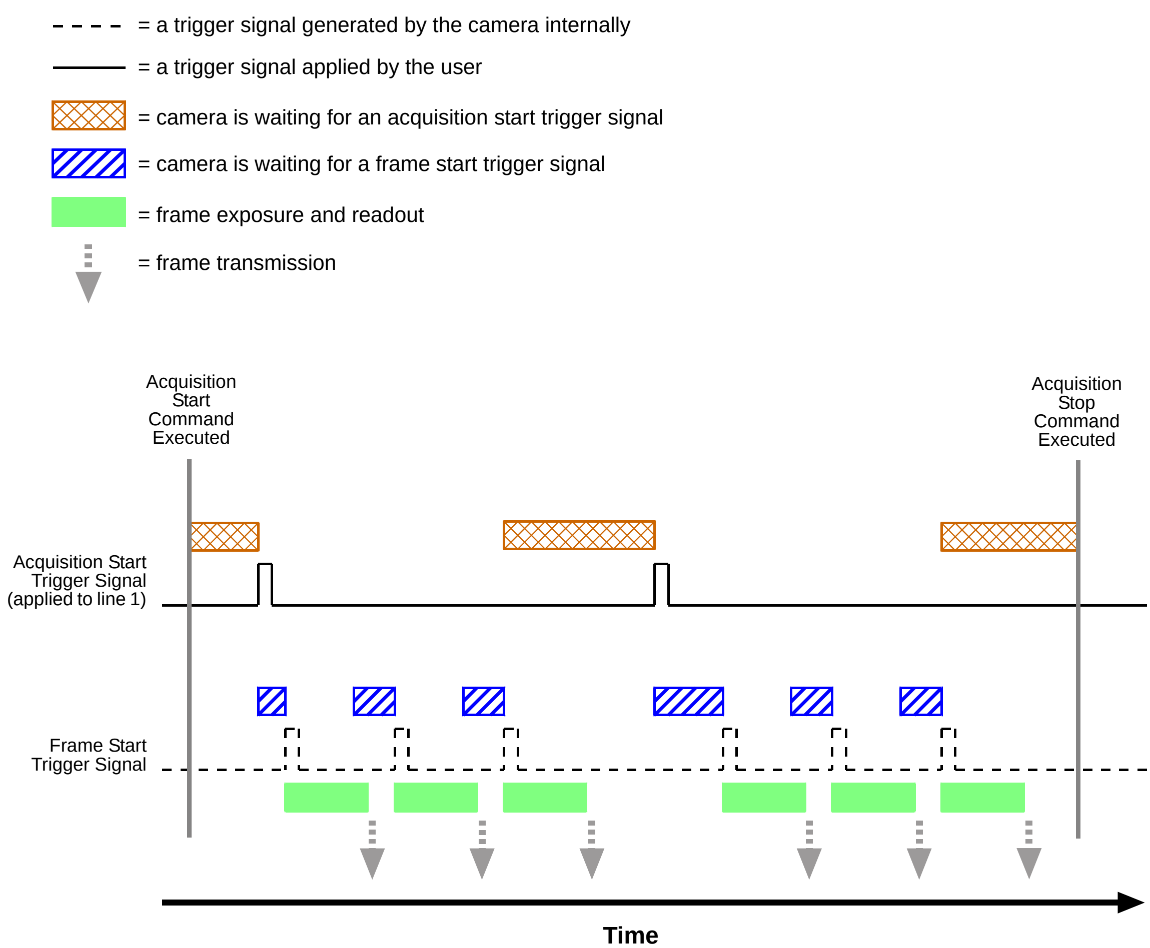


Fig. 79: Use Case 3 - Acquisition Start Trigger On and Frame Start Trigger Off

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

Use case four is illustrated on page 154.

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

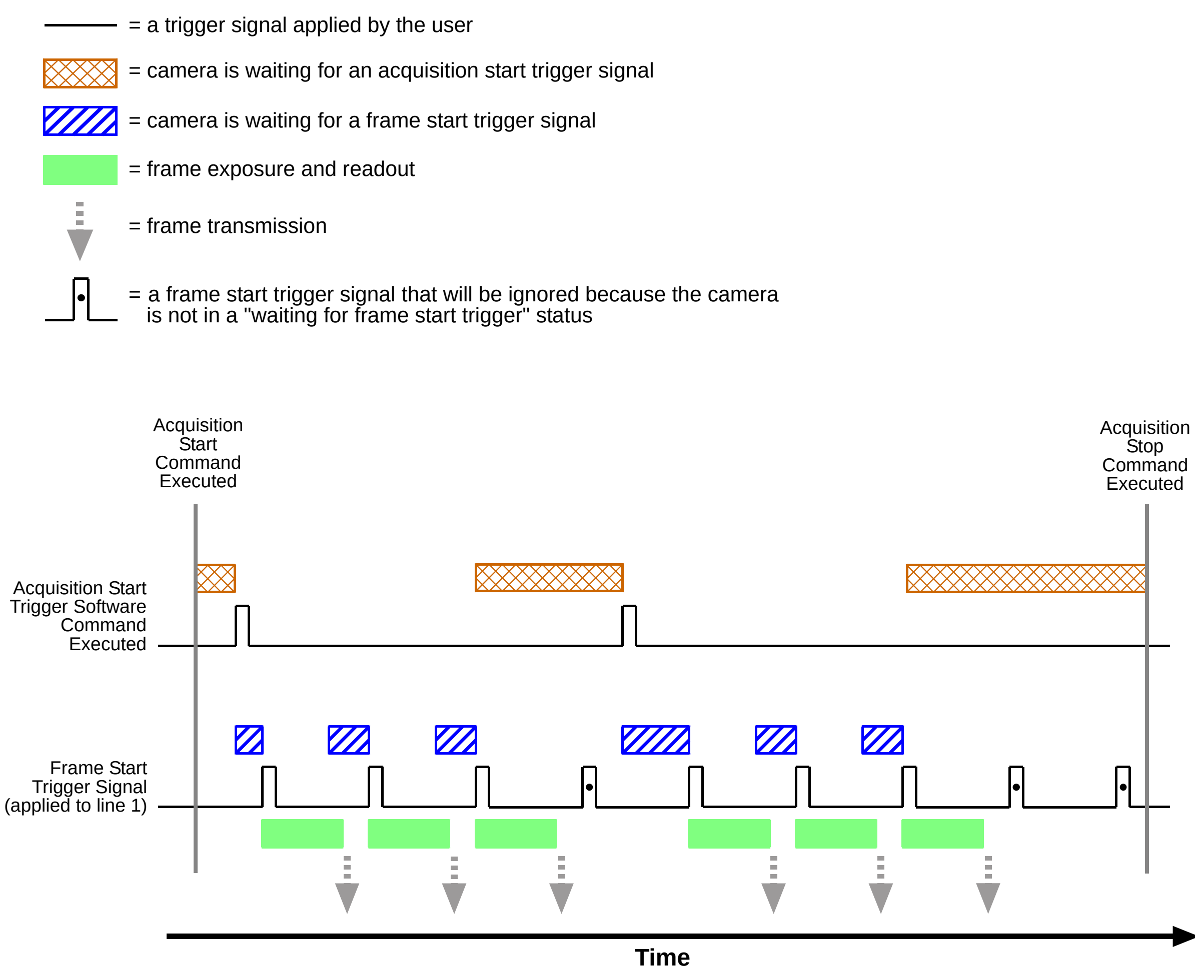


Fig. 80: Use Case 4 - Acquisition Start Trigger On and Frame Start Trigger On

8 Color Creation and Enhancement

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

8.1 Color Creation (All Color Models Except the acA750-30gc)

The sensors used in these cameras are 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 part of the filter 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 81 (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.)

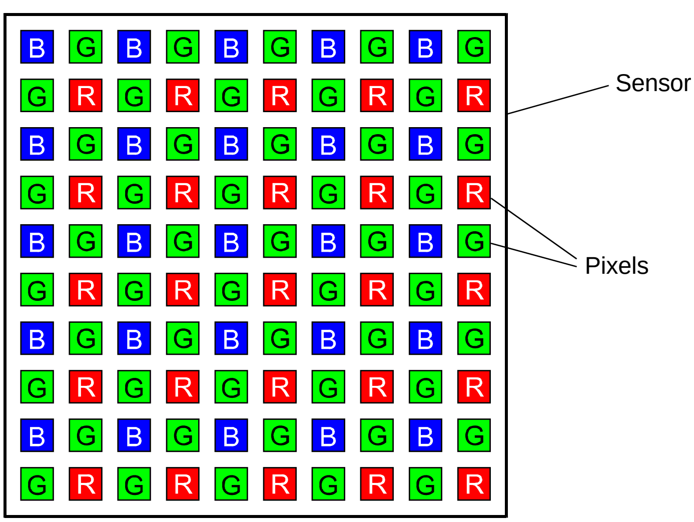


Fig. 81: Bayer Filter Pattern

8.1.1 Bayer Color Filter Alignment

The alignment of the Bayer filter to the pixels in the images acquired by color cameras depends on the camera model. Table 14 shows the filter alignment for each available camera model.

Color Camera Model	Filter Alignment
acA640-90	BG
acA640-100	BG
acA645-100	BG
acA780-75	BG
acA1300-30	BG
acA1600-20	BG
acA2000-50	GR
acA2040-25	GR
acA2500-14	BG

Table 14: Bayer Filter to Sensor Alignment

On all color camera models that have sensors equipped with a Bayer filter, the alignment of the filter to the pixels in the acquired images is Bayer BG or Bayer GR (see Table 14).
Bayer BG alignment, for example, 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.

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.

Because the size and position of the area of interest on color cameras with a Bayer filter must be adjusted in increments of 2, the color filter alignment will remain as Bayer BG or Bayer GR regardless of the camera's area of interest (AOI) settings.

For more information about the camera's AOI feature, see Section 10.5 on page 225.

8.1.2 Pixel Data Formats Available on Cameras with a Bayer Filter

Bayer Formats

Depending on the camera model the cameras equipped with a Bayer pattern color filter can output pixel data in the following formats:

- For all cameras except the acA2000-50gc and acA2040-25gc:
  Bayer BG 8, Bayer BG 12, or Bayer BG 12 Packed
- For the acA2000-50gc and acA2040-25gc only:
  Bayer GR 8, Bayer GR 12, or Bayer GR 12 Packed

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 three pixel data output formats, see Section 9.1 on page 183 and Section 9.3 on page 191.

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 9.3 on page 191.

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 9.3 on page 191.

8.2 Color Creation on the acA750-30gc

The sensor used in this camera is equipped with a complementary plus green color separation filter. The colors in the filter are cyan, magenta, yellow, and green (CMYeG). Each individual pixel is covered by a portion of the filter that allows light of only one color to strike the pixel. The filter has a repeating pattern as shown in Figure 82.

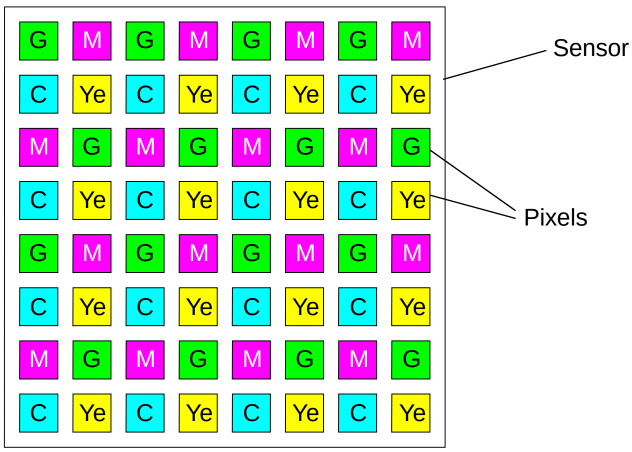


Fig. 82: Complementary Color Filter Pattern

Because there is only one vertical shift register for every two pixels in the camera's sensor, when a field is acquired, the colors from two pixels will be combined into a single "binned" pixel. As shown in Figure 83, when the camera acquires field 0, it will obtain the following color combinations for any group of four "binned" pixels:

Green + Cyan

Magenta + Cyan

Magenta + Yellow

Green + Yellow

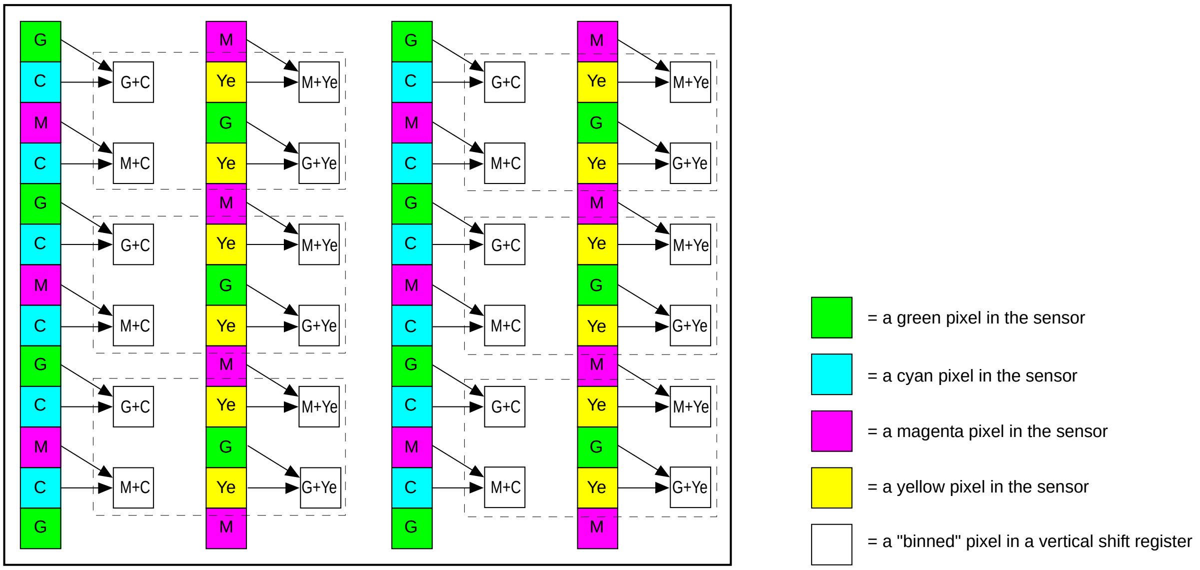


Fig. 83: Color Combinations for Field 0

As shown in Figure 84, when the camera acquires field 1, it will obtain the following color combinations for any group of four binned pixels:

Magenta + Cyan

Green + Cyan

Yellow + Green

Yellow + Magenta

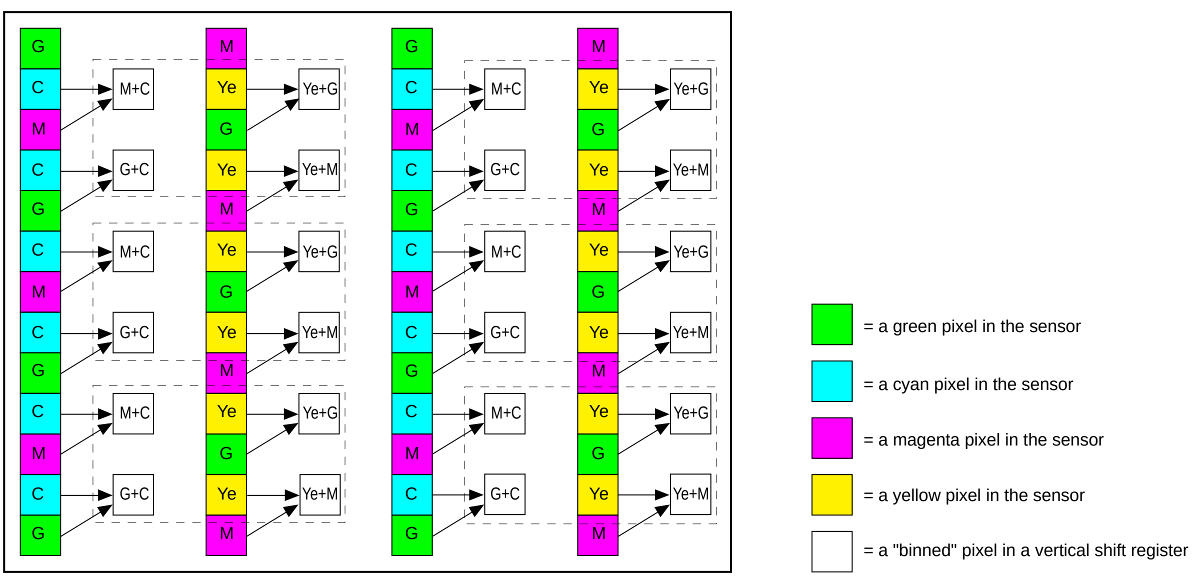


Fig. 84: Color Combinations for Field 1

If you compare the color combinations in the binned pixels for field 0 with the color combinations for the binned pixels in field 1, you will see that they are equivalent. The pattern of the colors in the complementary filter was designed specifically to make this possible, and it means that the color information can be manipulated in an identical fashion regardless of whether the camera is working with pixel values from field 0 or from field 1.

Preparing the combined color data in the binned pixels of an acquired field for transmission from the camera is a several step process:

- The CMYeG sensor colors are converted into a YUV color signal.
- A matrix color transformation is performed on the YUV color information to obtain full RGB color information for each binned pixel.
- If the camera's white balance feature is used, it will act on the RGB information for each binned pixel.
- If the camera's color adjustment feature is used, it will act on the RGB information for each binned pixel.
- If the camera's gamma correction feature is used, it will act on the RGB information for each binned pixel.
- A final transformation is performed on the RGB color information to convert it to YUV information for each binned pixel.
- The binned pixel values are transmitted from the camera in a YUV format.

8.2.1 Pixel Data Formats Available on Cameras with a CMYeG Filter

YUV Formats

On a color camera equipped with a CMYeG filter, the pixel values go through several conversion steps. This process yields Y, U, and V color information for the pixels.

These cameras can then output color pixel data in a YUV 4:2:2 Packed format or in a YUV 4:2:2 (YUYV) Packed format.

For complete details of the YUV data output formats, see Section 9.3 on page 191.

Mono Format

On cameras equipped with a CMYeG color filter, the pixel values are converted to the YUV color model as described earlier. The camera can then output pixel data in the Mono 8 format.

When a color camera is set for Mono 8, the 8 bit Y value for each pixel is transmitted 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 9.3 on page 191.

8.3 Integrated IR Cut Filter

All color camera models are equipped with an IR-cut filter as standard equipment. The filter is mounted in a filter holder located in the lens mount.

Monochrome cameras include a filter holder in the lens mount, but the holder is not populated with an IR-cut filter.

NOTICE
On all cameras, the lens thread length is limited. All cameras (mono and color) are equipped with a plastic filter holder located in the lens mount. The location of the filter holder 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 filter holder or the lens mount 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 1.4.2 on page 27.

8.4 Color Enhancement Features

8.4.1 White Balance

On all color cameras equipped with a Bayer pattern filter (i.e., all camera models except the acA750-30gc) the pixel values output from the sensor reside in the RGB color space.

On the acA750-30gc camera model, the pixel values output from the sensor are first converted to YUV and are then converted to the RGB color space.

The white balancing feature implemented in the camera acts on the colors when they are in the RGB color space, so the feature lets you perform red, green, and blue adjustments. The purpose of the feature is to let 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.

If color binning is enabled for the acA2500-14gc, white balancing will be applied after color binning was performed. For more information about color binning, see Section 10.8.2 on page 268.

Setting the White Balance

This section (Section 8.4.1) 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 10.12 on page 282.

For more information about the Balance White Auto function, see Section 10.12.8 on page 297.

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 you selected.

For more information about matrix color transformation, see Section 8.4.3 on page 168 and Section 8.4.4 on page 172.

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 Abs parameter is used to set the intensity of the color. If the Balance Ratio Abs 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 Abs for a color is set to 1.2, the intensity of that color will be increased by 20%.

The Balance Ratio Abs parameter value can range from 0.00 to 15.9844. 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 Abs 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 );
```

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 page 39.

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 page 39.

For more information about the Light Source Selector parameter setting, see Section 8.4.3 on page 168 or Section 8.4.4 on page 172.

8.4.2 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.

ⓘ	If color binning is enabled for the acA2500-14gc, gamma correction will be applied after color binning was performed. For more information about color binning, see Section 10.8.2 on page 268.

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 3.99998.

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 page 39.

8.4.3 Matrix Color Transformation on All Color Models Except the acA750-30gc

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.

ⓘ	If color binning is enabled for the acA2500-14gc, matrix color transformation will be applied after color binning was performed. For more information about color binning, see Section 10.8.2 on page 268.

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 the Bayer xx format, 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 a Bayer xx pixel data output format 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 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.

ⓘ	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 page 39.

8.4.3.1 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/2437.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.

8.4.4 Matrix Color Transformation on acA750-30gc Cameras

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 any imperfections in the sensor’s color generation process.

On this camera model, the pixel values output by the camera’s imaging sensor undergo a several step process before being transmitted by the camera:

- In the first step, the pixel values from the sensor are converted into a YUV color signal.
- In the second step, a first matrix transforrmation step converts the Y, U, and V components for very binned pixel to R, G, and B components and another 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. (For information about binned pixels, see the "Color Creation on the acA750-30gc" section).
- When the pixel values are in the RGB color space, gamma and white balance correction can be applied using the features described earlier in this chapter, and hue and saturation can be adjusted using the feature described later in this chapter.
- Finally, the pixel values are converted back to the YUV color space and transmitted from the camera.

Matrix Color Transformation Parameters

The first camera parameter associated with matrix color transformation is the Color Transformation Selector parameter. This parameter is used to select the type of transformation that will be performed. For acA750gc cameras, YUV to RGB is the only setting available.

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

- 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 the setting that populates 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 each 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.

> When the Light Source Selector parameter is set to custom, the Color Transformation Matrix Factor parameter will not be available.

Setting Matrix Transformation

You can set the Color Transformation Selector and Light Source Selector parameters from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values:

```
// Select the color transformation type
Camera.ColorTransformationSelector.SetValue
     ( ColorTransformationSelector_YUVtoRGB );

// 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 page 39.

8.4.4.1 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 YUV to RGB color matrix conversion is performed by multiplying a 1 x 3 matrix containing the Y, U, and V color values for a pixel with a 3 x 3 matrix containing correction values. In the 3 x 3 matrix, the first column is populated by values of 1.0 and cannot be changed. The second and third columns can be populated with values of your choice. In other words:

$$\begin{bmatrix} 1.0 & \text{Gain01} & \text{Gain02} \\ 1.0 & \text{Gain11} & \text{Gain12} \\ 1.0 & \text{Gain21} & \text{Gain22} \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Where Gain01, Gain12, etc. are settable values.

Each GainXY position can each 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/2437.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.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

8.4.5 Color Adjustment

On all color cameras equipped with a Bayer pattern filter (i.e., all camera models except the acA750-30gc) the pixel values output from the sensor reside in the RGB color space.

On the acA750-30gc camera model, the pixel values output from the sensor are first converted to YUV and are then converted to the RBG color space.

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 8.4.1 on page 164 for more information about the white balance and see Section 8.4.6 on page 181 for an 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 8.4.3 on page 168 and Section 8.4.4 on page 172 for more information about color matrix transformation.

> If color binning is enabled for the acA2500-14gc, color adjustment will be applied after color binning was performed. For more information about color binning, see Section 10.8.2 on page 268.

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 85 on page 177) 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 85) 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.

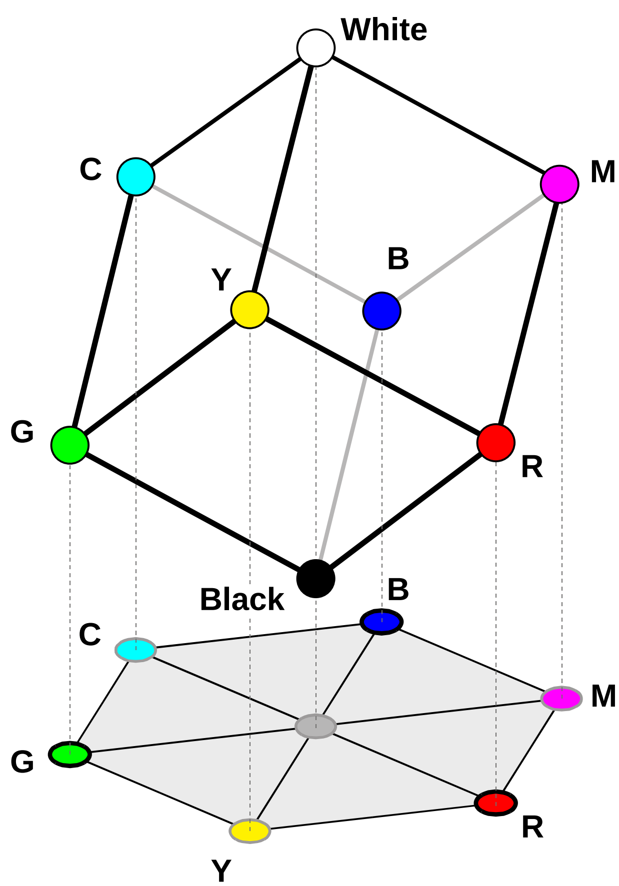


Fig. 85: RGB Color Cube With YCM Secondary Colors, Black, and White, Projected On a Plane

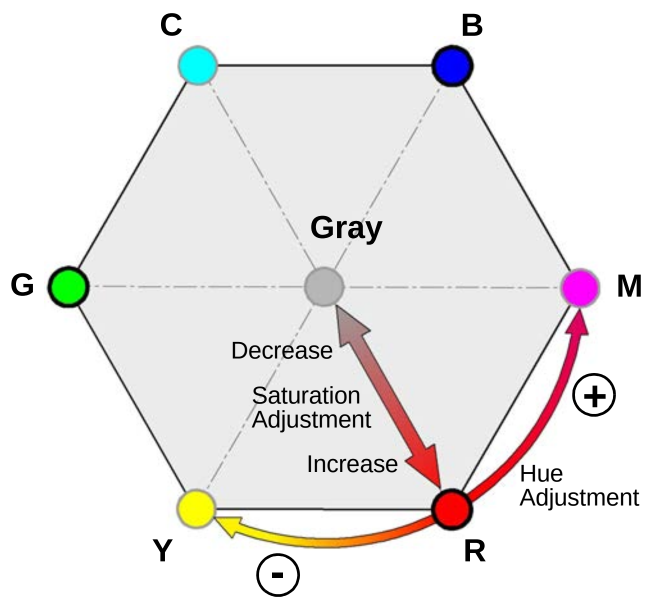


Fig. 86: Hue and Saturation Adjustment In the Color Hexagon. Adjustments Are Indicated for Red as an Example

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 a Bayer xx format, 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 a Bayer xx pixel data output format 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.

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 BG8 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 page 39.

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 page 39.

8.4.6 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 180).

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.

8.4.7 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 10.18 on page 311.

9 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.

9.1 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 15 lists the pixel formats available on each monochrome camera model and Table 16 lists the pixel formats available on each color camera model.

Mono Camera Model	Mono 8	Mono 12	Mono 12 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
acA640-90gm	●	●	●	●	●
acA640-100gm	●	●	●	●	●
acA645-100gm	●	●	●	●	●
acA750-30gm	●	●	●	●	●
acA780-75gm	●	●	●	●	●
acA1300-30gm	●	●	●	●	●
acA1600-20gm	●	●	●	●	●
acA2000-50gm	●	●	●	●	●
acA2000-50gmNIR	●	●	●	●	●
acA2040-25gm	●	●	●	●	●
acA2040-25gmNIR	●	●	●	●	●
acA2500-14gm	●	●	●	●	●

Table 15: Pixel Formats Available on Monochrome Cameras (• = format available)

Color Camera Model	Mono 8	Bayer GR 8	Bayer BG 8	Bayer GR 12	Bayer BG 12	Bayer GR 12 Packed	Bayer BG 12 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
acA640-90gc	●		●		●		●	●	●
acA640-100gc	●		●		●		●	●	●
acA645-100gc	●		●		●		●	●	●
acA750-30gc	●							●	●
acA780-75gc	●		●		●		●	●	●
acA1300-30gc	●		●		●		●	●	●
acA1600-20gc	●		●		●		●	●	●
acA2000-50gc	●	●		●		●		●	●
acA2040-25gc	●	●		●		●		●	●
acA2500-14gc	●		●		●		●	●	●

Table 16: Pixel Formats Available on Color Cameras (• = format available)

Details of the monochrome formats are described in Section 9.2 on page 185 and details of the color formats are described in Section 9.3 on page 191.

You can set the Pixel Format 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.PixelFormat.SetValue( PixelFormat_Mono8 );
Camera.PixelFormat.SetValue( PixelFormat_Mono12Packed );
Camera.PixelFormat.SetValue( PixelFormat_Mono12 );
Camera.PixelFormat.SetValue( PixelFormat_YUV422Packed );
Camera.PixelFormat.SetValue( PixelFormat_YUV422_YUYV_Packed );
Camera.PixelFormat.SetValue( PixelFormat_BayerBG8 );
Camera.PixelFormat.SetValue( PixelFormat_BayerBG12 );
Camera.PixelFormat.SetValue( PixelFormat_BayerBG12Packed );
Camera.PixelFormat.SetValue( PixelFormat_BayerGR8 );
Camera.PixelFormat.SetValue( PixelFormat_BayerGR12 );
Camera.PixelFormat.SetValue( PixelFormat_BayerGR12Packed );
```

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 page 39.

9.2 Pixel Data Formats for Mono Cameras

9.2.1 Mono 8 Format

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

9.2.2 Mono 12 Format

When a monochrome camera is set for the Mono12 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 Mono12 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 12, 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 Mono12 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

9.2.3 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

9.2.4 YUV 4:2:2 Packed Format

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 9.3.7 on page 202.

9.2.5 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 9.3.8 on page 204.

9.3 Pixel Data Output Formats for Color Cameras

9.3.1 Bayer BG 8 Format

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 filter, you get 8 bits of red data. For each pixel covered with a green filter, you get 8 bits of green data. And for each pixel covered with a blue filter, 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 8.1 on page 155.

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

9.3.2 Bayer GR 8 Format

When a color camera is set for the Bayer GR 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 filter, you get 8 bits of red data. For each pixel covered with a green filter, you get 8 bits of green data. And for each pixel covered with a blue filter, you get 8 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "GR" in the name Bayer GR 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 green, pixel two will be red, pixel three will be green, pixel four will be red, etc. For odd 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 more information about the Bayer filter, see Section 8.1 on page 155.

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 GR 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	Green value for P_0
B_1	Red value for P_1
B_2	Green value for P_2
B_3	Red value for P_3
B_4	Green value for P_4
B_5	Red value for P_5
2	•
2	•
2	•
B_{m-5}	Green value for P_{n-5}
B_{m-4}	Red value for P_{n-4}
B_{m-3}	Green value for P_{n-3}
B_{m-2}	Red value for P_{n-2}
B_{m-1}	Green value for P_{n-1}
B_m	Red value for P_n

Odd 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}
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

With the camera set for Bayer GR 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

9.3.3 Bayer BG 12 Format

When a color camera is set for the Bayer BG 12 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 12 the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red filter, you get 12 effective bits of red data. For each pixel covered with a green filter, you get 12 effective bits of green data. And for each pixel covered with a blue filter, 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 12 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 8.1 on page 155.

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 12 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 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	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}

Odd Rows	
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 blue 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-6}	High byte of green value for P_{n-3}
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 12, 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 12 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

ⓘ A camera that is set for Bayer BG 12 has only 12 effective bits out of the 16 bits transmitted for each pixel. The leader of each transmitted frame will indicate Bayer BG12 as the pixel format.

9.3.4 Bayer GR 12 Format

When a color camera is set for the Bayer GR 12 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 GR 12 the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red filter, you get 12 effective bits of red data. For each pixel covered with a green filter, you get 12 effective bits of green data. And for each pixel covered with a blue filter, you get 12 effective bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "GR" in the name Bayer GR 12 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 green, pixel two will be red, pixel three will be green, pixel four will be red, etc. For odd 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 more information about the Bayer filter, see Section 8.1 on page 155.

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 GR 12 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 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	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 blue value for P_{n-3}
B_{m-6}	High byte of blue value for P_{n-3}
B_{m-5}	Low byte of green value for P_{n-2}

Odd Rows	
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}
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-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

When the camera is set for Bayer GR 12, 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 GR 12 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

ⓘ A camera that is set for Bayer GR 12 has only 12 effective bits out of the 16 bits transmitted for each pixel. The leader of each transmitted frame will indicate Bayer GR as the pixel format.

9.3.5 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 filter, you get 12 bits of red data. For each pixel covered with a green filter, you get 12 bits of green data. And for each pixel covered with a blue 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 8.1 on page 155.

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

9.3.6 Bayer GR 12 Packed Format

When a color camera is set for the Bayer GR 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 GR 12 Packed coding, the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red filter, you get 12 bits of red data. For each pixel covered with a green filter, you get 12 bits of green data. And for each pixel covered with a blue 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 8.1 on page 155.

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 GR12 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	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	

Odd 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	

When a color camera is set for Bayer GR 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

9.3.7 YUV 4:2:2 Packed Format

When a color camera is set for the YUV 422 Packed pixel data format, each pixel value in the captured image goes through a 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 value.

For more information about the conversion processes, see Section 8 on page 155.

ⓘ	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.

9.3.8 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.

9.3.9 Mono 8 Format

When a color camera is set for the Mono 8 pixel data format, the values for each pixel are first converted to the YUV color model. 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. In the color camera, however, the Y component is derived from brightness values of the pixel and neighboring pixels. 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

9.4 Pixel Transmission Sequence

For each captured image, pixel data is transmitted from the camera in the sequence given below. The sequence assumes that the camera is set for full resolution:

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 pixel transmission sequence described above does not adequately describe the behavior of acA750-30 cameras. For more information about how the acA750-30 differs, see Section 7.5 on page 103.

10 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.

10.1 Gain

The camera's gain setting is adjustable. As shown in Figure 87, increasing the gain increases the slope of the response curve for the camera. This results in a higher gray value output from the camera for a given amount of output from the imaging sensor. Decreasing the gain decreases the slope of the response curve and results in a lower gray value for a given amount of sensor output.

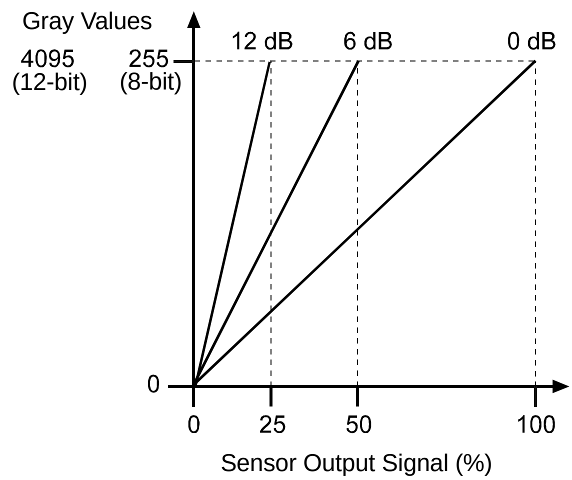


Fig. 87: Gain in dB

Increasing the gain is useful when at your brightest exposure, a gray value lower than 255 (in modes that output 8 bits per pixel) or 4095 (in modes that output 12 bits per pixels) is reached. For example, if you found that at your brightest exposure the gray values output by the camera were no higher than 127 (in an 8 bit mode), you could increase the gain to 6 dB (an amplification factor of 2) and thus reach gray values of 254.

10.1.1 Setting the Gain

This section (Section 10.1) describes how gain can be adjusted "manually", i.e., by setting the value of the Gain Raw parameter.

The camera also has a Gain Auto function that can automatically adjust the gain. **Manual adjustment of the Gain Raw parameter will only work correctly, if the Gain Auto function is disabled.**

For more information about auto functions in general, see Section 10.12 on page 282.

For more information about the Gain Auto function, see Section 10.12.4 on page 291.

All Models Except the acA2000-50, the acA2040-25, and the acA2500-14

The camera's gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on an integer scale. The minimum setting varies depending on the camera model and on whether vertical binning is enabled (see Table 17). The maximum setting depends on whether the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, Bayer BG 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed) or yields an effective pixel depth of 12 bits per pixel (Mono 12, Mono 12 Packed, Bayer BG 12, Bayer BG 12 Packed).

Camera Model	Min Setting	Min Setting with Vertical Binning (mono cameras)	Max Setting (8 bit depth)	Max Setting (12 bit depth)
acA640-90gm/gc	190	100	1023	400
acA640-100gm/gc	100	100	1023	600
acA645-100gm/gc	200	140	1023	400
acA750-30gm/gc	0	NA	1023	600
acA780-75gm/gc	250	140	1023	400
acA1300-30gm/gc	300	200	850	400
acA1600-20gm/gc	230	220	850	400

Table 17: 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 Basler 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 );
```

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 page 39.

On acA640-90gm/gc, acA640-100gm/gc, acA645-100gm/gc, acA780-75gm/gc, acA1300-30gm/gc, and on acA1600-20gm/gc cameras, the minimum setting for the Gain Raw parameter can be reduced to 0 by using the remove parameter limits feature.

For more information about the remove parameter limits feature, see Section 10.3 on page 217.

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:

$Gain_{db}$ = 0.0359 x Gain Raw Setting

Example:

Camera has a gain raw setting of 200. Gain calculation:

$Gain_{db}$ = 0.0359 x 200

$Gain_{db}$ = 7.2

Table 18 shows the minimum and maximum possible dB of gain for each camera model.

Camera Model	db Gain at Min Setting	db Gain at Max Setting (8 bit depth)	db Gain at Max Setting (12 bit depth)
acA640-90gm/gc	6.8	36.7	14.4
acA640-100gm/gc	3.6	36.7	21.5
acA645-100gm/gc	7.2	36.7	14.4
acA750-30gm/gc	0	36.7	21.5
acA780-75gm/gc	9.0	36.7	14.4
acA1300-30gm/gc	10.8	30.5	14.4
acA1600-20gm/gc	8.3	30.5	14.4

Table 18: Minimum and Maximum dB of Gain

acA2000-50 and acA2040-25 Only

The camera's gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on an integer scale. The minimum setting varies depending on the camera model and on whether vertical binning is enabled (see Table 19). 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 xx 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed) or yields an effective pixel depth of 12 bits per pixel (Mono 12, Mono 12 Packed, Bayer xx 12, Bayer xx 12 Packed).

Camera Model	Min Setting	Min Setting with Vertical Binning (mono cameras)	Max Setting (8 bit depth)	Max Setting (12 bit depth)
acA2000-50gm/gc	36	33	512	512
acA2000-50gmNIR	36	33	512	512
acA2040-25gm/gc	36	33	512	512
acA2040-25gmNIR	36	33	512	512

Table 19: Minimum and Maximum Allowed Gain Raw Settings (acA2000-50, acA2040-25 Only)

To set the Gain Raw parameter value using Basler pylon:

- 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 Basler 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 );
```

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 page 39.

ⓘ	On acA2000-50gm/gc, acA2000-50gmNIR, acA2040-25gm/gc and acA2040-25gmNIR, the minimum setting for the Gain Raw parameter can be reduced to 0 by using the remove parameter limits feature. For more information about the remove parameter limits feature, see Section 10.3 on page 217.

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:

Gain in dB = $20 \log_{10}$ (Gain Raw Setting / 32)

Example:

Assume that you are working with a camera that has a gain raw setting of 128. The gain is calculated as follows:

- Gain in dB = 20 $\log_{10}$ (128 / 32)
- Gain in dB = 12.0

Table 20 shows the minimum and maximum possible dB of gain for each camera model.

Camera Model	db Gain at Min Setting	db Gain at Max Setting (8 bit depth)	db Gain at Max Setting (12 bit depth)
acA2000-50gm/gc	1.02	24	24
acA2000-50gmNIR	1.02	24	24
acA2040-25gm/gc	1.02	24	24
acA2040-25gmNIR	1.02	24	24

Table 20: Minimum and Maximum dB of Gain (acA2000-50, acA2040-25 Only)

acA2500-14 Only

The camera’s gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on an integer scale. The minimum setting is 0 and the maximum setting is 63.

At a setting of 0, the camera’s gain will be 0 dB. At a setting of 63, the gain is approximately 26 dB

The range of integer settings does not map linearly to the dB gain range. The graph in Figure 88 shows the gain in dB that will be yielded for each Gain Raw parameter setting.

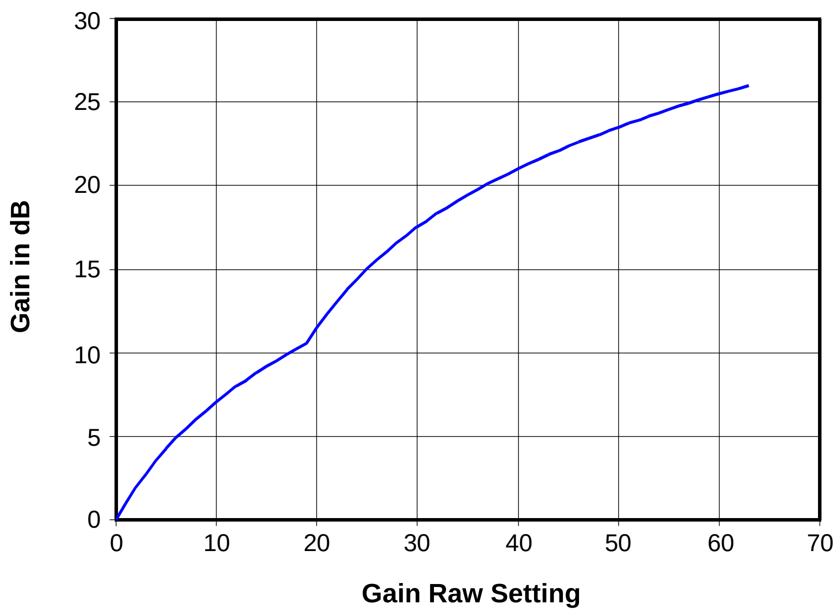


Fig. 88: Gain in dB Yielded by Gain Raw Settings

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 Basler 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( 40 );
```

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 page 39.

10.2 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.

All Models (Except acA2000-50, acA2040-25, and acA2500-14)

If the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, Bayer BG 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed), 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 12, Mono 12 Packed, Bayer BG 12, 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.

acA2000-50, acA2040-25, and acA2500-14 Only

If the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, Bayer xx 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed), an increase of 16 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. And a decrease of 16 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

If the camera is set for a pixel data format that yields an effective pixel depth of 12 bits per pixel (Mono 12, Mono 12 Packed, Bayer xx 12, Bayer xx 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.

10.2.1 Setting the Black Level

The black level can be adjusted by changing the value of the Black Level Raw parameter.

The range of the allowed settings for the Black Level Raw parameter value varies by camera model as shown in Table 21.

Camera Model	Min Allowed Black Level Raw Setting	Max Allowed Black Level Raw Setting
acA640-90gm/gc	0	1023
acA640-100gm/gc	0	1023
acA645-100gm/gc	0	1023
acA750-30gm/gc	0	1023
acA780-75gm/gc	0	1023
acA1300-30gm/gc	0	1023
acA1600-20gm/gc	0	1023
acA2000-50gm/gc	0	255
acA2000-50gmNIR	0	255
acA2040-25gm/gc	0	255
acA2040-25gmNIR	0	255
acA2500-14gm/gc	0	63

Table 21: Black Level Raw Parameter Range

To set the Black Level Raw parameter value:

- Set the Black Level Selector to Black Level All.
- Set the Black Level Raw parameter to your desired value.

You can set the Black Level Selector and the Black Level Raw parameter value from within your application software by using the Basler 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 );
```

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 page 39.

10.3 Remove Parameter Limits

For each camera feature, the allowed range of any associated parameter values is normally 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 remove parameter limits feature lets you remove the factory limits for parameters associated with certain camera features. When the factory limits are removed, 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 any extended limits can be determined by using the Basler pylon Viewer or from within your application via the pylon API.

Currently, the limits can be removed from:

- The gain feature.

  Removing the parameter limits on the gain feature will only remove the lower limit. The lower limit for the Gain parameter is reduced to 0. (For those cameras where the lower limit is already 0, removing the limits has no effect.)
- The maximum allowed frame rate on acA640-100 cameras.

  Removing the limit on the maximum allowed frame rate will let the camera operate at a higher than normal frame rate for the current parameter settings.

For more information about the gain feature, see Section 10.1 on page 209.

For more information about the frame rate limit on acA640-100 cameras, see Section 7.12.3 on page 146.

Removing Parameter Limits

To remove the limits for a parameter:

- Use the Parameter Selector to select the parameter whose limits you want to remove.
- 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 Basler 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 removed.
Camera.ParameterSelector.SetValue( ParameterSelector_Gain );
// Remove the limits for the selected feature.
Camera.RemoveLimits.SetValue( true );

// Select the feature whose factory limits will be removed.
Camera.ParameterSelector.SetValue( ParameterSelector_Framerate );
```

```
// Remove the limits for the selected feature.
Camera.RemoveLimits.SetValue( true );
```

You can also use the Basler pylon Viewer application to easily set the parameters. Note that the remove parameter limits feature will only be available at the "guru" viewing level.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

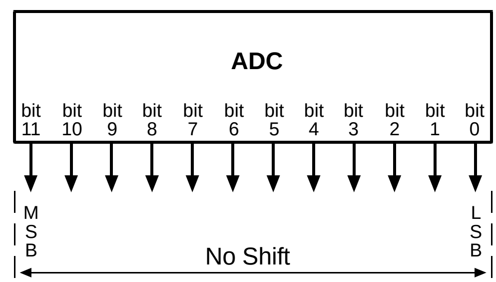
10.4 Digital Shift

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.

10.4.1 Digital Shift with 12 Bit Pixel Formats

No Shift

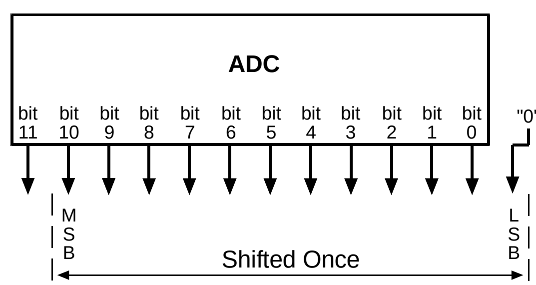
As mentioned in the Functional Description section of this manual, the camera uses a 12 bit ADC to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 12 bit effective depth, by default, the camera transmits the 12 bits that are output from 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.

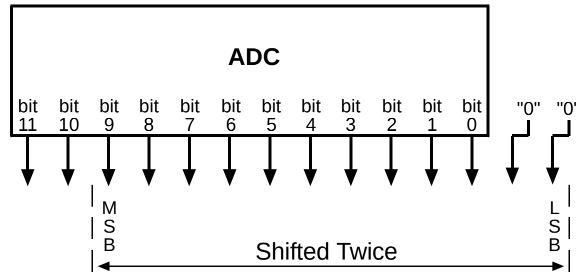


When the camera is set to shift by 1, the least significant bit output from the camera for each pixel value will be 0. This means that no odd gray values can be output and that the gray value scale will only include values of 2, 4, 6, 8, 10, and so on. This absence of some gray values is commonly referred 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.



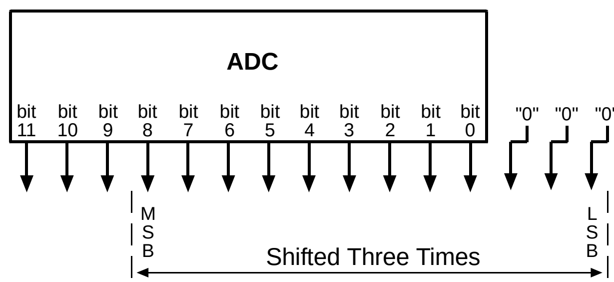
The result of shifting twice is that the output of the camera is effectively multiplied by 4.

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.



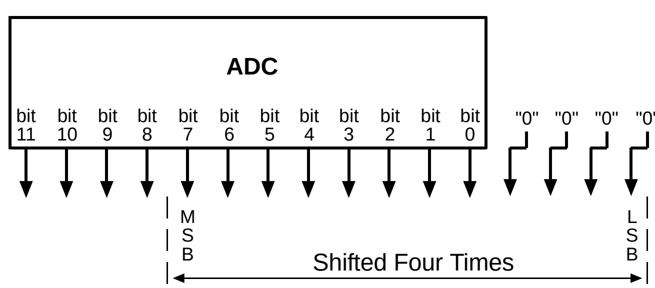
The result of shifting 3 times is that the output of the camera is effectively multiplied by 8.

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.

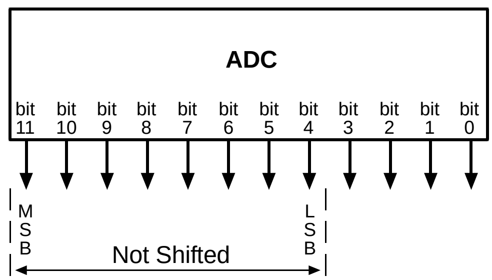
When the camera is set to shift by 4, the 4 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 16th gray value, for example, 16, 32, 48, 64, and so on.

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.

10.4.2 Digital Shift with 8 Bit Pixel Formats

No Shift

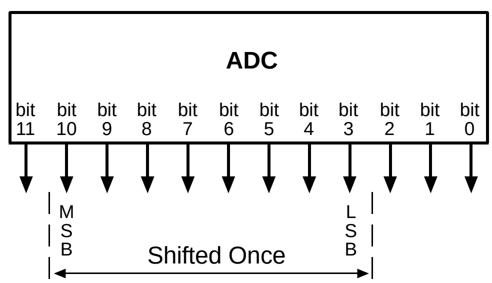
As mentioned in the Functional Description section of this manual, the camera uses a 12 bit ADC to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 8 bit effective depth, by default, the camera drops the 4 least significant bits from the ADC and transmits the 8 most significant bits (bit 11 through 4).



Shift by 1

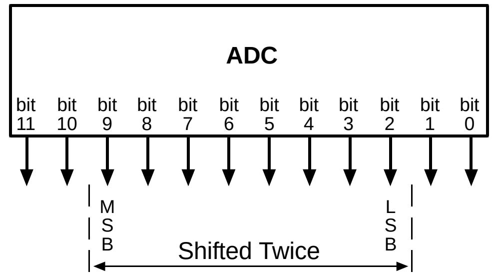
When the camera is set to shift by 1, the output from the camera will include bit 10 through bit 3 from the ADC.

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

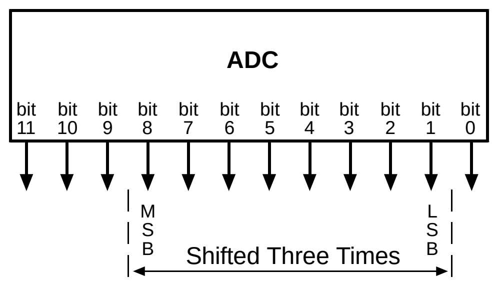


When the camera is set to shift by 2, the output from the camera will include bit 9 through bit 2 from the ADC.

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

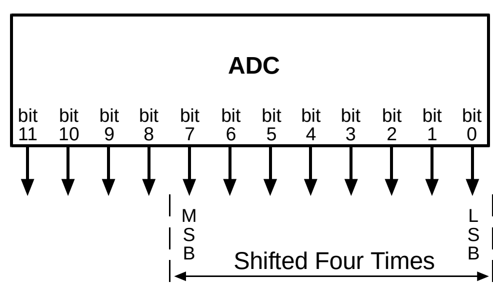


When the camera is set to shift by 3, the output from the camera will include bit 8 through bit 1 from the ADC.

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.

10.4.3 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.

10.4.4 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 );
```

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 page 39.

10.5 Image Area of Interest (AOI)

The image 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 read out of the sensor and into the camera's image buffer.

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 89.

The location and size of the area of interest is defined by declaring an offset X (coordinate), a width, an offset Y (coordinate), and a height. For example, suppose that you specify the offset X as 10, the width as 16, the offset Y as 6, and the height as 10. The area of the array that is bounded by these settings is shown in Figure 89.

The camera will only transmit pixel data from within the area defined by your settings. Information from the pixels outside of the area of interest is discarded.

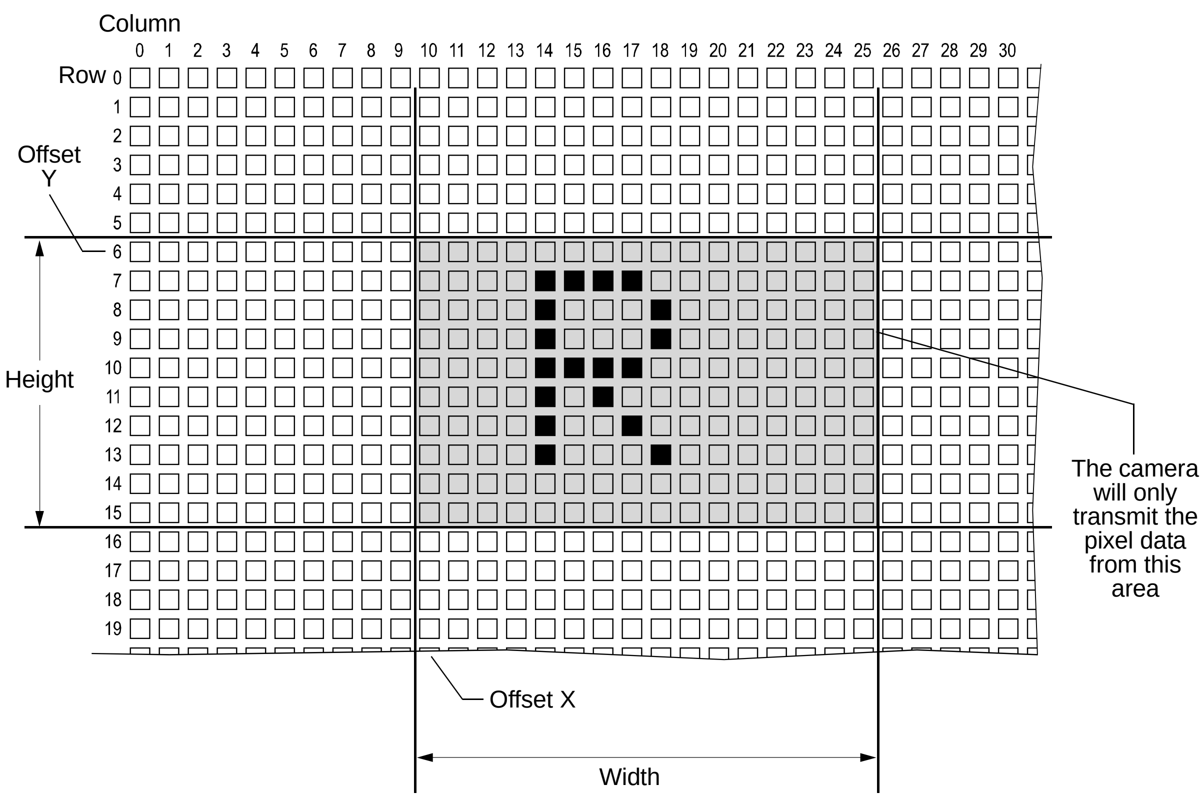


Fig. 89: Area of Interest

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 7.12 on page 143.

Setting the AOI

By default, the AOI is set to use the full resolution of the camera's sensor. You can change the size and the position of the AOI by changing the value of the camera's Offset X, Offset Y, Width, and Height parameters.

- The value of the Offset X parameter determines the starting column for the area of interest.
- The value of the Offset Y 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:

On all camera models:

- The sum of the Offset X setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the acA640-100gm, the sum of the Offset X setting plus the Width setting must not exceed 659.
- The sum of the Offset Y setting plus the Height setting must not exceed the height of the camera's sensor. For example, on the acA640-100gm, the sum of the Offset Y setting plus the Height setting must not exceed 494.

On all acA2500-14 cameras:

- The minimum Width setting is 64 and the minimum Height setting is 64.

On monochrome acA640-90, acA640-100, acA645-100, acA780-75, acA1300-30, acA1600-20, acA2000-50, acA2040-25, and acA2500-14 cameras:

- The Offset X, Offset Y, Width, and Height parameters can be set in increments of 1.

On monochrome acA750-30 cameras:

- The Offset X and Offset Y parameters can be set in increments of 2 and must be set to an even number. For example, the Offset Y parameter can be set to 0, 2, 4, 6, 8, etc.
- The Width and Height parameters can be set in increments of 4, i.e., to 4, 8, 12, 16, etc.

On color acA640-90, acA640-100, acA645-100, acA780-75, acA1300-30, acA1600-20, acA2000-50, acA2040-25, and acA2500-14 cameras:

- The Offset X, Offset Y, Width, and Height parameters can be set in increments of 2 and they must be set to an even number. For example, the Offset X parameter can be set to 0, 2, 4, 6, 8, etc.

On color acA750-30 cameras:

- The Offset X and Offset Y parameters can be set in increments of 2 and must be set to an even number. For example, the Offset Y parameter can be set to 0, 2, 4, 6, 8, etc.
- The Width and Height parameters can be set in increments of 4, i.e., 4, 8, 12, 16, etc.

ⓘ	Normally, the X Offset, Y Offset, Width, and Height parameter settings refer to the physical columns and rows of pixels in the sensor. But if binning is enabled, these parameters are set in terms of "virtual" columns and rows. For more information, see Section 10.8.3 on page 270.

You can set the Offset X, Offset Y, 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 Offset X, Offset Y, 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 );
```

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 page 39.

10.5.1 Changing AOI Parameters "On-the-Fly"

Making AOI parameter changes “on-the-fly” means making the parameter changes while the camera is capturing images continuously. On-the-fly changes are only allowed for the parameters that determine the position of the AOI, i.e., the Offset X and Offset Y parameters. Changes to the AOI size are not allowed on-the-fly.

10.6 Stacked Zone Imaging (acA2000-50, acA2040-25 Only)

The stacked zone imaging feature lets you define up to eight zones on the sensor array. When an image is acquired, only the pixel information from the areas within the defined zones will be read out of the sensor. The lines read out of the zones will then be stacked together and will be transmitted from the camera as a single image.

Using the stacked zone imaging feature increases the camera's frame rate.

The Stacked Zone Imaging Enable parameter is used to enable or disable stacked zone imaging. When the parameter is set to true, stacked zone imaging is enabled.

The Offset X and Width parameters are used to begin the process of setting up stacked zone imaging. Since all of the zones must be the same width and all of the zones must be vertically aligned, these two parameters define the left and right borders for all of the zones as shown in Figure 90 on page 229. In the figure, Offset X is set to 10 and the Width is set to 16.

The next step in the setup process is to define each individual zone. Up to 8 zones can be set up, with zone index numbers ranging from 1 through 8. Each zone can be enabled or disabled individually by first using the Stacked Zone Imaging Index parameter to select a zone number and then using the Stacked Zone Imaging Zone Enable parameter to enable the selected zone.
At least one zone must be enabled.

Once a zone has been enabled, you must use the Stacked Zone Imaging Zone Offset Y parameter to set the offset (in pixels) between the top of the sensor and the top of the zone. And you can use the Stacked Zone Imaging Zone Height parameter to set the height of the zone.

In Figure 90, for example, three zones have been enabled - zone 1, zone 2, and zone 3.

- The Offset X is set to 10 and the Width is set to 16. These settings apply to all zones.

For zone 1:

- The Stacked Zone Imaging Zone Offset Y parameter is set to 6
- The Stacked Zone Imaging Zone Height parameter is set to 6.

For zone 2:

- The Stacked Zone Imaging Zone Offset Y parameter is set to 20
- The Stacked Zone Imaging Zone Height parameter is set to 10.

For zone 3:

- The Stacked Zone Imaging Zone Offset Y parameter is set to 38.
- The Stacked Zone Imaging Zone Height parameter is set to 8.

With these settings, the camera would output an image that is 16 pixels wide and 24 lines (the total height of the three zones) high.

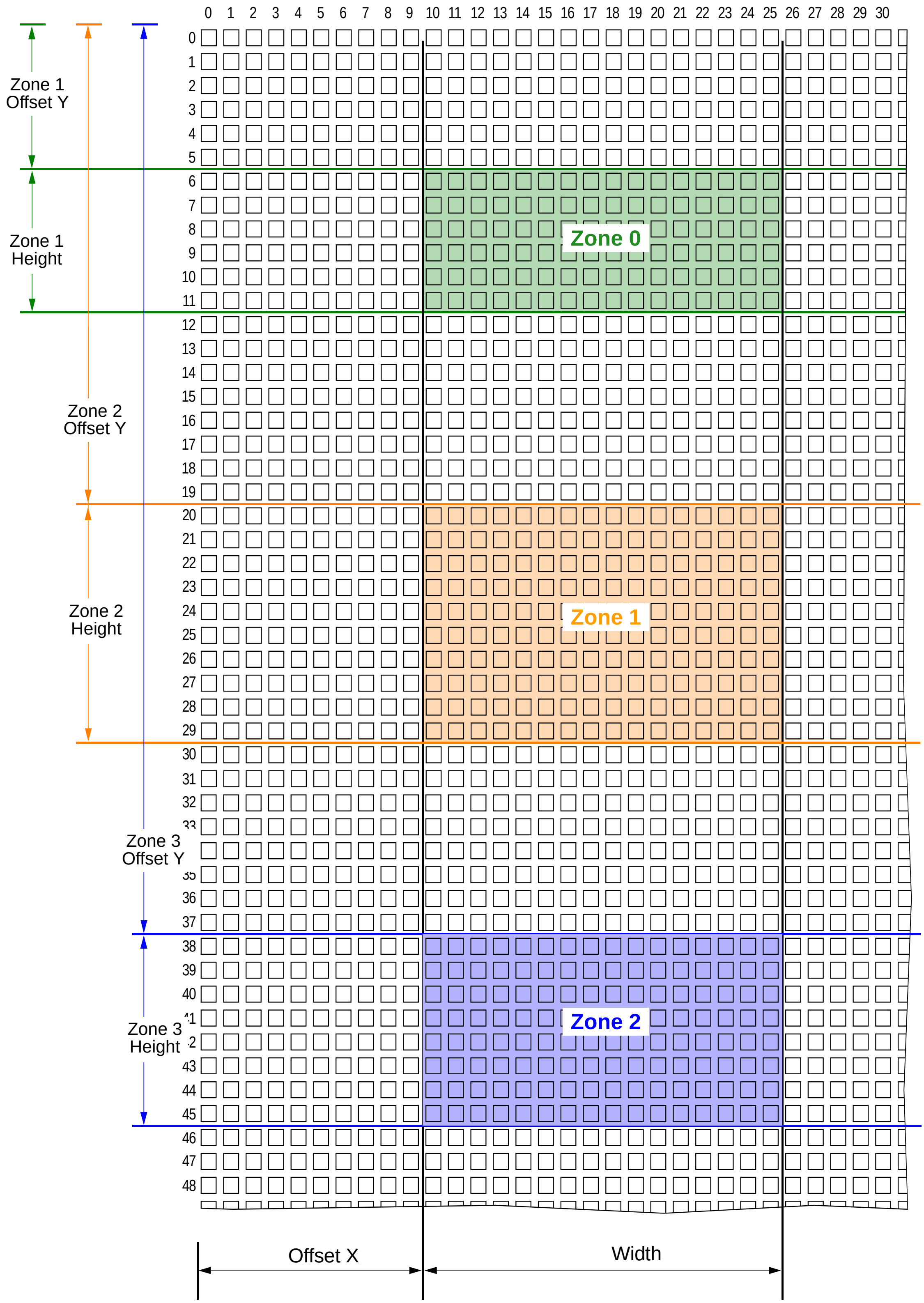


Fig. 90: Stacked Zone Imaging

There are several things to keep in mind when setting up zoned imaging:

- You are not required to enable the zones in sequence. For example, you can enable zones 2, 4, and 6 and not enable zones 1, 3, and 5.
- At least one zone must be enabled.
- Using binning effectively reduces the resolution of the camera's imaging sensor.
  As a consequence, if binning is enabled, the positions and the sizes of the set stacked zones are automatically adapted to the applied binning factors as follows: The stacked zones parameter values are divided by the corresponding binning factors (vertical and/or horizontal binning factor).

  If the stacked zone parameter values are not evenly divisible by the corresponding binning factor, the parameter values are automatically rounded down to the nearest whole number.

  Example for zone 1:

Stacked Zone Imaging Parameter	Without Binning	With Binning by 2	With Binning by 3
Offset X (valid for all zones)	10	5	3
Width (valid for all zones)	16	8	5
Offset Y	6	3	2
Height	6	3	2

Table 22: Examples: Binning Influence on Stacked Zone Imaging Feature

- You do not need to order the zones from top to bottom on the sensor. For example, you could place zone 1 near the bottom of the sensor, zone 3 near the top, and zone 2 in the middle.

  But note that the camera always reads out and transmits the zones starting from the top of the sensor and going to the bottom, regardless of how the zone numbers are ordered. So the lines in the transmitted images will always be ordered from top to bottom in relation to the sensor.
- The zones can be set so that they overlap. When this happens, the camera will internally transform the overlapped zones into a single large zone that will be read out and transmitted as if it were simply a single large zone. (The lines included in the overlapping area will only be read out and transmitted once.)

When stacked zone imaging is enabled, the camera's Offset Y parameter becomes read only, and this parameter indicates the Y offset for the zone nearest to the top of the sensor.

When stacked zone imaging is enabled, the Height parameter becomes read only, and this parameter indicates the total height of the image that will be transmitted from the camera (i.e., the sum of the heights of all zones).

10.6.1 Setting Stacked Zone Imaging

Guidelines

When you are setting the stacked zones, you must follow these guidelines:

On all camera models:

- The sum of the Offset X setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the acA2000-50gm, the sum of the Offset X setting plus the Width setting must not exceed 2048.
- For any given zone, the sum of the Stacked Zone Imaging Zone Offset Y setting plus the Stacked Zone Imaging Zone Height setting must not exceed the height of the camera's sensor. For example, on the acA2000-50gm, the sum of the Stacked Zone Imaging Zone Offset Y setting plus the Stacked Zone Imaging Zone Height setting must not exceed 1088.

On monochrome camera models:

- The Stacked Zone Imaging Zone Offset Y and Stacked Zone Imaging Zone Height parameters for any given zone can each be set in increments of 1.
- The Offset X and Width parameters can be set in increments of 1.

On color camera models:

- The Stacked Zone Imaging Zone Offset Y and Stacked Zone Imaging Zone Height parameters for any given zone can each be set in increments of 2.
- The Offset X and Width parameters can be set in increments of 2.

Setting Stacked Zone Imaging Using Basler pylon

You can set the parameter values associated with stacked zone imaging from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to set up two zones.

```
// Enable stacked zone imaging
Camera.StackedZoneImagingEnable.SetValue( true );

// Set the width and offset X for the zones
Camera.Width.SetValue( 200 );
Camera.OffsetX.SetValue( 100 );

// Set zone 1

// Select the zone
Camera.StackedZoneImagingIndex.SetValue( 1 );
// Enable the selected zone
Camera.StackedZoneImagingZoneEnable.SetValue( true );
// Set the offset Y for the selected zone
Camera.StackedZoneImagingZoneOffsetY.SetValue( 100 );
// Set the height for the selected zone
Camera.StackedZoneImagingZoneHeight.SetValue( 100 );

// Set zone 2

// Select the zone
Camera.StackedZoneImagingIndex.SetValue( 2 );
// Enable the selected zone
Camera.StackedZoneImagingZoneEnable.SetValue( true );
// Set the offset Y for the selected zone
Camera.StackedZoneImagingZoneOffsetY.SetValue( 250 );
// Set the height for the selected zone
Camera.StackedZoneImagingZoneHeight.SetValue( 200 );
```

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 page 39.

10.7 Sequencer

The sequencer feature will not work, if the auto functions feature is enabled. For more information about the auto functions feature, see Section 10.12 on page 282.

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. Since the active set controls camera operation, you will see a change in camera operation when you change a parameter in the active set.

The parameters in the active set can be divided into two types (Figure 91):

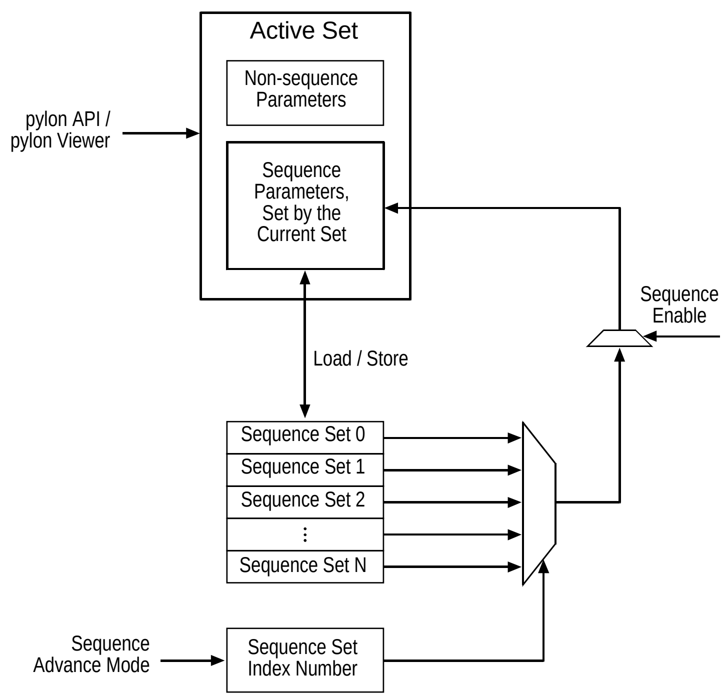


Fig. 91: Sequence Feature Block Diagram

- "**non-sequence**" parameters: Cannot be changed using the sequencer feature.
- "**sequence**" parameters: Because the sequence sets reside in the camera's FPGA, you can replace the values in the active set with values from one of the sequence sets almost instantaneously as images are acquired.

The following sequencer parameters determining the sequencer logic are stored in the factory set (see page 311) with default values:
Sequence Enable, Sequence Set Executions, Sequence Control Source, Sequence Address Bit Source, Sequence Set Total Number, Sequence Set Index.

Every time the camera is **restarted**, all sequencer parameters are **reset to the default** values, e.g. if you enable and use the sequencer feature with specially set values, and you turn off and on the camera, the sequencer feature is disabled after restart, and the user-defined parameters are reset to default values.

Make sure the sequencer feature is disabled when configuring sequence sets.

When the sequencer feature is enabled, the values of 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.

We recommend that you do not attempt to read or change any of the sequence parameters when the sequencer feature is enabled.

Using the sequencer feature has no effect on the camera's frame rate (see exception).

Exception (acA2500-14)

If you use the acA2500-14 in the **overlapped mode of operation**, and you activate the sequencer feature, it depends on the way you use the sequencer, whether it has an effect on the frame rate or not:

If the camera takes multiple images

- ... with the **same** sequence set, overlapped operation is possible and the sequencer feature has **no** effect on the camera's frame rate.
- ... with **alternating** sequence sets, overlapped operation is not possible.
  The camera must complete the entire exposure/readout process before a new sequence set can be loaded.
  In this case the initial overlapped operation turns out to work as non-overlapped operation.

  As a consequence the frame rate can be significantly reduced.

The sequence set currently setting the parameter values of the sequence parameters in the active set is also called the "current set".

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
X Offset
Y Offset
Center X
Center Y
Binning Horizontal
Binning Vertical
Pixel Format
Test Image
Gain
Processed Raw Enable
Black Level
Color Adjustment Enable
Color Adjustment Hue
Color Adjustment Saturation
Chunk Mode Active
Chunk Enable
Timer Delay*
Timer Duration*
Timer Delay Timebase*
Timer Duration Timebase*
Sequence Set Executions**

* This parameter is available for timer 1.
**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.

Make sure the sequencer feature is disabled when configuring sequence sets.

When the sequencer feature is enabled, the values of 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.

We recommend that you do not attempt to read or change any of the sequence parameters when the sequencer feature is enabled.

- 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.
- Sequence sets **can not be saved** in **user sets**.

Sequence Advance

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 line.

The regular cycling through the sequence sets according to the Auto or Controlled advance modes can be 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 the 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 the input line are also called "synchronous advance" and "synchronous restart" because the checking of the states of the 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

Part of the standard operation of the sequencer feature and should generally be used.
We strongly recommend to **only** use synchronous advance and synchronous restart for real-time applications.

Asynchronous advance and restart

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 ma be useful for testing purposes.

The sequence set index chunk feature adds a chunk to each acquired frame containing the index number of the sequence set that was used for frame acquisition. For more information about the Sequence Set Index chunk, see Section 11.8 on page 331.

Using the Load Command

Make sure the sequencer feature is disabled before issuing the Sequence Set Load command.

The **Sequence Set Load** command can be useful for testing purposes:

If you 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.
- 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.

The Sequence Set Load command is **not suitable** for real-time applications.

If you use the Sequence Set Selector 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.

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 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.

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.

10.7.1 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 is 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.

10.7.1.1 Operation

Operating the Sequencer

The following use case (see also Figure 92) 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

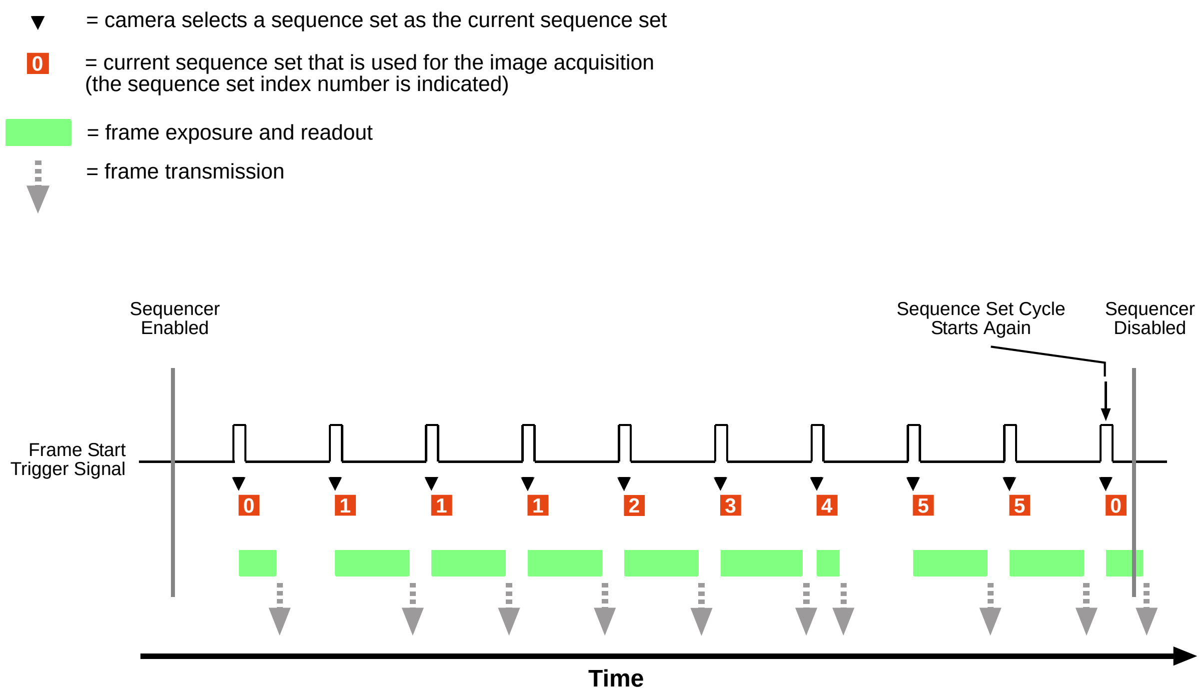


Fig. 92: Sequencer in Auto Sequence Advance Mode

Operating the Sequencer Using Basler 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 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.

10.7.1.2 Configuration

Configuring Sequence Sets and Advance Control

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 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 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.

10.7.2 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.
- Line 1: The states of the input line 1 control sequence set advance.
- Disabled: Sequence set advance 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.

10.7.2.1 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 93) 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 is 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 is 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

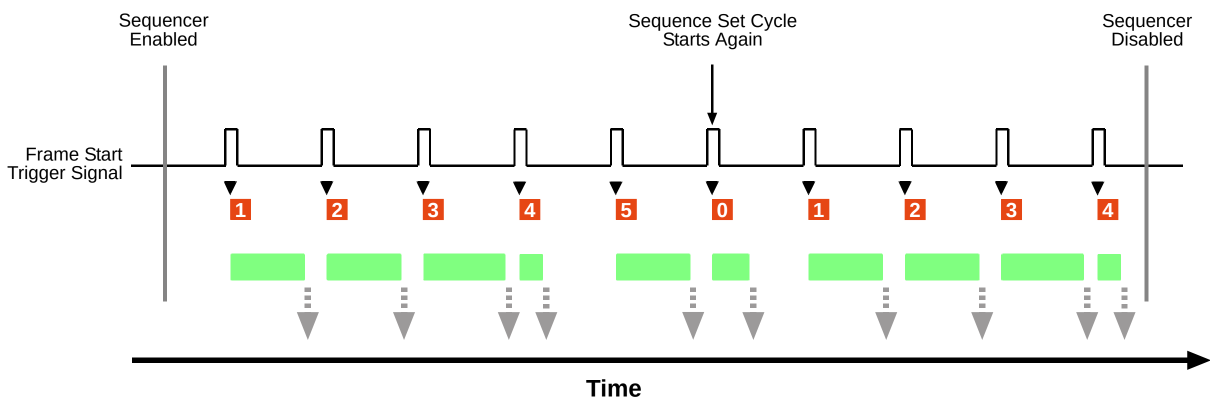


Fig. 93: Sequencer in Controlled Sequence Advance Mode with Always Active as the Sequence Control Source

Synchronous Restart

You can restart the sequence cycle with input line 1 as the source for controling sequence cycle restart.

In the following use case (see also Figure 94), 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 is 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 is 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 is 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 secquence 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.

> 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 94).

ⓘ	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 Acquisiton Monitoring Tools section.

- When the next frame start trigger is 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 is 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 secquence set advance.
- When the next frame start triggers are 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.

Use Case: Operation in controlled sequence advance mode with Always Active as the sequence control source:

Automatic cycling through the sequence set cycles with two synchronous restartscontrolled by input line 1.

Setting: Sequence Set Total Number = 6
Line 1 (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

= current sequence set that is used for the image acquisition (the sequence set index number is indicated)

= frame exposure and readout

= frame transmission

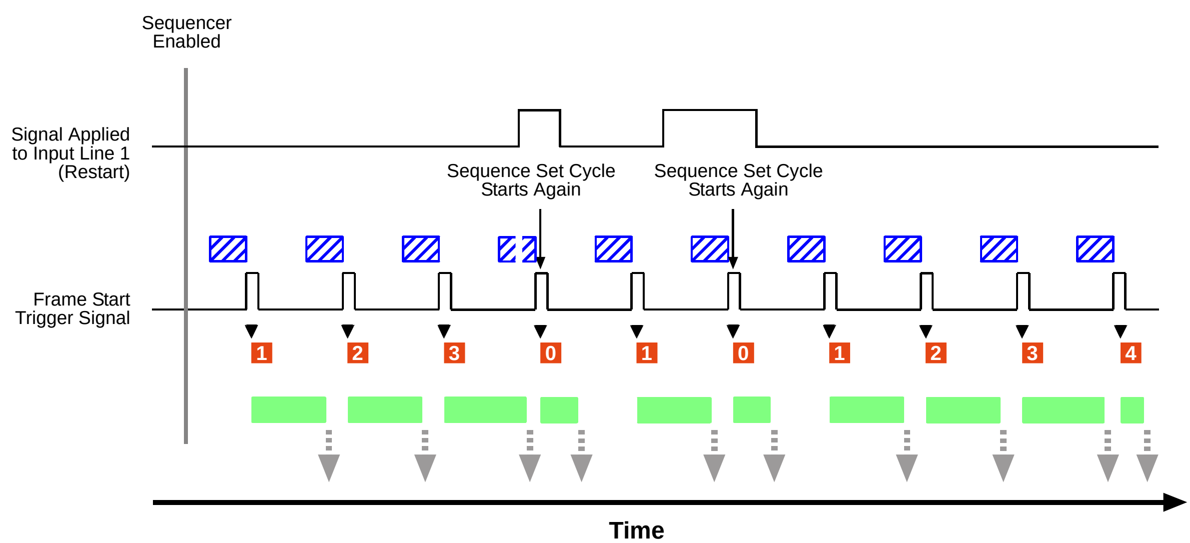


Fig. 94: Sequencer in Controlled Sequence Advance Mode with Always Active as the Sequence Control Source and Synchronous Restart Controlled by Line 1

10.7.2.2 Operation with the Input Line as Sequence Control Source

Operating the Sequencer

When the Line 1 sequence control source is selected the advance from one sequence set to the next is controlled according to the states of input line 1. The advance proceeds in ascending sequence set index numbers as frame start triggers are received.

The following use case (see also Figure 95) 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.

ⓘ	Note that sequence set advance is not influenced by the state of the input line at the time when the sequencer feature was enabled. For example, had Line 1 been high at the time of the enabling but then become low and remained there when the first frame start trigger signal was received then sequence set 0 had been used for the first image acquisition.

- When the next frame start trigger is 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.

ⓘ	To ensure reliable selection of a sequence set, allow the elapse of at least one microsecond between setting the states 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 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 95).

> 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 Acquisiton Monitoring Tools section.

- When the next frame start trigger is 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 is 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 is 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 is 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 is 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 is 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 is 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

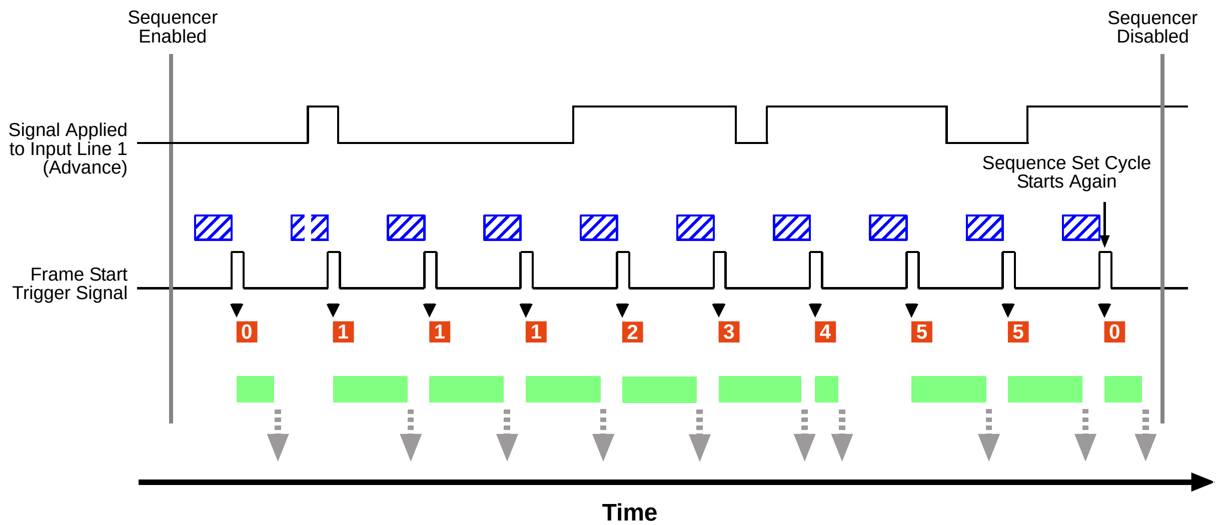


Fig. 95: Sequencer in Controlled Sequence Advance Mode with Line 1 as the Sequence Control Source

10.7.2.3 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 avance.
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 96) 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 96).

ⓘ	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 Acquisiton 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 96).

ⓘ	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 Acquisiton 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

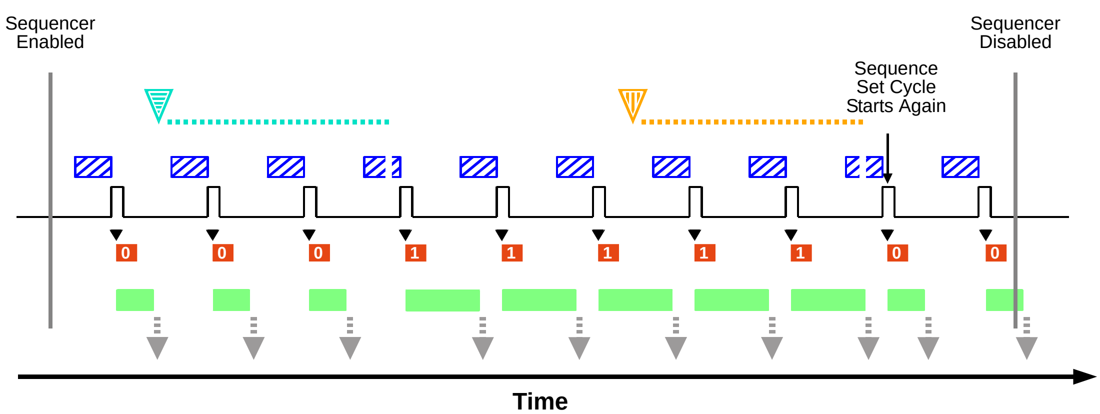


Fig. 96: Sequencer in Controlled Sequence Advance Mode with Disabled as the Sequence Control Source and Asynchronous Advance and Restart

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.

10.7.2.4 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 synchronous sequence set advance.
- Set the Sequence Control Source parameter to specify the source that will control sequence set advance.

> All sequence sets that will be available at the same time in the camera's memory must be set to the same source for sequence set advance. Accordingly, setting some sets to e.g. Disabled and some to Line 1 is not allowed.

The following sources are available:

- Always Active
- Line 1:
- 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.

ⓘ	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:
  - 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 6.1.3 on page 69.

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.

10.7.3 Free Selection Sequence Advance Mode

When the free selection sequence advance mode is selected the advance form 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 line 1: The states of the input line 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 Sequence Set Total Number parameter specifies the total number of sequence sets that are available. The maximum number is 2.

10.7.3.1 Operation

Operating the Sequencer

The following use case (see also Figure 97) 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 two. Accordingly, the available sequence set index numbers are 0 and 1. Input line 1 sets bit 0 of the sequence set address. The input line is 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 state of input line 1. Input line 1 is found to be low. This corresponds to the address of sequence set 0. Accordingly, sequence set 0 is selected. Its parameter values 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 state of input line 1. Because the state has not changed 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. Because the state has not changed 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. 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 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 changed its state. This happened when input line 1 went high before the frame start trigger was receceived (see also Figure 97).

ⓘ	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 Acquisiton 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. This corresponds to the address of sequence set 0. Accordingly, sequence set 0 is selected. Its parameter values are loaded into the active set and are used for the image acquisition.
- When the remaining frame start triggers were received, the camera checks the state of input line 1. Input line 1 is found to be high. 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 remaining image acquisitions.
- When the remaining frame start triggers were received, the camera checks the state of input line 1. Because the state has not changed and will not for the remaining frame start triggers the parameter values of sequence set 1 are used for the image acquisitions.

  Note that the camera briefly exits the "waiting for frame start trigger" status while the input line briefly changed its state before the ninth frame start trigger was received..
- While frame exposure and readout for the ninth 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 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 line.
Settings:	Sequence Set Total Number = 2 Input line 1 (not set for invert) sets bit 0 of the sequence set address.

 = camera is waiting for a frame start trigger

 = camera selects a sequence set as the current sequence set

 = current sequence set that is used for the image acquisition (the sequence set index number is indicated)

 = frame exposure and readout

 = frame transmission

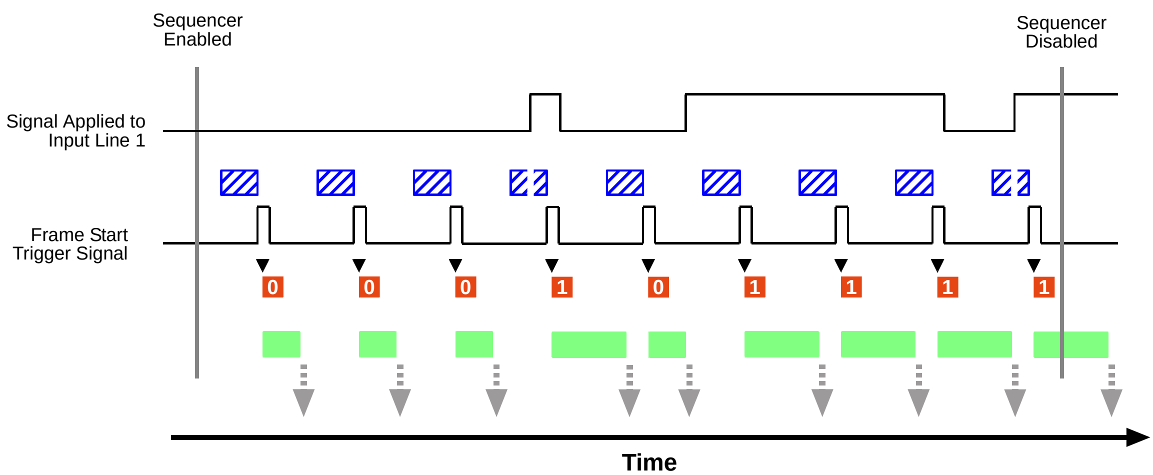


Fig. 97: Sequencer in Free Selection Mode

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 line selects between the sequence sets by setting bit 0 of the sequence set address. The address is simply the binary expression of the sequence set index number (see Table 23).

- 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.

A maximum of two sequence sets can be used.

Sequence Set Address	Related Sequence Set
Bit 0	
0	Sequence Set 0
1	Sequence Set 1

Table 23: Sequence Set Addresses and Related Sequence Sets (Input Line Not Set for Invert)

10.7.3.2 Configuration

Configuring Sequence Sets and Advance Control

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 2.
4. Select the sequence set address bit and set the input line that will act as the control source:
   a. Bit 0 will be selected by default as the sequence set address bit. Set input line 1 as the control source 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 and 1.
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 set, starting from step 5.

Configuring Sequence Sets and Advance Control Using Basler pylon

You can use the pylon API to set the parameters for populating 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 of the sequence set address, set the total number of sequence sets to 2, 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 Free Selection sequence advance mode
Camera.SequenceAdvanceMode.SetValue(
SequenceAdvanceMode_FreeSelection );

// Set the total number of sequence sets
Camera.SequenceSetTotalNumber.SetValue( 2 );
```

```
// Set line 1 as the control source for setting sequence set address
bit 0
Camera.SequenceAddressBitSelector.SetValue(
SequenceAddressBitSelector_Bit0 );
Camera.SequenceAddressBitSource.SetValue(
SequenceAddressBitSource_Line1 );

// 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( );
```

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

10.8 Binning

10.8.1 Binning on Monochrome Cameras

ⓘ	On all cameras, except the acA2500-14, the binning feature is only available on monochrome cameras. On the acA750-30gm, only horizontal binning by 2 is available.

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 rows, 3 rows, or a maximum of 4 rows in the imaging sensor array are summed and are reported out of the camera as a single pixel. Figure 98 illustrates vertical binning.

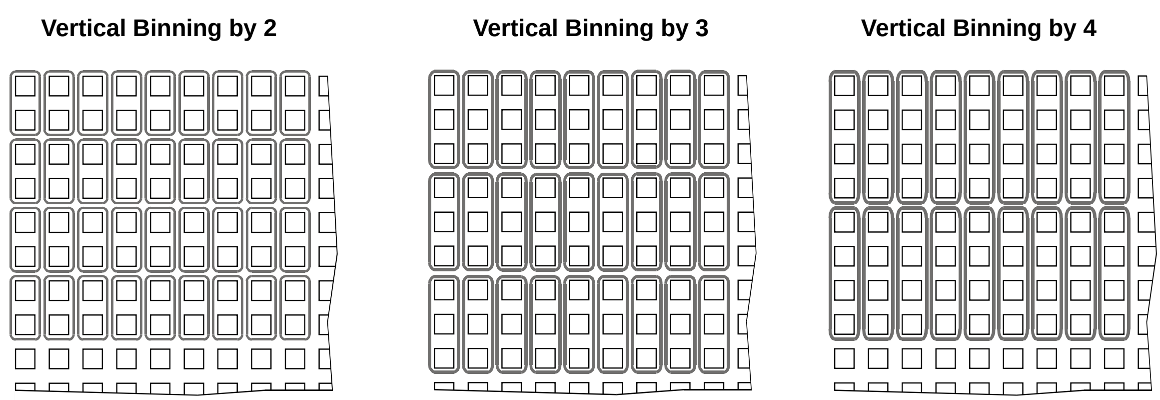


Fig. 98: Vertical Binning on Monochrome Cameras

ⓘ	For the acA2500-14gm, vertical binning works in a different way: ■ Vertical binning by 2 and by 4: The gray values of adjacent pixels from 2 rows or from 4 rows are averaged. As a consequence, the signal to noise ratio will be increased while the camera's response to light will not be increased. ■ Vertical binning by 3: The gray values of adjacent pixels from 3 rows are combined. As a consequence, the signal to noise ratio will be decreased while the camera's response to light will be slightly increased. We recommend using vertical binning by 2 or by 4.

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 99 illustrates horizontal binning.

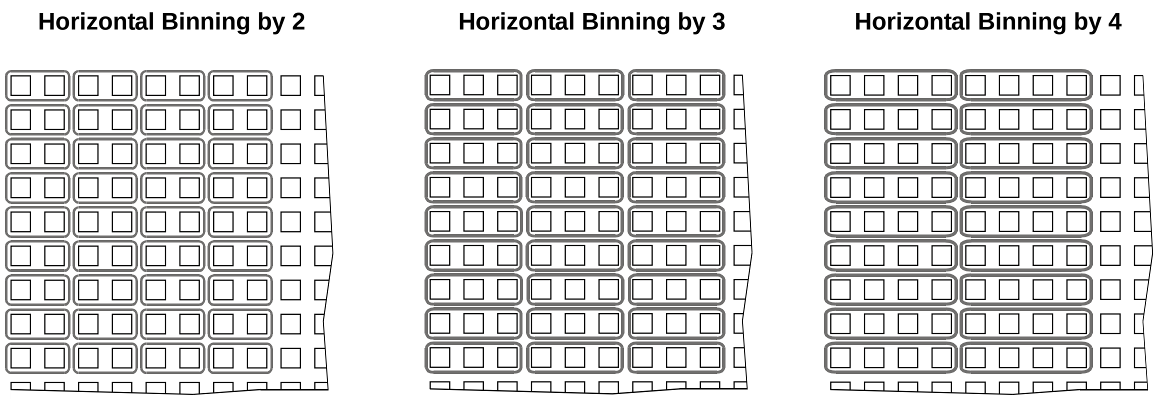


Fig. 99: Horizontal Binning on Monochrome Cameras

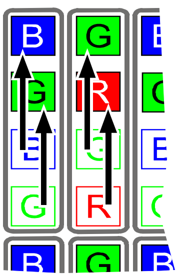
You can combine vertical and horizontal binning. This, however, may cause objects to appear distorted in the image. For more information about possible image distortion due to combined vertical and horizontal binning, see Section 10.8.3 on page 270.

10.8.2 Binning on Color Cameras (acA2500-14gc Only)

The acA2500-14gc color camera allows you to realize color binning, where pixel data for identical colors are binned vertically and/or horizontally.

With **vertical** color binning, the gray values of adjacent pixels of the same color from 2 rows, 3 rows, or a maximum of 4 rows in the imaging sensor array are **averaged** and are reported out of the camera as a single pixel. The number of binned pixels depends on the vertical color binning setting (see the example in Figure 100).

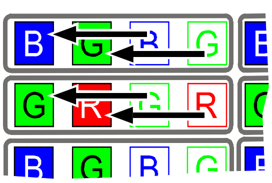
As the gray values are averaged during vertical color binning and not summed, the signal to noise ratio will be increased while the camera's response to light will not be increased.



Example:
Vertical Color Binning by 2
(Shown for 2 Columns)

Fig. 100: Vertical Color Binning by 2

With **horizontal** color binning, the gray values of adjacent pixels of the same color from 2 columns, 3 columns, or a maximum of 4 columns in the imaging sensor array are **summed** and are reported out of the camera as a single pixel. The number of binned pixels depends on the horizontal color binning setting (see example in Figure 101).



Example:
Horizontal Color Binning by 2
(Shown for 2 Rows)

Fig. 101: Horizontal Color Binning by 2

Combining Horizontal and Vertical Color Binning

You can combine vertical and horizontal color binning (see the example in Figure 102).

Example: Horizontal and Vertical Color Binning by 2

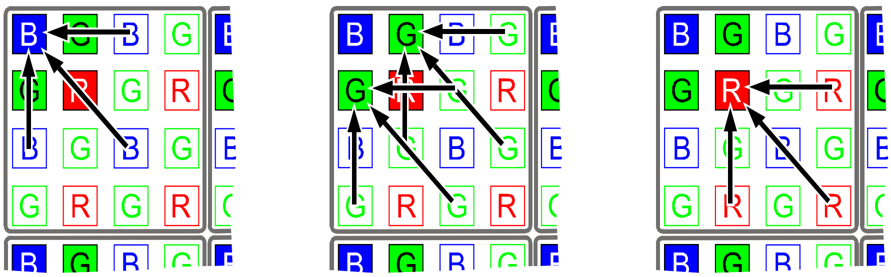


Fig. 102: Combining Vertical and Horizontal Color Binning

You can combine vertical and horizontal binning. This, however, may cause objects to appear distorted in the image. For more information about possible image distortion due to combined vertical and horizontal binning, see Section 10.8.3 on page 270.

Setting Binning

You can enable vertical color binning for the acA2500-14gc color camera by setting the Binning Vertical parameter. Setting the parameter's value to 2, 3, or 4 enables vertical color binning by 2, by 3, or by 4, respectively. Setting the parameter's value to 1 disables vertical color binning.

You can enable horizontal color binning for the acA2500-14gc color camera by setting the Binning Horizontal parameter. Setting the parameter's value to 2, 3, or 4 enables horizontal color binning by 2, by 3, or by 4, respectively. Setting the parameter's value to 1 disables horizontal color 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 );
```

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 page 39.

10.8.3 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, the intensity of your illumination, the camera's exposure time setting, or the camera's gain setting.

When using vertical binning on monochrome cameras, 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 10.1 on page 209.
Note: The vertical binning of the acA2500-14gm works differently. For more information, see Section 10.8 on page 266.

Reduced Resolution

Using binning effectively reduces the resolution of the camera's imaging sensor. For example, the sensor in the acA640-100gm camera normally has a resolution of 659 (H) x 494 (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 219 (H) by 164 (V). (Note that neither dimension of the sensor was 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 numbers 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 an acA640-100gm camera set for 3 by 3 binning as described above. In this case, you would act as if you were actually working with a 219 column by 164 line sensor when setting your AOI parameters. The maximum AOI width would be 219 and the maximum AOI height would be 164. 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 10.5 on page 225.

Binning's Effect on Stacked Zone Imaging (acA2000-50, acA2040-25 Only)

Using binning effectively reduces the resolution of the camera's imaging sensor.
As a consequence, if binning is enabled, the positions and the sizes of the set stacked zones are automatically adapted to the applied binning factors as follows: The stacked zone imaging parameter values are divided by the corresponding binning factors (vertical and/or horizontal binning factor).

If the stacked zone imaging parameter values are not evenly divisible by the corresponding binning factor, the parameter values are automatically rounded down to the nearest whole number.

Example for zone 1:

Stacked Zone Imaging Parameter	Without Binning	With Binning by 2	With Binning by 3
Offset X (valid for all zones)	10	5	3
Width (valid for all zones)	16	8	5
Offset Y	6	3	2
Height	6	3	2

Table 24: Examples: Stacked Zone Imaging Settings for Zone 1

For more information about the stacked zone imaging feature, see Section 10.6 on page 228.

Binning's Effect on Decimation

If vertical binning is used, vertical decimation is automatically disabled, and vice versa, i.e. if vertical decimation is used, vertical binning is disabled.

Horizontal binning works independently of the decimation feature.

10.9 Vertical Decimation (acA2000-50, acA2040-25 Only)

The vertical decimation feature (sub-sampling) lets you specify the extent of vertical sub-sampling of the acquired frame, i.e. you can define rows that you want to be left out from transmission.

The acA2000-50 and acA2040-25 cameras only support decimation in vertical direction.

Examples
(Blue rows will be transmitted):

If vertical decimation is set to

- 1: the complete frame will be transmitted out of the camera (no sub-sampling is realized); see Figure 103.
  This is valid for mono and color cameras.
- 2 for mono cameras: only every second row of the acquired frame will be transmitted out of the camera (Figure 104).
- 2 for color cameras: only every second pair of rows of the acquired frame will be transmitted out of the camera (Figure 105).

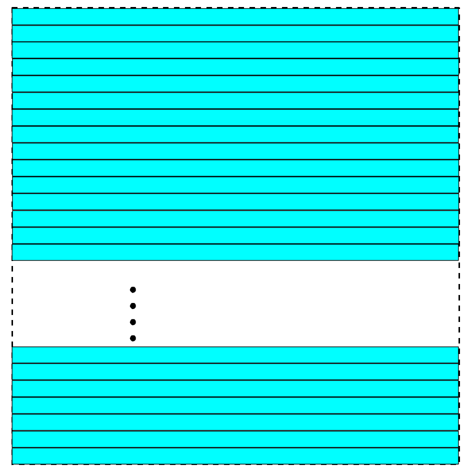


Fig. 103: Decimation Disabled

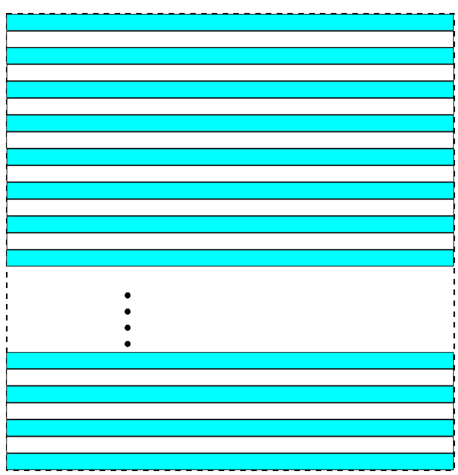


Fig. 104: Decimation of 2 (Mono Cameras)

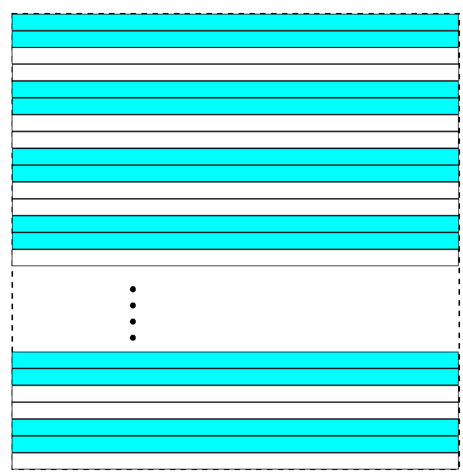


Fig. 105: Decimation of 2 (Color Cameras)

By using the vertical decimation feature, you can increase the frame rate of the camera.

Setting Vertical Decimation

You can enable vertical decimation for the acA2500-50 and acA2040-25 cameras by setting the Vertical Decimation parameter. Setting the parameter’s value to 1 disables vertical decimation.

You can set the Vertical Decimation 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 Decimation by 8
Camera.DecimationVertical.SetValue( 8);

// Disable Vertical Decimation
Camera.DecimationVertical.SetValue( 1 );
```

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 page 39.

10.9.1 Considerations When Using Decimation

Reduced Vertical Resolution

Using vertical decimation effectively reduces the vertical resolution of the camera's imaging sensor. For example, the sensor in the acA2000-50gm camera normally has a resolution of 2048 (H) x 1088 (V). If you set this camera to use vertical decimation by 5, the effective resolution of the sensor is reduced to 2048 (H) by 217 (V).

If you reduce the vertical resolution by using the vertical decimation feature, you can increase the frame rate of the camera.

Possible Image Distortion

Objects will only appear undistorted in the image, if the numbers of lines and columns are equal. With all other combinations, the imaged objects will appear distorted. If, for example, vertical decimation is set to 2, the imaged objects will appear shrunk by a factor of 2 compared to an image without vertical decimation.

Binning and Vertical Decimation

If vertical binning is used, vertical decimation is automatically disabled, and vice versa, i.e. if vertical decimation is used, vertical binning is disabled.

Horizontal binning works independently from the decimation feature.

Decimation’s Effect on AOI Settings

If vertical decimation is activated, the camera automatically adapts the AOI settings to the modified image size based on the formulas below.

For evaluating the new AOI height, the camera takes into account the number of physical lines that are between the first transmitted line (L1) and the last transmitted line (Ln), i.e. the so-called covered lines (see Figure 106). The line Ln + 1 in our example would not be part of the covered lines when the decimation feature is activated.

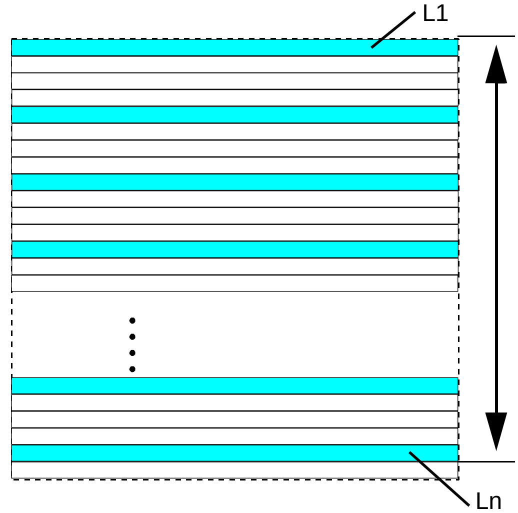


Fig. 106: Covered Lines (Mono Camera)

Calculating the **covered lines** (C)

- For **mono** cameras:
  (C) = H_old x D_old - D_old + 1
- For **color** cameras:
  (C) = H_old x D_old - 2 x D_old +2

As soon as the covered lines are determined, the camera calculates the new AOI height:

- For **mono** cameras:
  New AOI height = Round up (C / D_new)
- For **color** cameras:
  New AOI height = 2 x Round up ((C / 2) / D_new)

C = Coverage
D_new = New decimation value
D_old = Old decimation value
H_new = New AOI height
H_old = Old AOI height

The following examples show how the modification of the vertical decimation value influences the automatically calculated new AOI height:

Vertical decimation set ...	acA2000-50gm [Original AOI height =1088]	acA2040-25gc [Original AOI height =2046]
from 1	Covered lines: 1088 AOI height: 1088	Covered lines: 2046 AOI height: 2046
to 10	Covered lines: 1088 New AOI height: 109	Covered lines: 2042 New AOI height: 206
to 1 again	Covered lines: 1081 New AOI height (*): 1081	Covered lines: 2042 New AOI height (*): 2042
to 7	Covered lines: 1081 New AOI height: 155	Covered lines: 2042 New AOI height: 292
to 1 again	Covered lines: 1079 New AOI height (*): 1079	Covered lines: 2032 New AOI height (*): 2032

- The values in this table are calculated based on the formulas explained above.
- (*) = See note on "AOI height can be reset to maximum" below.

Table 25: Vertical Decimation and AOI Heights (Sample Calculations)

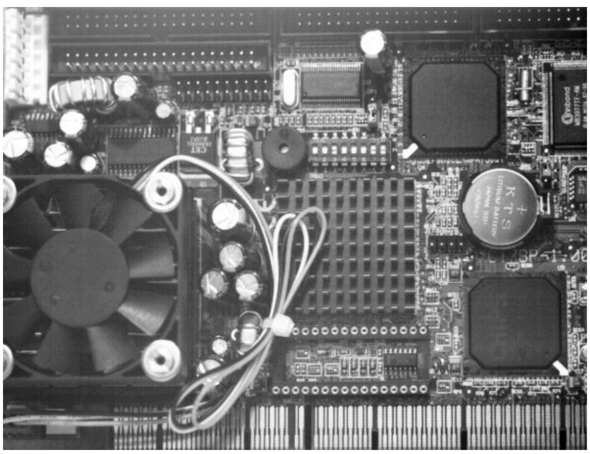
ⓘ	**AOI height** If you use the vertical decimation feature and you reset the decimation parameter back to 1, i.e. decimation is deactivated, the AOI height can be smaller than the maximum possible height (determined by the pixel resolution in vertical direction), see examples in Table 25. In this case you can manually set the AOI height back to the maximum possible height.

10.10 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 107 shows a normal image on the left and an image captured with reverse X enabled on the right.

Normal Image



Mirror Image

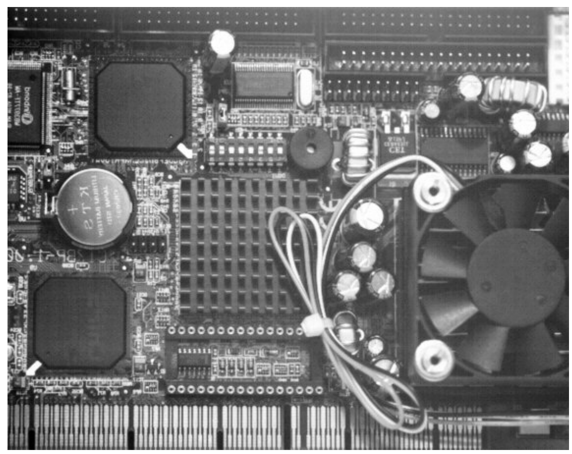


Fig. 107: Reverse X Mirror Imaging

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.

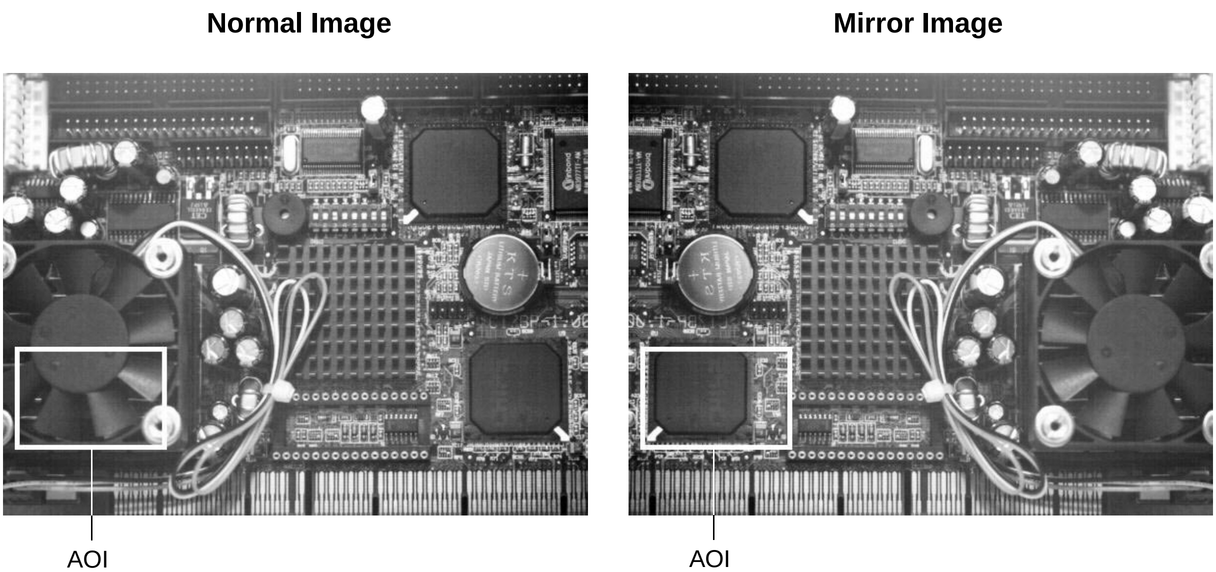


Fig. 108: Using an AOI with Reverse X Mirror Imaging

ⓘ	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 auto 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 10.12 on page 282.

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);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

10.11 Luminance Lookup Table

Pixel data from the imaging sensor is digitized by the ADC at 12 bit depth. Whenever the camera is set for a 12 bit pixel format (e.g., Mono 12), the 12 bits transmitted out of the camera for each pixel normally 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 12 bits that will be transmitted out of the camera when the ADC 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 12 bits that will be transmitted out of the camera when the ADC 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 12 bits that will be transmitted out of the camera when the ADC 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 12 bits that will be transmitted out of the camera when the ADC 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 ADC 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 ADC 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 ADC reports a value above 4088, the camera will not be able to perform an interpolation. In cases where the ADC 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 digitized 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 digitized sensor output moves from 0 through 2048 and increases gradually as the digitized sensor output moves from 2049 through 4096.

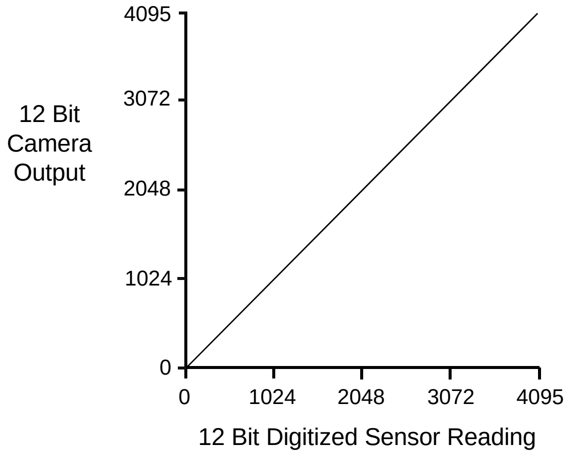


Fig. 109: Lookup Table with Values Mapped in a Linear Fashion

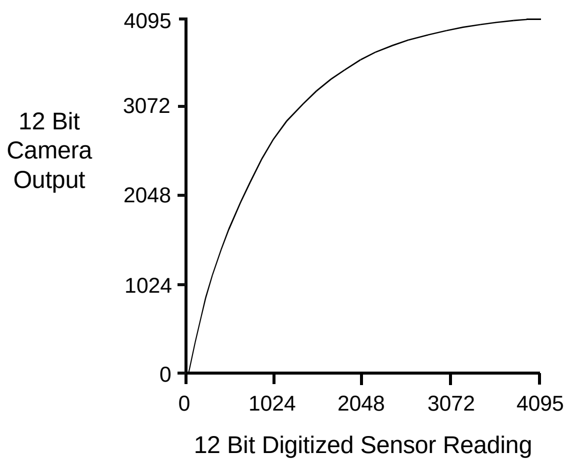


Fig. 110: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

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 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 (e.g., Mono 8). 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 Basler 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 );
```

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 page 39.

10.12 Auto Functions

ⓘ	The auto functions feature will not work,if the sequencer feature is enabled. For more information about the sequencer feature, see Section 10.7 on page 233.

10.12.1 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 and the acA2500-14gc only). An auto function uses the binned pixel data and controls the image property of the binned image.

For more information about binning, see Section 10.8 on page 266.

10.12.2 Auto Function Operating Modes

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.

10.12.3 Auto Function AOIs

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 always share an Auto Function AOI and some auto functions can use their own individual Auto Function AOIs. Within these limitations, auto functions can be assigned to Auto Function AOIs as desired.

Each Auto Function AOI has its own specific set of parameter settings, and the parameter settings for the Auto Function AOIs are not tied to the settings for the AOI that is used to define the size of captured images (Image AOI). For each Auto Function 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. Note that an Auto Function AOI can be positioned anywhere on the sensor array.

An Auto Function AOI is referenced to the top left corner of the sensor array. The top left corner of the sensor array is designated as column 0 and row 0 as shown in Figure 111.

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 111.

Only the pixel data from the area of overlap between the Auto Function AOI defined by your settings and the Image AOI will be used by the related auto function.

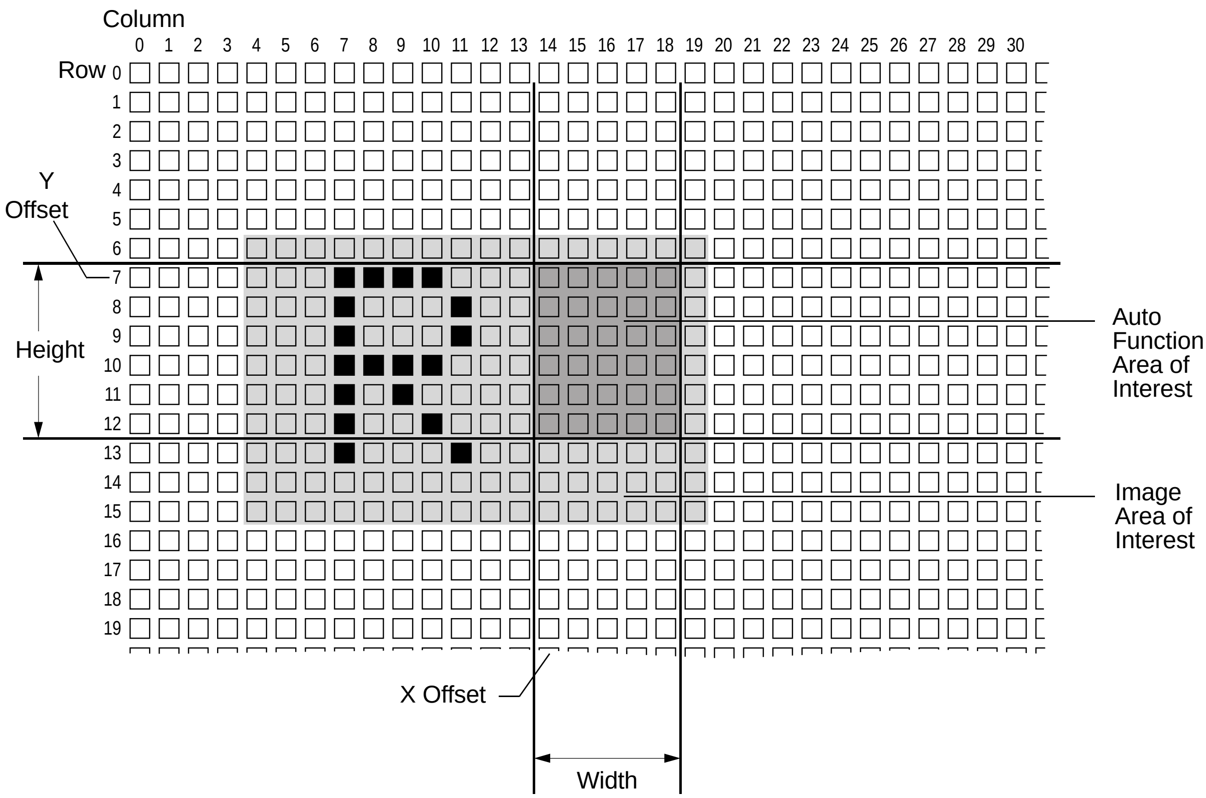


Fig. 111: Auto Function Area of Interest and Image Area of Interest

10.12.3.1 Assignment of an Auto Function to an Auto Function AOI

By default, the Gain Auto and the Exposure Auto auto functions are assigned to Auto Function AOI 1 and the Balance White Auto auto function is assigned to Auto Function AOI 2. The assignments can, however, be set as desired. For example, the Balance White Auto auto function can be assigned to Auto Function AOI 1 or all auto functions can be assigned to the same Auto Function AOI.

ⓘ	We strongly recommend not to assign an auto function to more than one Auto Function AOI although the assignment can be made.

One limitation must be borne in mind: For the purpose of making assignments, the Gain Auto and the Exposure Auto auto functions are always considered as a single "Intensity" auto function and therefore the assignment is always identical for both auto functions. For example, if you assign the

"Intensity" auto function to Auto Function AOI 2 the Gain Auto and the Exposure Auto auto functions are both assigned to Auto Function AOI 2. This does not imply, however, that the Gain Auto and the Exposure Auto auto functions must always be used at the same time.

You can assign auto functions to Auto Function AOIs from within your application software by using the pylon API.

As an example, the following code snippet illustrates using the API to assign the Gain Auto and Exposure Auto auto function - considered as a single "Intensity" auto function - and the Exposure Auto auto function to Auto Function AOI 1.

The snippet also illustrates disabling the unused Auto Function AOI 2 to avoid assigning any auto function to more than one Auto Function AOI.

```
// Select Auto Function AOI 1
// Assign auto functions to the selected Auto Function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
Camera.AutoFunctionAOIUsageIntensity.SetValue( true );
Camera.AutoFunctionAOIUsageWhiteBalance.SetValue( true );

// Select the unused Auto Function AOI 2
// Disable the unused Auto Function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI2 );
Camera.AutoFunctionAOIUsageIntensity.SetValue( false );
Camera.AutoFunctionAOIUsageWhiteBalance.SetValue( false );
```

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

10.12.3.2 Positioning of an Auto Function AOI Relative to the Image 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 112. 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 112), 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 112), 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 112), 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 112), 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.

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.

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.

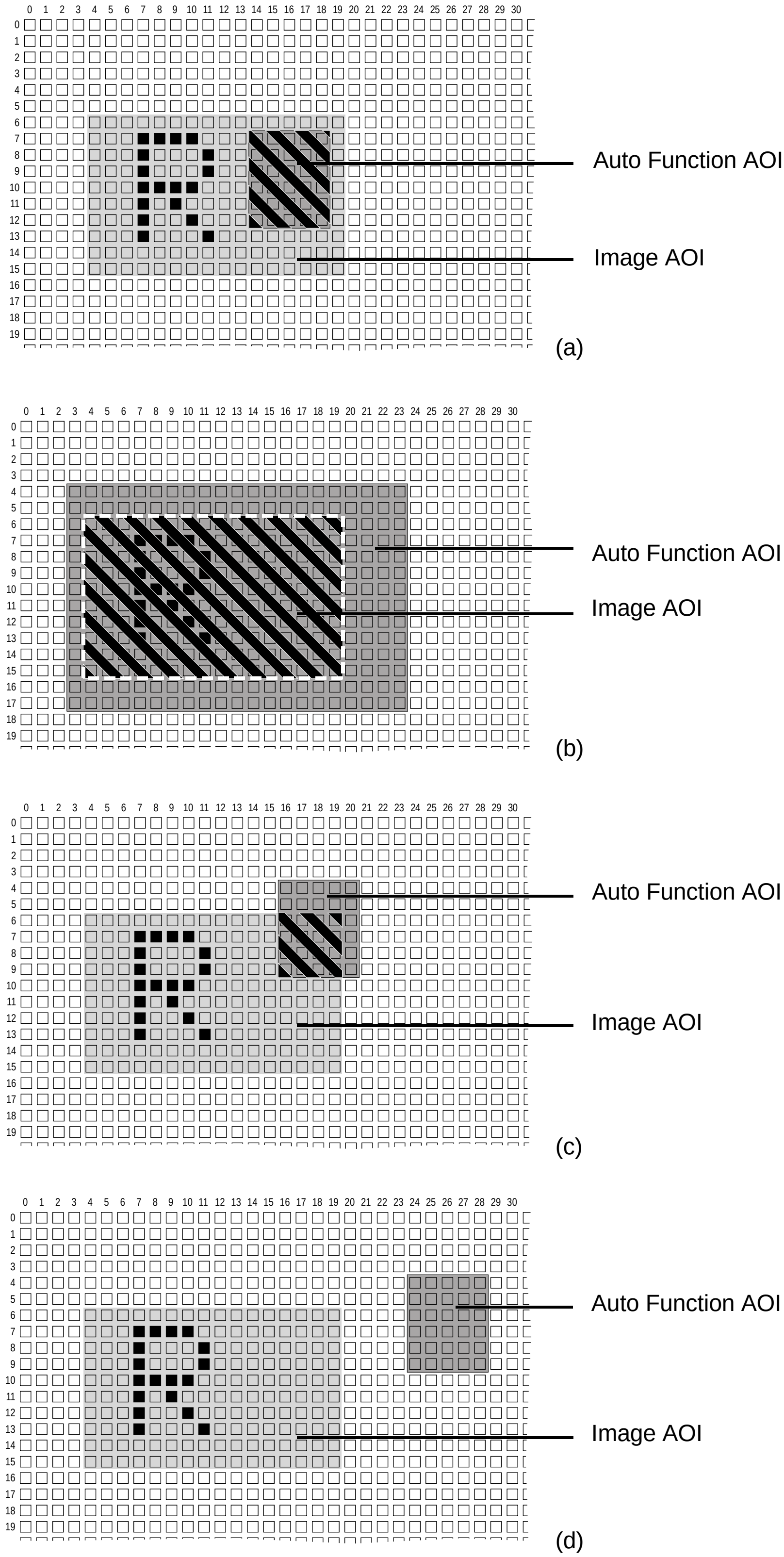


Fig. 112: Various Degrees of Overlap Between the Auto Function AOI and the Image AOI

10.12.3.3 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 row 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 acA640-100gm, 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 acA640-100gm, 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 color 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), 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 10.8.3 on page 270.

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() );
```

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 page 39.

10.12.4 Gain Auto

Gain Auto is the "automatic" counterpart to manually setting the Gain Raw parameter. When the gain auto function is operational, the camera will automatically adjust the Gain Raw parameter value within set limits until a target average gray value for the pixel data from the related Auto Function AOI is reached.

The gain auto function can be operated in the "once" and continuous" modes of operation.

If the related Auto Function AOI does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from the Auto Function AOI will not be used to control the gain. Instead, the current manual setting for the Gain Raw parameter value will control the gain.

The gain auto function and 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 setting the gain "manually", see Section 10.1 on page 209.

For more information about the auto function profile feature, see Section 10.12.7 on page 296.

The limits within which the camera will adjust the Gain Raw parameter are defined by the Auto Gain Raw Upper Limit and the Auto Gain Raw Lower Limit parameters. The minimum and maximum allowed settings for the Auto Gain Raw Upper Limit and Auto Gain Raw Lower Limit parameters depend on the current pixel data format, on the current settings for binning, and on whether or not the parameter limits for manually setting the gain feature are disabled.

The Auto Target Value parameter defines the target average gray value that the gain auto function will attempt to achieve when it is automatically adjusting the Gain Raw value. The target average gray value can range from 0 (black) to 255 (white) when the camera is set for an 8 bit pixel format or from 0 (black) to 4095 (white) when the camera is set for a 12 bit pixel format.

Setting the gain auto functionality using Basler pylon is a several step process:

- Select the Auto Function AOI 1.
- Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
- Set the Gain Selector to All.
- Set the value of the Auto Gain Raw Lower Limit and Auto Gain Raw Upper Limit parameters.
- Set the value of the Auto Target Value parameter.
- Set the value of the Gain Auto parameter for the "once" or the "continuous" mode of operation.

You can set the gain auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the exposure auto functionality:

```
// Select auto function AOI 1
// 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() );

    // Select gain all and set the upper and lower gain limits for the
    // gain auto function
    Camera.GainSelector.SetValue( GainSelector_All );
    Camera.AutoGainRawLowerLimit.SetValue( Camera.GainRaw.GetMin() );
    Camera.AutoGainRawUpperLimit.SetValue( Camera.GainRaw.GetMax() );

    // Set the target gray value for the gain auto function
  // (If exposure auto is enabled, this target is also used for
    //  exposure auto control.)
    Camera.AutoTargetValue.SetValue( 128 );

    // Set the mode of operation for the gain auto function
    Camera.GainAuto.SetValue( GainAuto_Once );
```

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 page 39.

For general information about auto functions, see Section 10.12 on page 282.

For information about Auto Function AOIs and how to set them, see Section 10.12.3 on page 284.

10.12.5 Exposure Auto

> The exposure auto function will not work, if the camera's exposure mode is set to trigger width. For more information about the trigger width exposure mode, see Section 7.4.3.2 on page 99.

Exposure Auto is the "automatic" counterpart to manually setting the Exposure Time Abs parameter. 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 AOI 1 is reached.

The exposure auto function can be operated in the "once" and continuous" modes of operation.

If Auto Function AOI 1 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI 1 will not be used to control the exposure time. Instead, the current manual setting of the Exposure Time Abs parameter value will control the exposure time.

The exposure auto function and the gain auto function can be used at the same time. In this case, however, you must also set the auto function profile feature.

When trigger width exposure mode is selected, the exposure auto function is not available.

For more information about setting the exposure time "manually", see Section 7.11 on page 140.

For more information about the trigger width exposure mode, see Section 7.4.3.2 on page 99.

For more information about the auto function profile feature, see Section 10.12.7 on page 296.

The limits within which the camera will adjust the Auto Exposure Time Abs parameter are defined by the Auto Exposure Time Abs Upper Limit and the Auto Exposure Time Abs Lower Limit parameters. The current minimum and the maximum allowed settings for the Auto Exposure Time Abs Upper Limit parameter and the Auto Exposure Time Abs Lower Limit parameters depend on the minimum allowed and maximum possible exposure time for your camera model.

The Auto Target Value parameter defines the target average gray value that the exposure auto function will attempt to achieve when it is automatically adjusting the Exposure Time Abs value. The target average gray value may range from 0 (black) to 255 (white) when the camera is set for an 8 bit pixel format or from 0 (black) to 4095 (white) when the camera is set for a 12 bit pixel format.

> If the Auto Exposure Time Abs Upper Limit parameter is set to a sufficiently high value the camera's frame rate may be decreased.

Setting the exposure auto functionality using Basler pylon is a several step process:

- Select the Auto Function AOI 1.
- Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
- Set the value of the Auto Exposure Time Abs Lower Limit and Auto Exposure Time Abs Upper Limit parameters.
- Set the value of the Auto Target Value parameter.
- Set the value of the Exposure Auto parameter for the "once" or the "continuous" mode of operation.

You can set the exposure auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the exposure auto functionality:

```
// Select auto function AOI 1
Camera.AutoFunctionAOISelector.SetValue(
AutoFunctionAOISelector_AOI1 );

// Set the position and size of the selected auto function AOI. In
this example,
// we set the auto function AOI to cover the entire sensor
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue(
Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue(
Camera.AutoFunctionAOIHeight.GetMax() );

// Set the exposure time limits for exposure auto control
Camera.AutoExposureTimeAbsLowerLimit.SetValue( 1000 );
Camera.AutoExposureTimeAbsUpperLimit.SetValue( 1.0E6 );

// Set the target gray value for the exposure auto function
// (If gain auto is enabled, this target is also used for
//  gain auto control.)
Camera.AutoTargetValue.SetValue( 128 );

// Set the mode of operation for the exposure auto function
Camera.ExposureAuto.SetValue( ExposureAuto_Continuous );
```

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 page 39.

For general information about auto functions, see Section 10.12 on page 282.

For information about Auto Function AOIs and how to set them, see Section 10.12.3 on page 284.

For information about minimum allowed and maximum possible exposure time, see Section 7.11 on page 140.

10.12.6 Gray Value Adjustment Damping

The gray value adjustment damping controls the rate by which pixel gray values are changed when Exposure Auto and/or Gain Auto are enabled.

If an adjustment damping factor is used, the gray value target value is not immediately reached, but after a certain "delay". This can be useful, for example, when objects move into the camera's view area and where the light conditions are gradually changing due to the moving objects.

By default, the gray value adjustment damping is set to 0.6836. This is a setting where the damping control is as stable and quick as possible.

Setting the Adjustment Damping

The gray value adjustment damping is determined by the value of the Gray Value Adjustment Damping Abs parameter. The parameter can be set in a range from 0.0 to 0.78125.

The higher the value, the lower the adjustment damping is, i.e.

- the sooner the target value will be reached,
- the adaptation is realized over a smaller number of frames.

Examples:
0.6836 = Default value the camera starts with. There is a relatively immediate continuous adaptation to the target gray value.
If you set the value to 0.5, there would be more interim steps; the target value would be reached after a "higher" number of frames.

You can set the gray value adjustment damping from within your application software by using the pylon API. The following code snippets illustrate using the API to set the gray value adjustment damping:

```
Camera.GrayValueAdjustmentDampingRaw.SetValue(600);
Camera.GrayValueAdjustmentDampingAbs.SetValue(0.5859);
```

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

10.12.7 Auto Function Profile

If you want to use the gain auto function and the exposure auto function at the same time, the auto function profile feature also takes effect. The auto function profile specifies whether the gain or the exposure time will be kept as low as possible when the camera is making automatic adjustments to achieve a target average gray value for the pixel data from the Auto Function AOI that was related to the gain auto and the exposure auto function. By default, the auto function profile feature minimizes gain.

If you want to use the gain auto and the exposure auto functions at the same time, you should set both functions for the continuous mode of operation.

Setting the camera with Basler pylon to use the gain auto function and the exposure auto function at the same time is a several step process:

- Set the value of the Auto Function Profile parameter to specify whether gain or exposure time will be minimized during automatic adjustments.
- Set the value of the Gain Auto parameter to the "continuous" mode of operation.
- Set the value of the Exposure Auto parameter to the "continuous" mode of operation.

You can set the auto function profile from within your application software by using the pylon API. The following code snippet illustrates using the API to set the auto function profile. As an example, Gain Auto is set to be minimized during adjustments:

```
// Use GainAuto and ExposureAuto simultaneously
Camera.AutoFunctionProfile.SetValue( AutoFunctionProfile_GainMinimum
);
Camera.GainAuto.SetValue( GainAuto_Continuous );
Camera.ExposureAuto.SetValue( ExposureAuto_Continuous );
```

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 page 39.

10.12.8 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 more information about setting the white balance "manually", see Section 8.2 on page 159.

Setting the balance white auto functionality using Basler pylon is a several step process:

- Select the Auto Function AOI 2.
- Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
- Set the value of the Exposure Auto parameter for the "once" or the "continuous" mode of operation.

You can set the white balance auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the balance auto functionality:

```
// Select auto function AOI 2
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI2
);

// Set the position and size of selected auto function AOI. In this
example, we set
// auto function AOI to cover the entire sensor.
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue(
Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue(
Camera.AutoFunctionAOIHeight.GetMax() );

// Set mode of operation for balance white auto function
Camera.BalanceWhiteAuto.SetValue( BalanceWhiteAuto_Once );
```

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 page 39.

For general information about auto functions, see Section 10.12 on page 282.

For information about Auto Function AOIs and how to set them, see Section 10.12.3 on page 284.

10.12.9 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. Set the GrayValueAdjustmentDampingAbs parameter.
8. If necessary, set the auto function profile to define priorities between auto functions.
9. Enable the auto function by setting it to "once" or "continuous".

For more information about the individual settings, see the previous sections that describe the individual auto functions.

10.13 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.

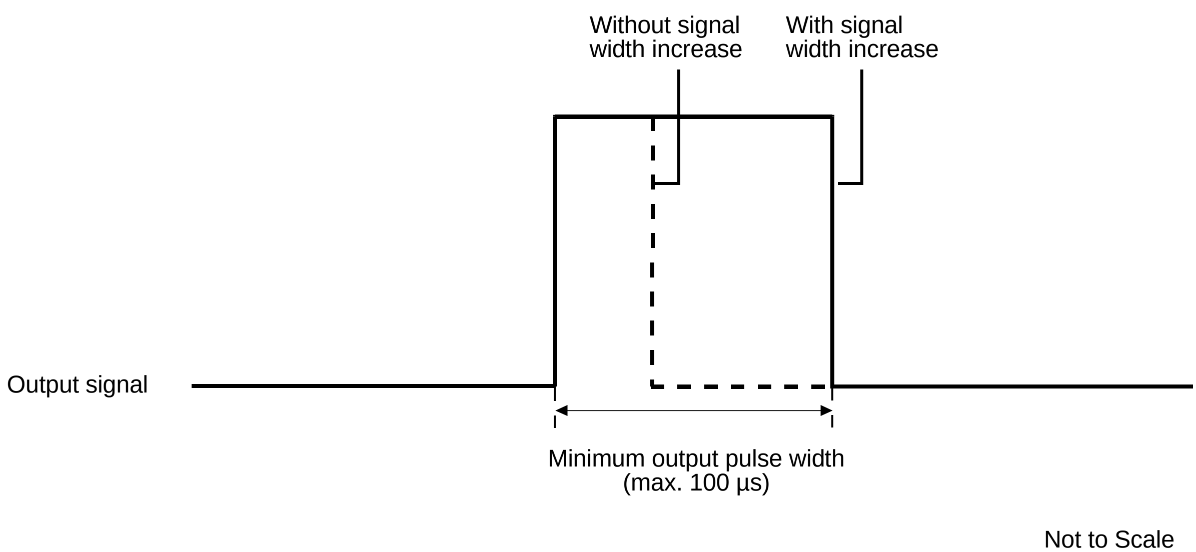


Fig. 113: Increasing the Signal Width of an Output Signal

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 1.
- 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.

For more information about the pylon Viewer, see Section 3.1.1 on page 40.

10.14 Event Reporting

Event reporting is available on the camera. With event reporting, the camera can generate an "event" and transmit a related event message to the PC whenever a specific situation has occurred.

The camera can generate and transmit events for the following types of situations:

- An acquisition start trigger has occured (AcquisitionStartEvent).
- Overtriggering of the acquisition start trigger has occurred (AcquisitionStartOvertriggerEventData).

  This happens, if the camera receives an acquisition start trigger signal when it is not in a "waiting for acquisition start" acquisition status.
- A frame start trigger has occured (FrameStartEvent).
- Overtriggering of the frame start trigger has occurred (FrameStartOvertriggerEventData).

  This happens, if the camera receives a frame start trigger signal when it is not in a "waiting for frame start trigger" acquisition status.
- The end of an exposure has occurred (ExposureEndEventData).
- An event overrun has occurred (EventOverrunEventData).

  This situation is explained later in this section.

An Example of Event Reporting

An example related to the Frame Start Overtrigger event illustrates how event reporting works. The example assumes that your system is set for event reporting (see below) and that the camera has received a frame start trigger when the camera is not in a "waiting for frame start trigger" acquisition status. In this case:

1. A Frame Start Overtrigger event is created. The event contains the event in the strict sense plus supplementary information:

   An *Event Type Identifier*. In this case, the identifier would show that a frame start overtrigger type event has 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 acknowledgement. If no acknowledgement is received within a specified timeout, the camera will resend the event message. If an acknowledgement is still not received, the timeout and resend mechanism will repeat until a specified maximum number of 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 acknowledgement, no new event messages can be transmitted.

4. Event reporting involves making some additional software-related steps and settings. 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, 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
Acquisition Start	AcquisitionStartEventData	AcquisitionStartEventStreamChannelIndex
		AcquisitionStartEventTimestamp
Acquisition Start Overtrigger	AcquisitionStartOvertriggerEventData	AcquisitionStartOvertriggerEventStreamChannelIndex
		AcquisitionStartOvertriggerEventTimestamp
Frame Start	FrameStartEventData	FrameStartEventStreamChannelIndex
		FrameStartEventTimestamp
Frame Start Overtrigger	FrameStartOvertriggerEventData	FrameStartOvertriggerEventStreamChannelIndex
		FrameStartOvertriggerEventTimestamp
Exposure End	ExposureEndEventData	ExposureEndEventFrameID
		ExposureEndEventStreamChannelIndex
		ExposureEndEventTimestamp
Event Overrun	EventOverrunEventData	EventOverrunEventStreamChannelIndex
		EventOverrunEventTimestamp

Table 26: 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. The pylon software development kit includes a "Grab_CameraEvents" code sample that illustrates the entire process.

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

10.15 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 );
```

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 page 39.

10.15.1 Test Image Descriptions

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 114.

The mathematical expression for this test image:

Gray Value = [column number + row number] MOD 256

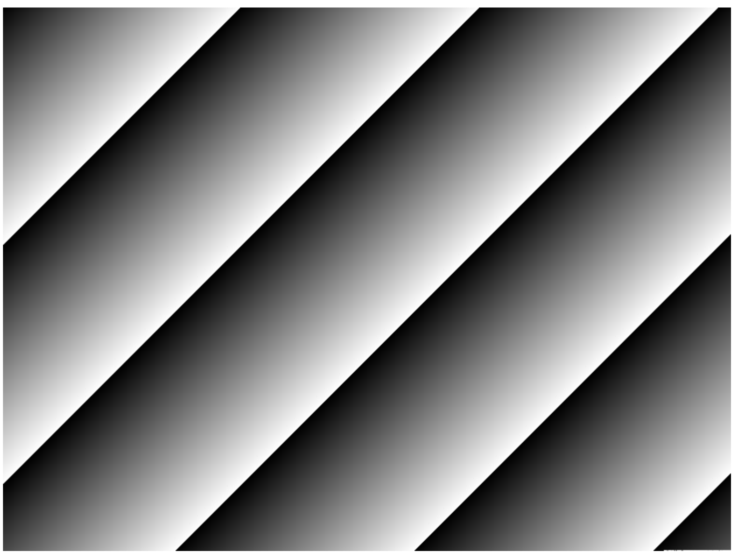


Fig. 114: Test Image One

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

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.

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 115, 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.

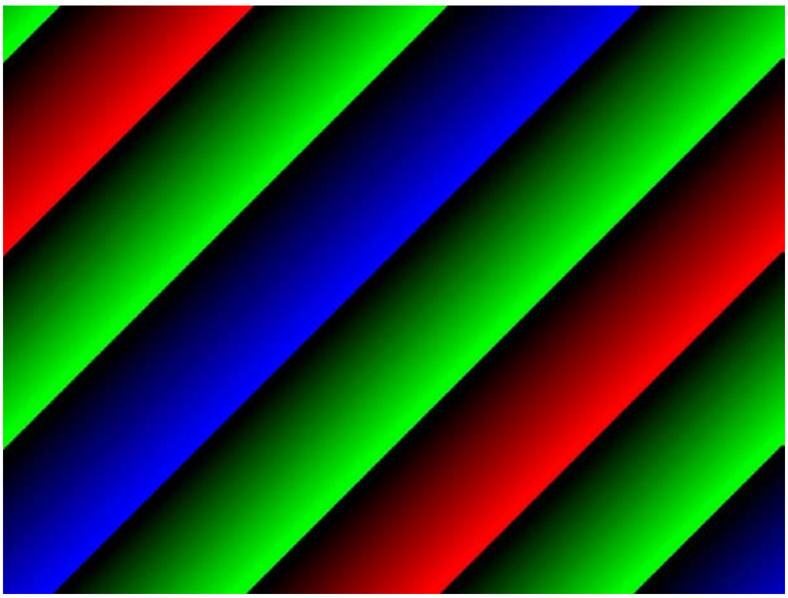


Fig. 115: Test Image Six

10.16 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 camera vendor's name.
- Device Model Name (read only) - contains the model name of the camera.
- 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.
- Device Firmware Version (read only) - contains the version of the firmware in the camera.
- Device ID (read only) - contains the serial number of the camera.
- Device User ID (read / write) - is used to assign a user defined name to a device. This name will be displayed in the Basler pylon Viewer and the Basler pylon IP Configuration Tool. The name will also be visible in the "friendly name" field of the device information objects returned by pylon's device enumeration procedure.
- Device Scan Type (read only) - contains the scan type of the camera, for example, area scan.
- Sensor Width (read only) - contains the physical width of the sensor in pixels.
- Sensor Height (read only) - contains the physical height of the sensor.
- Max Width (read only) - Indicates the camera's maximum area of interest (AOI) width setting.
- Max Height (read only) - Indicates the camera's maximum area of interest (AOI) height setting.

You can read the values for all of the device information parameters or set the value of the Device User ID parameter from within your application software by using the Basler 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();
```

You can also use the Basler pylon Viewer application to easily read the parameters and to read or write the Device User ID.

You can use the Basler pylon IP Configuration tool to read or write the Device User ID.

For more information about the pylon API, the pylon Viewer, and the pylon IP Configuration Tool, see Section 3 on page 39.

10.17 User Defined Values

The camera can store two "user defined values". These two values are 32 bit signed integer values that you can set and read as desired. They simply serve as convenient storage locations for the camera user and have no impact on the operation of the camera.

The two values are designated as Value 1 and Value 2.

Setting User Defined Values

Setting a user defined value using Basler pylon is a two step process:

- Set the User Defined Value Selector to Value 1 or Value 2.
- Set the User Defined Value parameter to the desired value for the selected value.

You can use the pylon API to set the User Defined Value Selector and the User Defined Value parameter value from within your application software. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set user defined value 1
Camera.UserDefinedValueSelector.SetValue(
UserDefinedValueSelector_Value1 );
Camera.UserDefinedValue.SetValue( 1000 );

// Set user defined value 2
Camera.UserDefinedValueSelector.SetValue(
UserDefinedValueSelector_Value2 );
Camera.UserDefinedValue.SetValue( 2000 );

// Get the value of user defined value 1
Camera.UserDefinedValueSelector.SetValue(
UserDefinedValueSelector_Value1 );
int64_t UserValue1 = Camera.UserDefinedValue.GetValue();
```

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

For more information about the Basler pylon API and the pylon Viewer, see Section 3 on page 39.

10.18 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.

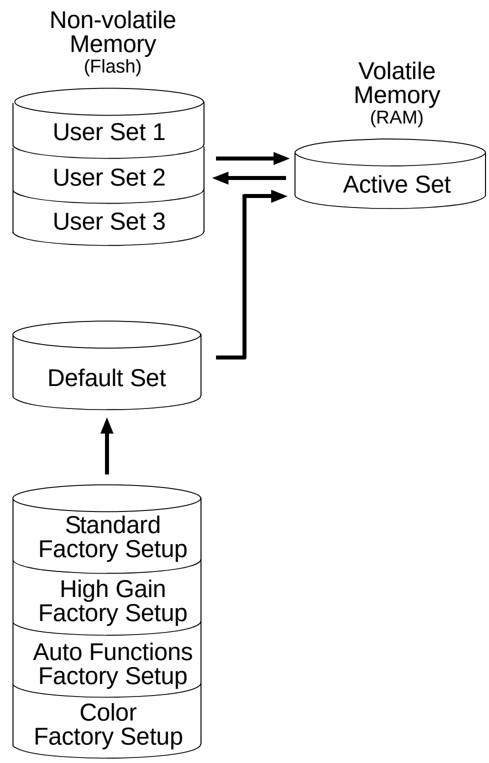


Fig. 116: Configuration Sets

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:

- 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 this 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 10.12 on page 282.

10.18.1 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.

For more information about the Basler pylon API and the pylon Viewer, see Section 3 on page 39.

10.18.2 Saving a User Set

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 Basler pylon API and the pylon Viewer, see Section 3 on page 39.

10.18.3 Loading the User 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.

For more information about the Basler pylon API and the pylon Viewer, see Section 3 on page 39.

10.18.4 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 );
```

For more information about the Basler pylon API and the pylon Viewer, see Section 3 on page 39.

11 Chunk Features

This section provides detailed information about the chunk features available on each camera.

11.1 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.

11.2 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).

11.3 Frame Counter

The rrame 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();
```

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 page 39.

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 I/O input line 1 or via software. You can also disable the ability to perform a reset. by setting the reset source to off. By default, 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 line1, 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 applied input line 1
Camera.CounterResetSource.SetValue( CounterResetSource_Line1 );

// Select reset by software
Camera.CounterResetSource.SetValue( CounterResetSource_Software );
```

```
// execute reset by software
Camera.CounterReset.Execute();

// Disable reset
Camera.CounterResetSource.SetValue( CounterResetSource_Off );
```

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 page 39.

For more information about using line 1 as the source signal for a frame counter reset, see Section 6.1.1 on page 67.

11.4 Time Stamp

The time stamp feature adds a chunk to each acquired frame containing a time stamp that was generated when frame 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 on.

ⓘ	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 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 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();
```

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 page 39.

11.5 Trigger Input Counter

The trigger input counter feature numbers external frame acquisition triggers sequentially as they are received. When the feature is enabled, a chunk is added to each image containing the 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 reaching the maximum value, the counter will reset to 0 and then continue counting.

Be aware that if the camera is operating with the frame trigger off, the trigger input counter will not be available.

> 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 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 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();
```

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 page 39.

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 line 1 or software. You can also disable the ability to perform a reset by setting the reset source to off. By default, trigger input counter reset is disabled.

To use the trigger input counter reset feature:

- 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, software, or 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 Basler 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 applied to input line 1
Camera.CounterResetSource.SetValue( CounterResetSource_Line1 );

// Select reset by software
```

```
Camera.CounterResetSource.SetValue( CounterResetSource_Software );
// execute reset by software
Camera.CounterReset.Execute();

// Disable reset
Camera.CounterResetSource.SetValue( CounterResetSource_Off );
```

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 page 39.

For more information about using line 1 as the source signal for a trigger input counter reset, see Section 6.1.1 on page 67.

11.6 Line Status All

The line status all feature samples the status of the camera's input line and output line each time a frame acquisition is triggered. It then adds a chunk to each acquired frame containing the line status information.

The line status all information is a 32 bit value. As shown in Figure 117, 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.

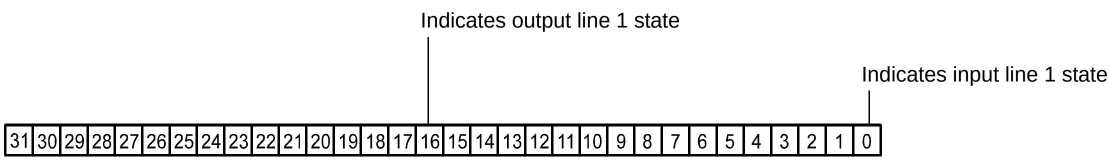


Fig. 117: Line Status All Parameter Bits

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 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 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();
```

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 page 39.

11.7 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.

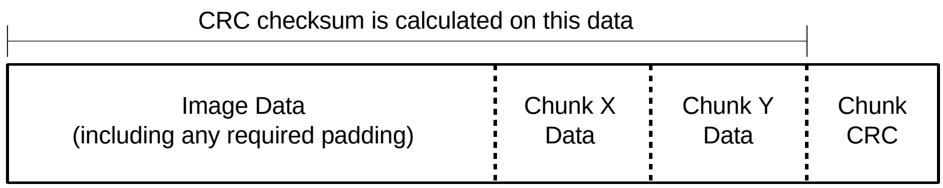


Fig. 118: CRC Checksum

> 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;
```

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 page 39.

11.8 Sequence Set Index

The sequence set index chunk feature adds a chunk to each acquired frame containing the index number of the sequence set that was used for frame acquisition.

ⓘ	The sequencer feature must be enabled before you can enable the sequence set index feature. For more information about the sequencer feature, see the "Sequencer" section.

ⓘ	The chunk mode must be active before you can enable the sequence set index feature or any of the other chunk features. 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.

12 Troubleshooting and Support

This chapter outlines the resources available to you, if you need help working with your camera.

12.1 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.

12.2 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

12.3 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 occur?	☐ Once. ☐ Regularly when: __________ ☐ Occasionally when: __________ __________ __________ __________	☐ Every time.
8	How severe is the problem?	☐ Camera can still be used. ☐ Camera can be used after I take this action: __________ __________ ☐ Camera can no longer be used.	
9	Did your application ever run without problems?	☐ Yes	☐ No

10 Parameter set

It is very important for Basler technical support to get a copy of the exact camera parameters that you were using when the problem occurred.

To make note of the parameters, use Basler’s 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.

Appendix A
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, see the Installation and Setup Guide for Cameras Used with Basler's pylon API (AW000611xx000).

For more information about installing the network drivers, see the Installation and Setup Guide for Cameras Used with Basler's pylon API (AW000611xx000).

.

A.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. Make sure 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 );
```

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 page 39.

A.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 Basler's pylon API* (AW000611xx000).

.

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.

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.

A.2.1 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.

A.2.2 Threshold Resend Mechanism Parameters

The threshold resend request mechanism is illustrated in Figure 119 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.

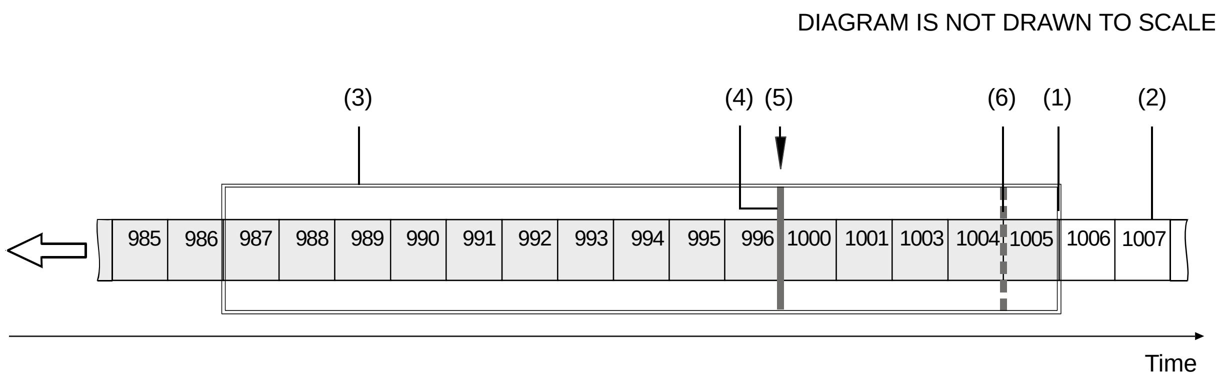


Fig. 119: Example of a Receive Window with Resend Request Threshold & Resend Request Batching Threshold

(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 119. The parameter value is in per cent of the width of the receive window. In Figure 119 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 119). 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 119, 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 119). 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 119 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 119). 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).

A.2.3 Timeout Resend Mechanism Parameters

The timeout resend mechanism is illustrated in Figure 120 where the following assumptions are made:

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

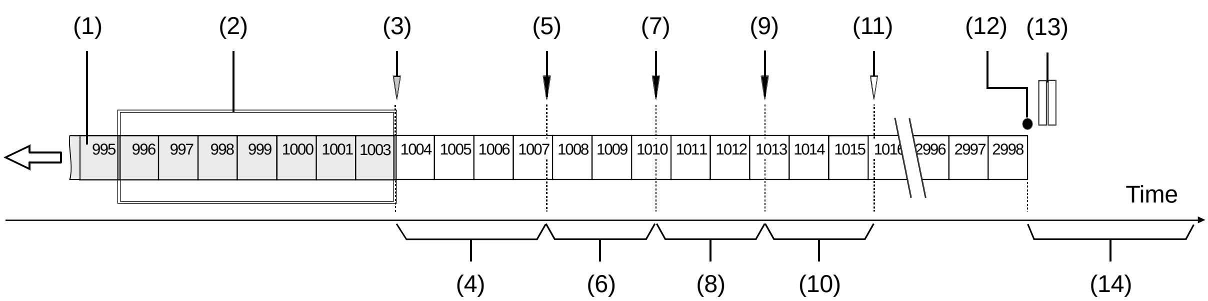


Fig. 120: Incomplete Stream of Packets and Part of the Resend Mechanism

(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.

A.2.4 Threshold and Timeout Resend Mechanisms Combined

Figure 121 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 121). In addition, resend request batching will not occur.

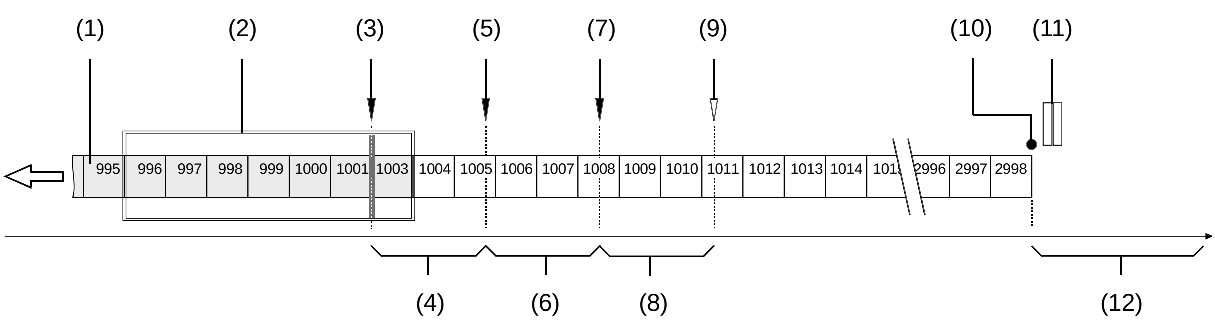


Fig. 121: Combination of Threshold Resend Mechanism and Timeout Resend Mechanism

(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 );
```

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 API and the pylon Viewer, see Section 3 on page 39.

A.2.5 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.

A.2.6 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.

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
```

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 page 39.

Appendix B 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.

B.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 using 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:

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

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

Y = number of bytes in each packet

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

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

Bandwidth Reserve (read/write)

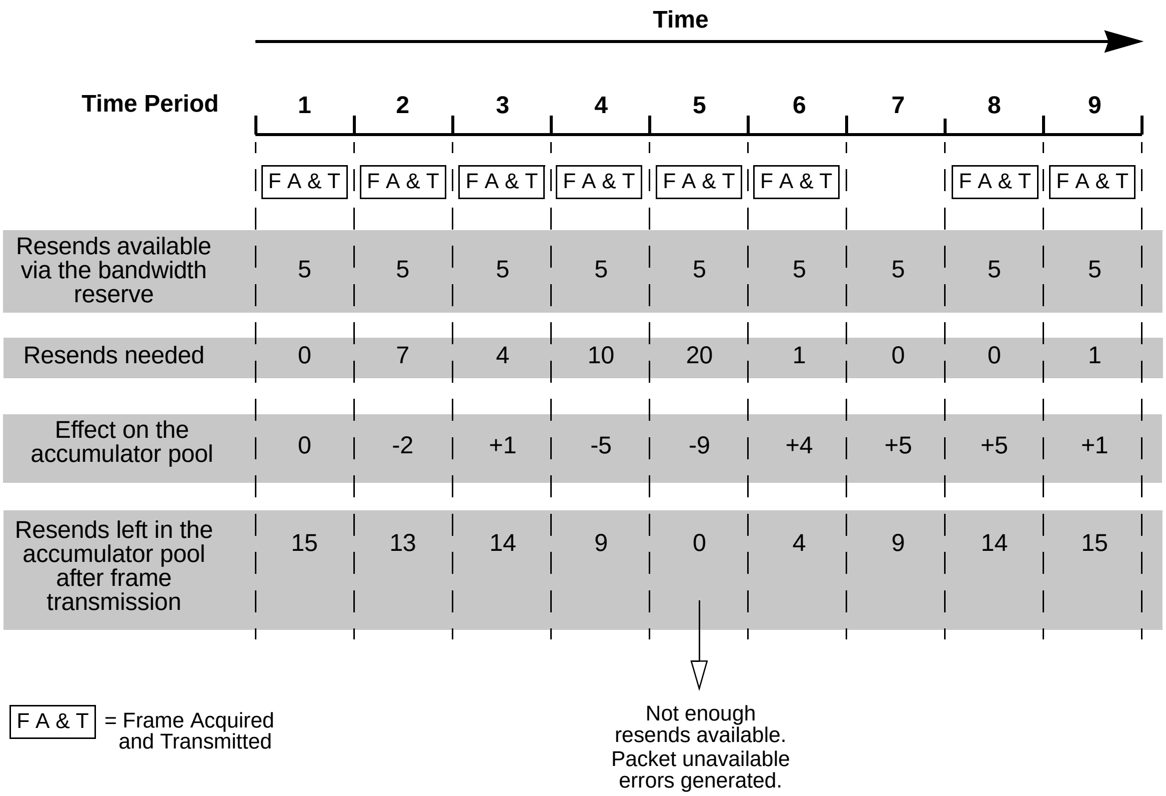
Used to reserve a portion of the assigned bandwidth for packet resends and for the transmission of control data between the camera and the host PC. The setting is expressed as a percentage of the Bandwidth Assigned parameter. For example, if the Bandwidth Assigned parameter indicates that 30 MByte/s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

Bandwidth Reserve Accumulation (read/write)

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

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

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



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

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

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

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

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

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

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

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

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

Frame Max Jitter (read only)

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

Device Max Throughput (read only)

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

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

Device Current Throughput (read only)

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

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

Note that the Device Current Throughput parameter indicates the bandwidth needed to transmit the actual image data and chunk data. The Bandwidth Assigned parameter, on the other hand, indicates the bandwidth needed to transmit image data and chunk data plus the bandwidth reserved for 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();
```

You can also use the Basler pylon Viewer application to easily set or view the parameter values.

For more information about the pylon API and the pylon Viewer, see Section 3 on page 39.

B.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 122.

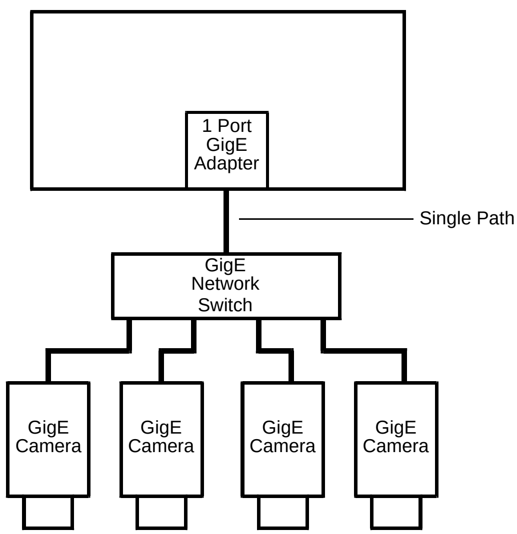


Fig. 122: Multiple Cameras on a Network

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.

B.3 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.

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

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

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 7.12 on page 143.

For more information about the AOI, see Section 10.5 on page 225.

Revision History

Doc. ID Number	Date	Changes
AW00089301000	8 Feb 2010	This release is a preliminary version of the document.
AW00089302000	9 Mar 2010	Indicated that UL certification was in preparation and corrected the camera weight specified in the specification tables in Section 1 on page 1. Corrected the voltages stated in the "Voltages outside of specified range can cause damage" notice box in Section 1.8 on page 34. The status of this document remains preliminary.
AW00089303000	30 Jul 2010	Made the appropriate changes throughout the manual to add the new acA750-30gm camera model. Updated Section 5.8.2 on page 62 to describe the current behavior of the output line. Updated the entire contents of Section 7 on page 79 to more completely describe the acquisition control options.
AW00089304000	28 Sep 2010	Corrected several typographical errors in Section 7 on page 79.
AW00089305000	30 Nov 2010	Made the appropriate changes throughout the manual to add the new acA750-30gc camera model. Added Section 6.1.3 on page 69 to describe the input line invert function. Added Section 7.10 on page 127 to describe the tools available for monitoring acquisition. Added Section 10.17 on page 310 to describe the user defined values feature.
AW00089306000	16 Dec 2010	Made the appropriate changes throughout the manual to add the new acA2500-14gm/gc camera models.
AW00089307000	4 Feb 2011	Corrected timing and parameter values stated for the acA2500-14gm/gc camera models.
AW00089308000	4 Apr 2011	Made the appropriate changes throughout the manual to add the new acA640-90gm/gc camera models.
AW00089309000	5 Apr 2011	Corrected an omission in the sensor size listings for the camera specifications.
AW00089310000	6 Jun 2011	Made the appropriate changes throughout the manual to add the new acA1600-20gm/gc camera models.

Doc. ID Number	Date	Changes
AW00089**11**000	19 Aug 2011	Added mechanical stress test results in Section 1.4.3 on page 30. Updated the descriptions of matrix color transformation and color adjustments in Section 8.4 on page 164 and Section 10.18 on page 311. Removed the Tungsten and Daylight 5000K Color Transformation Selector parameters from Section 8.4.4 on page 172. Added Section 10.7 on page 233 describing the sequencer feature. Added a note that auto functions will not work when the sequencer is enabled in Section 10.12 on page 282. Added the color factory setup in Section 10.7 on page 233. Added Section 11.8 on page 323 describing the sequence set index chunk and modified Section 11.2 on page 310 accordingly.
AW00089**12**000	16 Dec 2011	Indicated Basler AG as bearer of the copyright on the back of the front page. Corrected the pixel size for the acA640-90gm/gc camera in Section 1.2 on page 2. Modified and extended the description of the sequencer feature in Section 10.7 on page 233.

AW00089**13**000	27 Jan 2012	▪ Integrated new sensors ICX414 and ICX415 in the following sections:
		  ▪ Section 1.2 on page 2
		  ▪ Section 1.3 on page 15
		  ▪ Section 7.11 on page 140
		  ▪ Section 9.1 on page 183
		  ▪ Section 10.1.1 on page 210
		  ▪ Section 10.2.1 on page 216
		▪ Integrated CS-mount for Basler ace GigE cameras:
		  ▪ Section 1.2 on page 2
		  ▪ Section 1.4.1 on page 26
		  ▪ Section 1.4.2 on page 28
		▪ Updated 'Standard Power and I/O Cable' drawing in Section 5.4.2 on page 52: added shield.
		▪ Adapted the frame start trigger delay range from 10 s to 1 s in Section 7.4.3.3 on page 101.
		▪ Entered maximum exposure time for the acA645-100gm/gc and for the acA780-75gm/gc in the table in Section 7.6 on page 109
		▪ Modified maximum exposure time for the acA1300-30gm/gc in the table in Section 7.6 on page 109.
		▪ Updated minimum delay values for the acA2500- 14gm/gc in the table in Section 7.11 on page 140.
		▪ Entered new section on events used for acquisition monitoring, see Section 7.10.5 on page 139.
		▪ Changed 'Bayer filter pattern' to 'color filter pattern' in Section 10.12.3 on page 284.
		▪ Integrated note on Sequence Set Index chunk in Section 10.7 on page 233.
		▪ Adapted vertical binning description for the acA 2500-14 gm in Section 10.8 on page 266.
		▪ Integrated two new events (acquisition start and frame start) in Section 10.14 on page 301.

Doc. ID Number	Date	Changes

AW00089**14**000	30 Apr 2012	**Section 1**
		■ Integrated the new CMOSIS sensors CMV2000-2Exx and CMV4000-2Exx for mono, color and mono NIR in the technical specifications tables in Section 1.2 on page 2.
		■ Integrated the spectral response curves in Section 1.3 on page 15.
		Section 4
		Integrated functional description of the acA2000-50 and acA2040-25 models in Section 4.2 on page 44.
		Section 5
		Changed the 270 Ω value to 510 Ω in Figure 38 on page 63.
		Section 7
		■ Integrated the new camera models based on the CMOSIS sensors CMV2000-2Exx and CMV4000-2Exx for mono, color and mono NIR in in Section 7.6 on page 109 and in Section 7.11 on page 140.
		■ Changed subtitle of Figure 58 on page 115 from 'Rolling Shutter in the Global Release Mode' to 'Rolling Shutter in the Global Reset Release Mode'.
		■ Integrated note on the slowdown of the sequencer feature for the acA2500-14 in Section 7.9 on page 123.
		Section 8
		Integrated a table showing the Bayer filter color alignment for the different camera models.
		Section 9
		Integrated the new camera models based on the CMOSIS sensors CMV2000-2Exx and CMV4000-2Exx for mono, color and mono NIR in in Section 9.1 on page 183.
		Section 10
		■ Integrated the new camera models based on the CMOSIS sensors CMV2000-2Exx and CMV4000-2Exx for mono, color and mono NIR in
		  ■ Section 10.1.1 on page 210
		  ■ Section 10.2.1 on page 216
		  ■ Section 10.5 on page 225
		■ Integrated minimum output pulse width feature, see Section 10.13 on page 299.
		■ Integrated note on the slowdown of the sequencer feature for the acA2500-14 in Section 10.7.
		Section 11
		Replaced Z-modem by X-modem in Section 11.7 on page 321.

Doc. ID Number	Date	Changes

AW00089**15**000	6 Jun 2012	**Section 5**
		▪ Replaced figures in Section 5.7 on page 57 by simplified versions.
		▪ Replaced figures in Section 5.8 on page 62 by simplified versions.
		▪ Added notes in Section 5.5 on page 55, Section 5.7.1 on page 57, Section 5.8.1 on page 62:
		  ▪ that the recommended voltage range for camera power differs from the voltage ranges for the input and output lines and
		  ▪ that the recommended voltage range for camera power for Basler ace GigE cameras differ from the recommended voltage range for camera power for other Basler cameras
		Section 10
		▪ Added Section 10.6 on page 228 describing the stacked zone imaging feature.
		▪ Integrated the color binning feature for the acA2500-14gc (Section 10.8.2 on page 268).
		▪ Integrated section on binning's effect on the stacked zone imaging feature in Section 10.8.3 on page 270.

AW00089**16**000	8 Aug 2012	**General** Corrected the pixel data format for the acA2000-50 and acA2040-25 camera models throughout the manual (changed BG to GR). **Section 1** ▪ Integrated the new CMOSIS sensors CMV2000-2Exx and CMV4000-2Exx for mono, color and mono NIR in the technical specifications tables in Section 1.2 on page 2. ▪ Added front view of CMOSIS cameras in mechanical drawings in Section 1.4 on page 25. **Section 5** ▪ Inserted note that power to the camera must be supplied either via Power over Ethernet (PoE) or via the camera's 6-pin connector. ▪ Corrected resistor symbol in Figure 34 on page 58. **Section 6** ▪ Corrected description of Section 6.2.4.3 on page 76. ▪ Modified the Line Debouncer Time Abs parameter value to 20 µs in Section 6.1.2 on page 68. **Section 7** Inserted information in Section 7.12 that the stacked zone imaging feature increases the camera's frame rate. **Section 9** Replaced the "For each pixel covered with a red/green/blue **lens**..." expression by the "For each pixel covered with a red/green/blue **filter**..." throughout this section. To be continued.

Doc. ID Number	Date	Changes

AW00089**16**000	8 Aug 2012	**Section 10**
		■ Updated the minimum gain setting for the acA2000 and acA2040 models in Table 19 on page 212 and Table 20 on page 213.
		■ Inserted information in Section 10.6 on page 228 that the stacked zone imaging feature on the acA2000-50 and acA2040-25 increases the camera's frame rate.
		■ In the sequencer feature Section 10.7:
		  ■ Entered sequencer parameters that are stored in the factory set.
		  ■ Added that overlapped operation is not possible for the acA2500-14 with the sequencer feature enabled.
		■ Added the vertical decimation feature in Section 10.9 on page 272.
		■ Added Section 10.12.3.1 on page 285 describing the assignment of an auto function to an auto function AOI.
		■ Added the gray value adjustment damping feature in Section 10.12.6 on page 295.

Index

A

B

C

D

E

F

T

U

V

W

Z