Basler ace

USER'S MANUAL FOR GigE CAMERAS

Document Number: AW000893
Version: 21 Language: 000 (English)
Release Date: 16 October 2015

The manual includes information about the following prototype cameras: acA640-300, acA800-200, acA1300-75

BASLER
the power of sight

For customers in the USA

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

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

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

For customers in Canada

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

Pour utilisateurs au Canada

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

Life support applications

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

Warranty Information

To ensure that your warranty remains in force, adhere to the following guidelines:

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.

Prevent ingress or insertion of foreign substances into the camera housing

Prevent liquid, flammable, or metallic substances from entering the camera housing. If operated with any foreign substances 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 in original packaging

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

Clean with care

Avoid cleaning the sensor if possible. If you must clean it, follow the guidelines in the notice on page 61. This notice also provides information on cleaning the housing.

Read the manual

Read the manual carefully before using the camera.

Contacting Basler Support Worldwide

Europe, Middle East, Africa

Basler AG
An der Strusbek 60–62
22926 Ahrensburg
Germany

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

support.europe@baslerweb.com

The Americas

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

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

support.usa@baslerweb.com

Asia-Pacific

Basler Asia Pte. Ltd.
35 Marsiling Industrial Estate Road 3
#05–06
Singapore 739257

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1 Specifications, Requirements, and Precautions

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

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

1.1 Models

The current Basler 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 resolution, 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

1.2.1 Cameras with CCD Sensor

Specification	acA640-90gm/gc	acA640-120gm/gc
Resolution (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"
Effective Sensor Diagonal	6.1 mm	4.6 mm
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	120 fps
Mono/Color	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 hardware trigger, via software trigger, or free run	
Exposure Time Control	Via hardware trigger or programmable via the camera API	

Table 1: General Specifications (acA640-90gm/gc, acA640-120gm/gc)

Specification	acA640-90gm/gc	acA640-120gm/gc
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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 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.
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line. Power supplies must meet the SELV and LPS requirements (see page 59).	
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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 1: General Specifications (acA640-90gm/gc, acA640-120gm/gc)

Specification	acA645-100gm/gc	acA750-30gm/gc
Resolution (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"
Effective Sensor Diagonal	8.2 mm	6.2 mm
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)	
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	Color Models: Mono 8 YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	12 bits	
Synchronization	Via hardware trigger, via software trigger, or free run	
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 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 Power supplies must meet the SELV and LPS requirements (see page 59).	

Table 2: General Specifications (acA645-100gm/gc, acA750-30gm/gc)

Specification	acA645-100gm/gc	acA750-30gm/gc
Lens Adapter	C-mount; CS-mount (only available for color models)	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)	
Weight	< 90 g	
Conformity	CE, UL, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 2: General Specifications (acA645-100gm/gc, acA750-30gm/gc)

Specification	acA780-75gm/gc	acA1300-22gm/gc
Resolution (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"
Effective Sensor Diagonal	8.3 mm	6.1 mm
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	22 fps
Mono/Color	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 hardware trigger, via software trigger, or free run	
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 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. Power supplies must meet the SELV and LPS requirements (see page 59).	

Table 3: General Specifications (acA780-75gm/gc, acA1300-22gm/gc)

Specification	acA780-75gm/gc	acA1300-22gm/gc
Lens Adapter	C-mount; CS-mount (only available for color models)	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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 3: General Specifications (acA780-75gm/gc, acA1300-22gm/gc)

Specification	acA1300-30gm/gc	acA1600-20gm/gc
Resolution (H x V pixels)	gm: 1296 x 966 gc: 1294 x 964	gm: 1626 x 1236 gc: 1624 x 1234
Sensor Type	Sony ICX445 AL/AQ Progressive scan CCD Global shutter	Sony ICX274 AL/AQ Progressive scan CCD Global shutter
Optical Size	1/3"	1/1.8"
Effective Sensor Diagonal	6.1 mm	8.9 mm
Pixel Size	3.75 µm x 3.75 µm	4.4 µm x 4.4 µm
Max. Frame Rate (at full resolution)	30 fps	20 fps
Mono/Color	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 hardware trigger, via software trigger, or free run	
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 2.5 W when using Power over Ethernet ≈ 2.2 W @ 12 VDC when supplied via the camera's 6-pin connector	≈ 3.4 W when using Power over Ethernet ≈ 2.9 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. Power supplies must meet the SELV and LPS requirements (see page 59).	
Lens Adapter	C-mount; CS-mount	C-mount; CS-mount (only available for mono models)
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 4: General Specifications (acA1300-30gm/gc, acA1600-20gm/gc)

Specification	acA1300-30gm/gc	acA1600-20gm/gc
Weight	< 90 g	
Conformity	CE, UL, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 4: General Specifications (acA1300-30gm/gc, acA1600-20gm/gc)

1.2.2 Cameras with CMOS Sensors

Camera Models with Full and Default Resolution

Some camera models have, after initial camera start up, a slightly reduced resolution available (marked as default resolution in the following tables).
If you want to use the maximum resolution of the camera, you can obtain it by setting the offset values (Offset X and Offset Y) to 0 and the image AOI to the maximum possible pixels. The maximum resolution is marked as full resolution.

In the following tables camera models with full and default resolution are marked with two asteriks ** in the "Resolution..." lines.

Camera Models with Sensor Readout Modes

Some camera models have two sensor readout modes (settable via software):

- **normal readout mode**:
  In this mode the camera delivers images at a normal frame rate.
- **fast readout mode**:
  In this mode the camera delivers images at higher frame rates.

In the following tables the camera models with two sensor readout modes are marked with three asteriks *** in the "Max. Frame Rate" lines.

For more information about the sensor readout modes, see Section 6.12.2.1 on page 199.

Specification	acA640-300gm/gc	acA800-200gm/gc
Resolution (H x V pixels) **	gm/gc: 672 x 512 (full resolution) 640 x 480 (default resolution)	gm/gc: 832 x 632 (full resolution) 800 x 600 (default resolution)
Sensor Type	ON Semiconductor® PYTHON NOIP1SN0300A/ PYTHON NOIP1SE0300A Progressive scan CMOS Global shutter	ON Semiconductor® PYTHON NOIP1SN0500A/ PYTHON NOIP1SE0500A Progressive scan CMOS Global shutter
Optical Size	1/4"	1/3.3"
Effective Sensor Diagonal	3.9 mm	4.8 mm
Pixel Size (H x V)	4.8 µm x 4.8 µm	
Max. Frame Rate (at default resolution) ***	347 fps (at fast sensor readout) 277 fps (at normal sensor readout)	222 fps (at fast sensor readout) 197 fps (at normal sensor readout)
Mono/Color	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)	

Table 5: General Specifications (acA640-300gm/gc, acA800-200gm/gc)

Specification	acA640-300gm/gc	acA800-200gm/gc
Pixel Data Formats	Mono Models: Mono 8 Mono 10	Mono 10p
	Color Models: Mono 8 Bayer BG 8* Bayer BG10* Bayer BG 10p*	YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed
	* If you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change into Bayer BG, GR or RG as indicated in Section 9.12 on page 319.	
Synchronization	Via hardware trigger, via software trigger, or free run	
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 3.4 W when using Power over Ethernet ≈ 3.0 W @ 12 VDC when supplied via the camera's 6-pin connector	
I/O Ports	1 opto-isolated input line, 1 opto-isolated output line. 1 GPIO (can be set to operate as an input or an output). Power supplies must meet the SELV and LPS requirements (see page 59).	
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)	
Weight	< 90 g	
Conformity	CE, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 5: General Specifications (acA640-300gm/gc, acA800-200gm/gc)

Specification	acA1280-60gm/gc
Resolution (H x V pixels)	gm: 1282 x 1026 gc: 1280 x 1024
Sensor Type	gm: e2V EV76C560 ABT gc: e2V EV76C560 ACT Progressive scan CMOS Rolling shutter
Optical Size	1/1.8"
Effective Sensor Diagonal	8.7 mm
Pixel Size (H x V)	5.3 µm x 5.3 µm
Max. Frame Rate (at full resolution)	gm: 60 fps gc: 60 fps (only, if camera is set for Bayer RG 8 format and if GigE connection does not limit the frame rate)
Mono/Color	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 [(*) 12-bit image data based on 10-bit sensor data.]
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m
	≈ 2.4 W when using Power over Ethernet ≈ 2.0 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. Power supplies must meet the SELV and LPS requirements (see page 59).
Lens Adapter	C-mount

Table 6: acA1280-60gm/gc

Specification	acA1280-60gm/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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).

Table 6: acA1280-60gm/gc

Specification	acA1300-60gm/gc	acA1300-60gmNIR
Resolution (H x V pixels)	gm: 1282 x 1026 gc: 1280 x 1024	
Sensor Type	gm: e2V EV76C560 ABT gc: e2V EV76C560 ACT Progressive scan CMOS Global shutter Rolling shutter The shutter mode can be set via the software.	e2V EV76C661 ABT Progressive scan CMOS Global shutter Rolling shutter The shutter mode can be set via the software.
Optical Size	1/1.8"	
Effective Sensor Diagonal	8.7 mm	
Pixel Size (H x V)	5.3 µm x 5.3 µm	
Max. Frame Rate (at full resolution)	gm: 60 fps * gc: 60 fps * * only, if camera is set for an 8-bit pixel format (e.g. Bayer RG 8) and if GigE connection does not limit the frame rate)	gmNIR: 60 fps * * only, if camera is set for an 8-bit pixel format (e.g. Bayer RG 8) and if GigE connection does not limit the frame rate)
Mono/Color	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 RG 8 Bayer RG 12 (*) Bayer RG 12 Packed (*) YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed [(*) 12-bit image data based on 10-bit sensor data.]	-
Synchronization	Via hardware trigger, via software trigger, or free run	
Exposure Time Control	Programmable via the camera API	

Table 7: General Specifications (acA1300-60gm/gc, acA1300-60gmNIR)

Specification	acA1300-60gm/gc	acA1300-60gmNIR
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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 2.4 W when using Power over Ethernet ≈ 2.0 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. Power supplies must meet the SELV and LPS requirements (see page 59).	
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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 7: General Specifications (acA1300-60gm/gc, acA1300-60gmNIR)

Specification	acA1300-75gm/gc
Resolution (H x V pixels)	gm/gc: 1280 x 1024
Sensor Type	ON Semiconductor® PYTHON NOIP1SN1300A/ PYTHON NOIP1SE1300A Progressive scan CMOS Global shutter
Optical Size	1/2 "
Effective Sensor Diagonal	7.9 mm
Pixel Size	4.8 µm x 4.8 µm
Max. Frame Rate (at full resolution) ***	81 fps (at fast sensor readout and at normal sensor readout)
Mono/Color	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 10
	Color Models: Mono 8 — YUV 4:2:2 Packed Bayer BG 8* — YUV 4:2:2 (YUYV) Packed Bayer BG 10* * If you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change into Bayer BG, GR or RG as indicated in Section 9.12 on page 319.
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m
	≈ 3.4 W when using Power over Ethernet ≈ 3.0 W @ 12 VDC when supplied via the camera's 6-pin connector
I/O Ports	1 opto-isolated input line, 1 opto-isolated output line. 1 GPIO (can be set to operate as an input or an output). Power supplies must meet the SELV and LPS requirements (see page 59).
Lens Adapter	C-mount

Table 8: General Specifications (acA1300-75gm/gc)

Specification	acA1300-75gm/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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).

Table 8: General Specifications (acA1300-75gm/gc)

Specification	acA1600-60gm/gc	acA1920-25gm/gc
Resolution (H x V pixels)	gm: 1602 x 1202 gc: 1600 x 1200	gm: 1920 x 1080 gc: 1920 x 1080
Sensor Type	gm: e2V EV76C570 ABT gc: e2V EV76C570 ACT Progressive scan CMOS Global shutter Rolling shutter The shutter mode can be set via the software.	Aptina MT9P031 Progressive scan CMOS Rolling shutter
Optical Size	1/1.8"	1/3.7"
Effective Sensor Diagonal	9.0 mm	4.85 mm
Pixel Size	4.5 µm x 4.5 µm	2.2 µm x 2.2 µm
Max. Frame Rate (at full resolution)	gm: 60 fps * gc: 60 fps * * only, if camera is set for an 8-bit pixel format (e.g. Bayer RG 8) and if GigE connection does not limit the frame rate)	25 fps
Mono/Color	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 RG 8 Bayer RG 12 (*) Bayer RG 12 Packed (*) YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed [(*) 12-bit image data based on 10-bit sensor data.]	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
Synchronization	Via hardware trigger, via software trigger, or free run	
Exposure Time Control	Programmable via the camera API	Via hardware trigger or programmable via the camera API

Table 9: General Specifications (acA1600-60gm/gc, acA1920-25gm/gc)

Specification	acA1600-60gm/gc	acA1920-25gm/gc
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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 2.5 W when using Power over Ethernet ≈ 2.1 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. Power supplies must meet the SELV and LPS requirements (see page 59).	
Lens Adapter	C-mount; CS-mount (only available for color models)	
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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 9: General Specifications (acA1600-60gm/gc, acA1920-25gm/gc)

Specification	acA1920-40m/gc
Resolution (H x V pixels) **	gm/gc: 1936 x 1216 (full resolution) 1920 x 1200 (default resolution)
Sensor Type	Sony IMX249LLJ-C Progressive scan CMOS Global shutter
Optical Size	1/1.2 "
Effective Sensor Diagonal	13.3 mm
Pixel Size	5.86 µm x 5.86 µm
Max. Frame Rate (at default resolution)	42 fps
Mono/Color	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
	Color Models: Mono 8 Bayer RG 8 Bayer RG 12 Bayer RG 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed * If you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change into Bayer BG, GR or RG as indicated in Section 9.12 on page 319.
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m
	≈ 3.3 W when using Power over Ethernet ≈ 2.9 W @ 12 VDC when supplied via the camera's 6-pin connector
I/O Ports	1 opto-isolated input line, 1 opto-isolated output line. 1 GPIO (can be set to operate as an input or an output). Power supplies must meet the SELV and LPS requirements (see page 59).
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 10: acA1920-40gm/gc

Specification	acA1920-40m/gc
Weight	< 90 g
Conformity	CE, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).

Table 10: acA1920-40gm/gc

Specification	acA1920-50gm/gc
Resolution (H x V pixels) **	gm/gc: 1936 x 1216 (full resolution) 1920 x 1200 (default resolution)
Sensor Type	Sony IMX174LLJ-C Progressive scan CMOS Global Shutter
Optical Size	1/1.2"
Effective Sensor Diagonal	13.4 mm
Pixel Size	5.86 µm x 5.86 µm
Max. Frame Rate (at full resolution)	51 fps
Mono/Color	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
	Color Models: Bayer RG 8 Bayer RG 12 Bayer RG 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed * If you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change Bayer BG, GR or RG as indicated in Section 9.12 on page 319.
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m
	≈ 3.4 W when using Power over Ethernet ≈ 3.0 W @ 12 VDC when supplied via the camera's 6-pin connector
I/O Ports	1 opto-isolated input line, 1 opto-isolated output line. 1 GPIO (can be set to operate as an input or an output). Power supplies must meet the SELV and LPS requirements (see page 59).
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)
Weight	< 90 g

Table 11: General Specifications (acA1920-50gm/gc)

Specification	acA1920-50gm/gc
Conformity	CE, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).

Table 11: General Specifications (acA1920-50gm/gc)

Specification	acA2000-50gm/gc	acA2000-50gmNIR	acA2040-25gm/gc
Resolution (H x V pixels)	gm: 2048 x 1088 gc: 2046 x 1086	gmNIR: 2048 x 1088	gm: 2048 x 2048 gc: 2046 x 2046
Sensor Type	CMOSIS CMV2000-2E5M / CMV2000-3E5C Progressive scan CMOS Global shutter	CMOSIS CMV2000-2E12M Progressive scan CMOS Global shutter	CMOSIS CMV4000-3E5M / CMV4000-2EM5C Progressive scan CMOS Global shutter
Optical Size	2/3"		1"
Effective Sensor Diagonal	12.75 mm		15.9 mm
Pixel Size	5.5 µm x 5.5 µm		
Max. Frame Rate (at full resolution)	50 fps		25 fps
Mono/Color	Mono or color (color models include a Bayer pattern RGB filter on the sensor)	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	-	Color Models: Mono 8 Bayer GR 8 Bayer GR 12 Bayer GR 12 Packed YUV 4:2:2 Packe YUV 4:2:2 (YUYV) Packed
Synchronization	Via hardware trigger, via software trigger, or free run		
Exposure Time Control	Via hardware trigger 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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m		
	≈ 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

Table 12: General Specifications (acA2000-50gm/gc, acA2000-50gmNIR, acA2040-25gm/gc)

Specification	acA2000-50gm/gc	acA2000-50gmNIR	acA2040-25gm/gc
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line. Power supplies must meet the SELV and LPS requirements (see page 59).		
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)		
Weight	< 90 g		
Conformity	CE, UL, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com		
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).		

Table 12: General Specifications (acA2000-50gm/gc, acA2000-50gmNIR, acA2040-25gm/gc)

Specification	acA2040-25gmNIR	acA2500-14gm/gc
Resolution (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"
Effective Sensor Diagonal	15.9 mm	7.13 mm
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 GB 8 Bayer GB 12 Bayer GB 12 Packed YUV 4:2:2 Packed YUV 4:2:2 (YUYV) Packed
Synchronization	Via hardware trigger, via software trigger, or free run	
Exposure Time Control	Via hardware trigger or programmable via the camera API	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; Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 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. Power supplies must meet the SELV and LPS requirements (see page 59).	
Lens Adapter	C-mount	C-mount; CS-mount

Table 13: General Specifications (acA2040-25gmNIR, acA2500-14gm/gc)

Specification	acA2040-25gmNIR	acA2500-14gm/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, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 13: General Specifications (acA2040-25gmNIR, acA2500-14gm/gc)

Specification	acA3800-10gm/gc	acA4600-7gc
Resolution (H x V pixels)	gm: 3856 x 2764 gc: 3856 x 2764	gc: 4608 x 3288
Sensor Type	Aptina MT9J003 Progressive scan CMOS Rolling shutter	Aptina MT9F002 Progressive scan CMOS Rolling shutter
Optical Size	1/2.3"	
Effective Sensor Diagonal	7.9 mm	7.9 mm
Pixel Size (H x V)	1.67 µm x 1.67µm	1.4 µm x 1.4 µm
Max. Frame Rate (at full resolution)	10 fps	7 fps
Mono/Color	Mono or color (color models include a Bayer pattern RGB filter on the sensor)	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
Synchronization	Via hardware trigger, via software trigger, or free run	
Exposure Time Control	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 Cable must be at least a 26 AWG cable. Max. cable length: 10 m	
	≈ 3.5 W when using Power over Ethernet ≈ 3.3 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. Power supplies must meet the SELV and LPS requirements (see page 59).	
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 14: General Specifications (acA3800-10gm/gc, acA4600-7gc)

Specification	acA3800-10gm/gc	acA4600-7gc
Weight	< 90 g	
Conformity	CE, UL, FCC, GenICam, GigE Vision, IP30, RoHS, IEEE 802.3af (PoE) The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon Camera Software Suite (version 4.0 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM).	

Table 14: General Specifications (acA3800-10gm/gc, acA4600-7gc)

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.

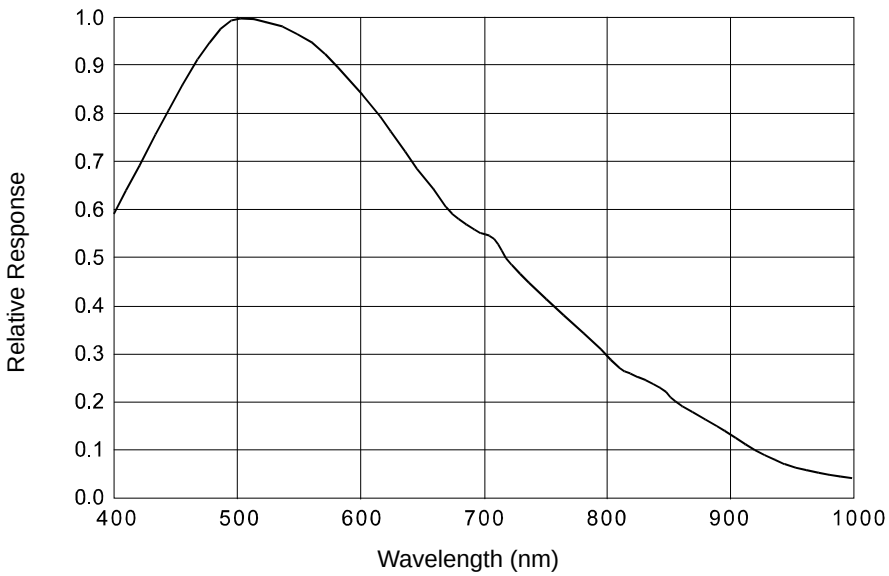


Fig. 1: acA640-90gm Spectral Response (From Sensor Data Sheet)

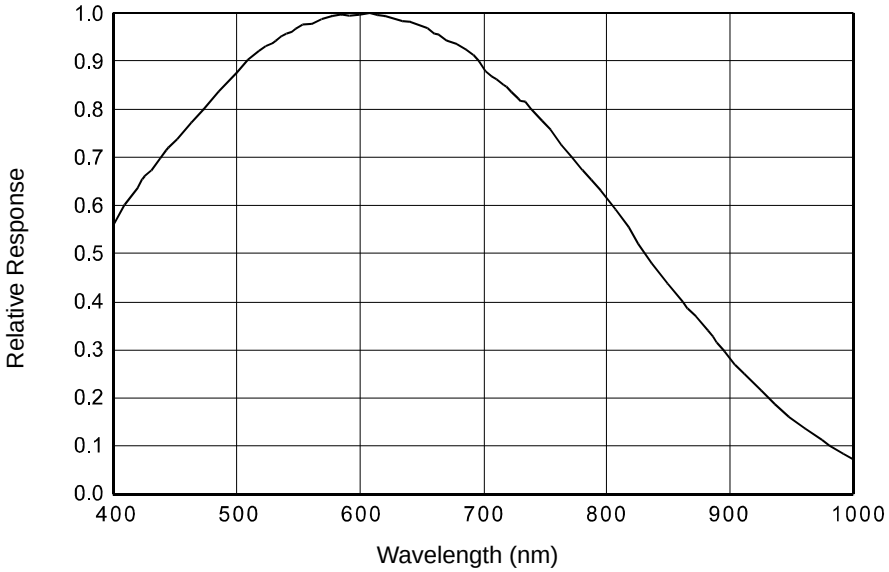


Fig. 2: acA640-120gm Spectral Response (From Sensor Data Sheet)

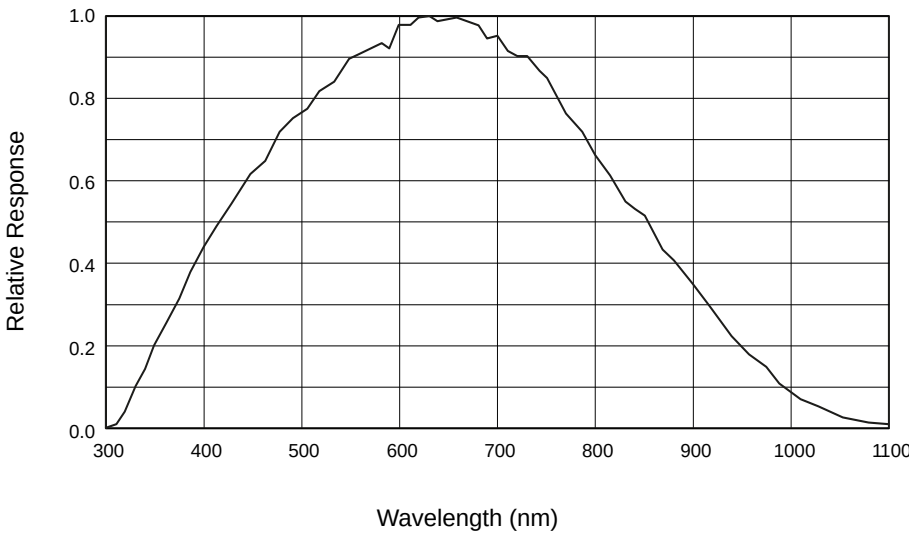


Fig. 3: acA640-300gm, acA800-200gm, acA1300-75gm Spectral Response (From Sensor Data Sheet)

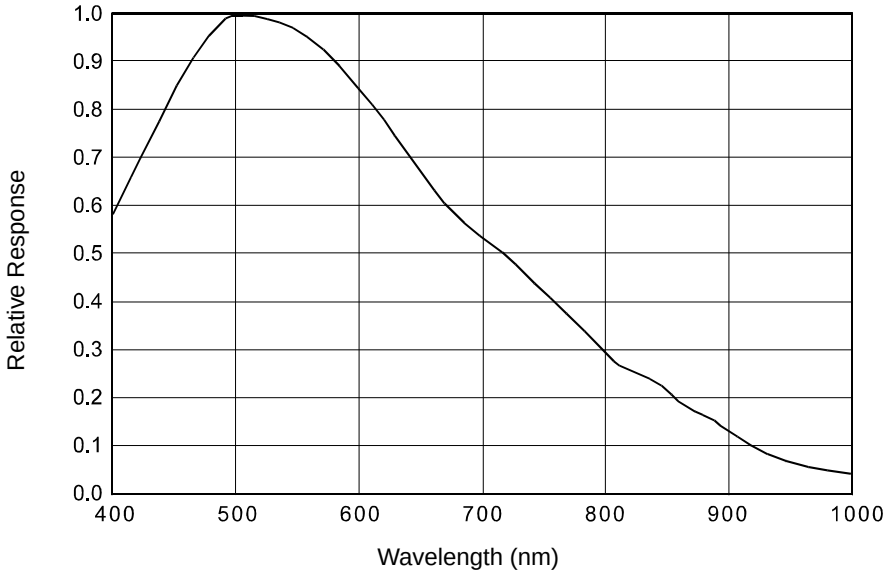


Fig. 4: acA645-100gm Spectral Response (From Sensor Data Sheet)

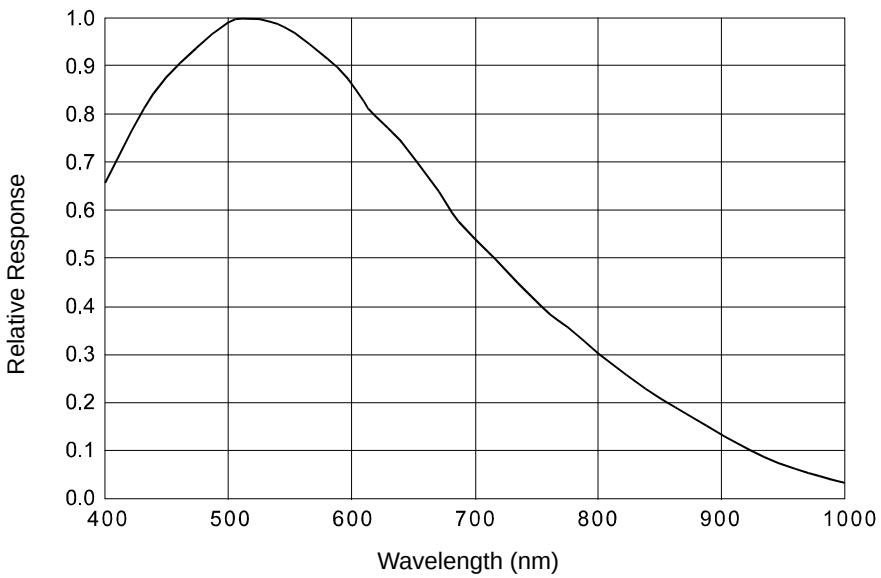


Fig. 5: acA750-30gm Spectral Response (From Sensor Data Sheet)

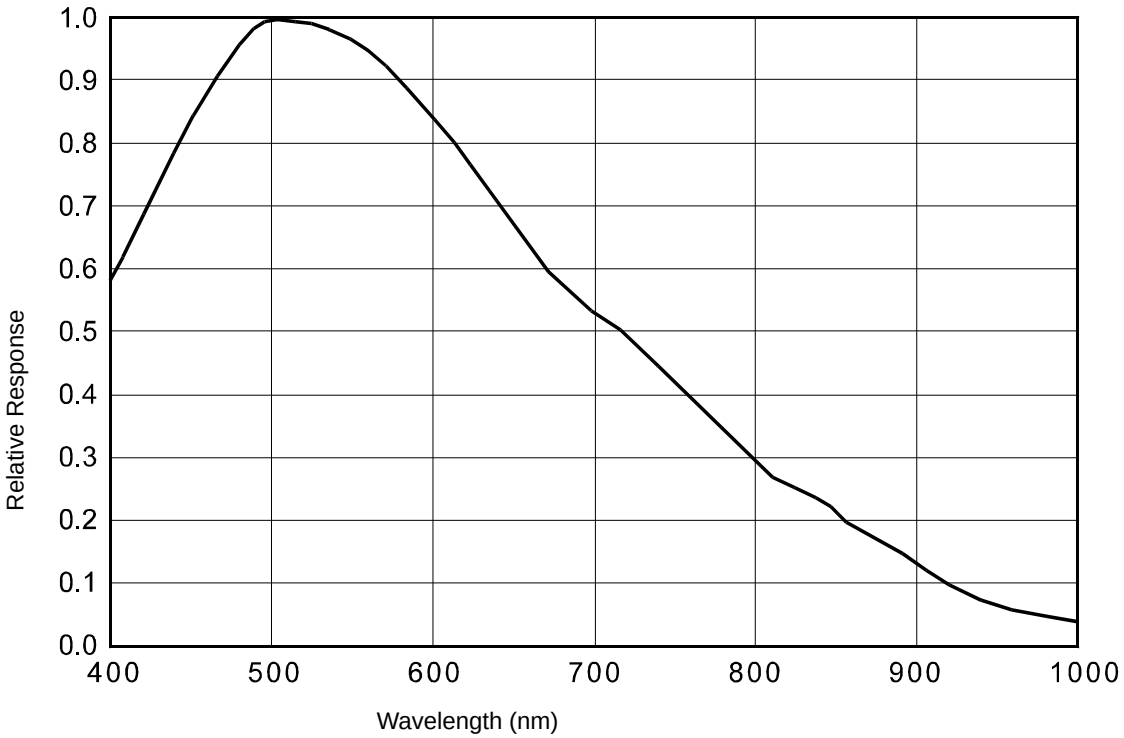


Fig. 6: acA780-75gm Spectral Response (From Sensor Data Sheet)

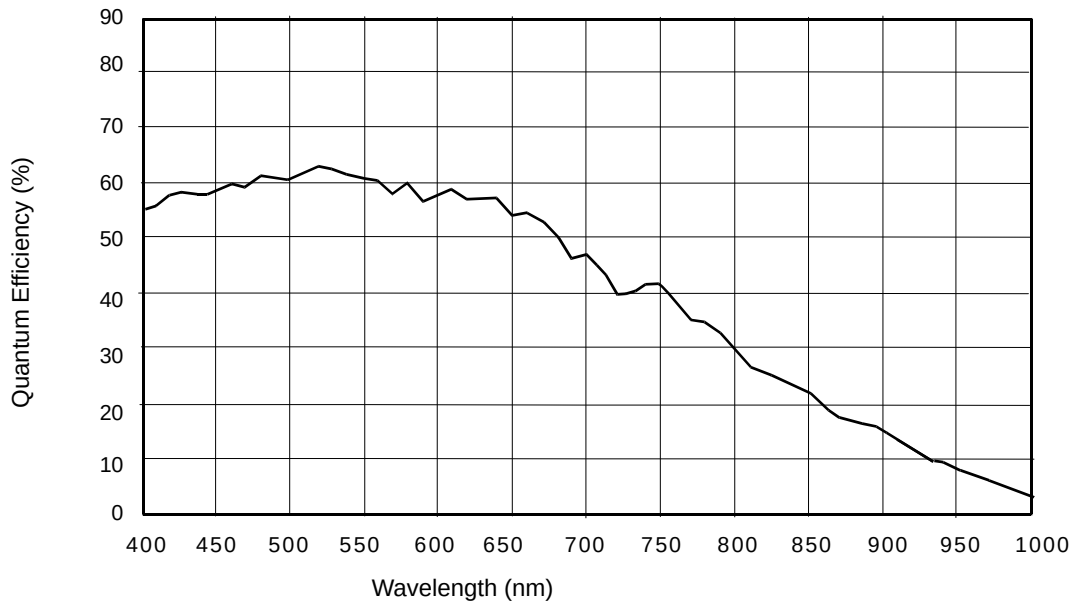


Fig. 7: acA1280-60gm, acA1300-60gm Spectral Response (From Sensor Data Sheet)

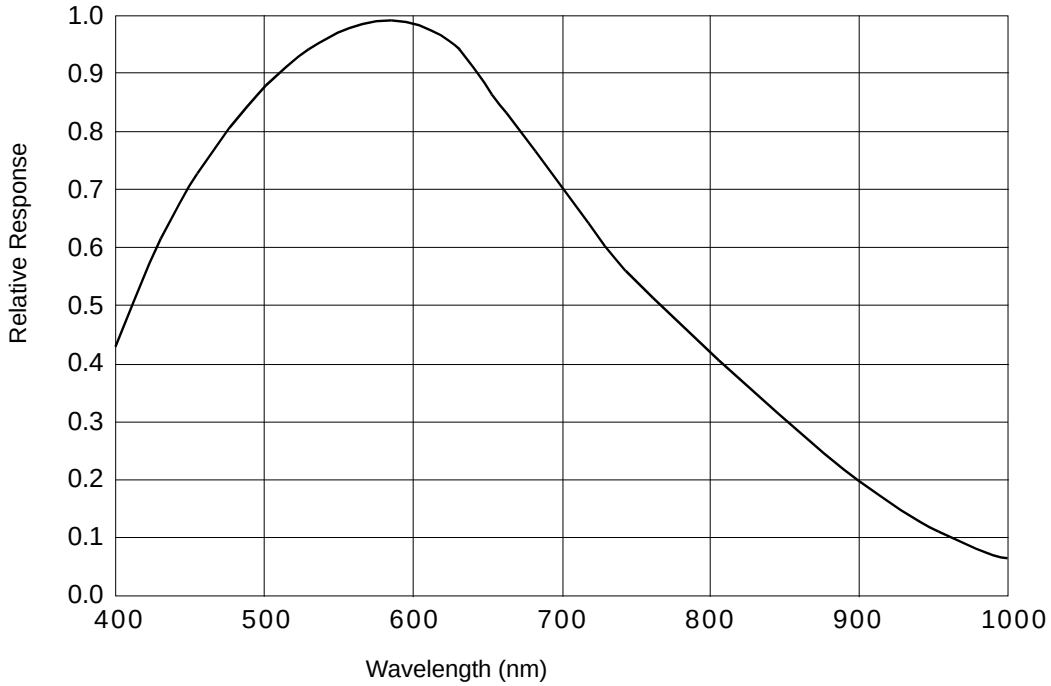


Fig. 8: acA1300-22gm, acA1300-30gm Spectral Response (From Sensor Data Sheet)

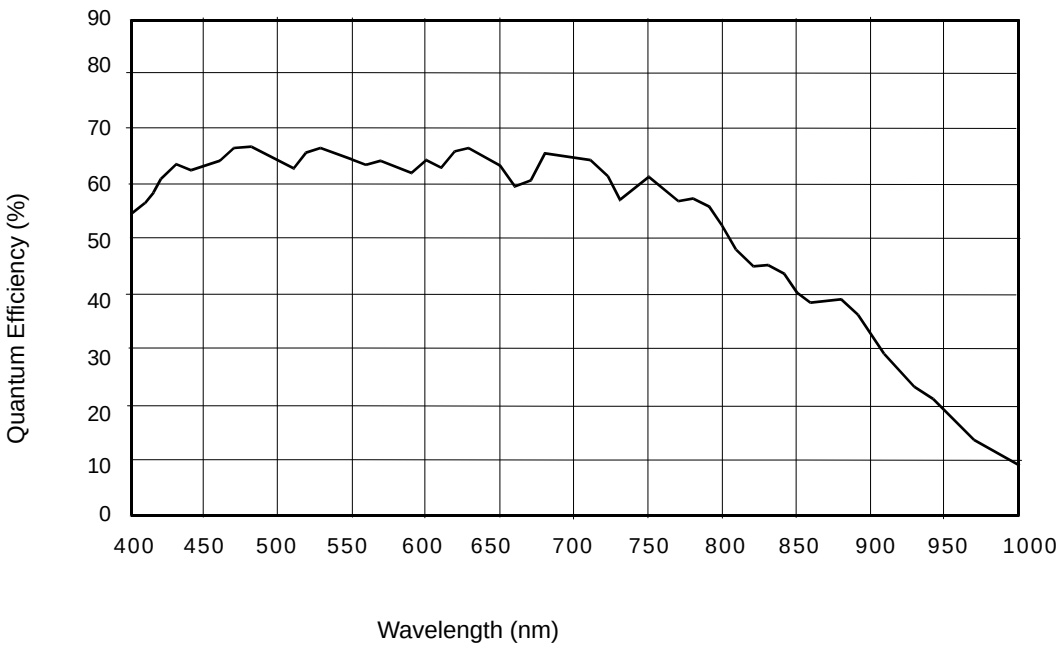


Fig. 9: acA1300-60gmNIR Spectral Response (From Sensor Data Sheet)

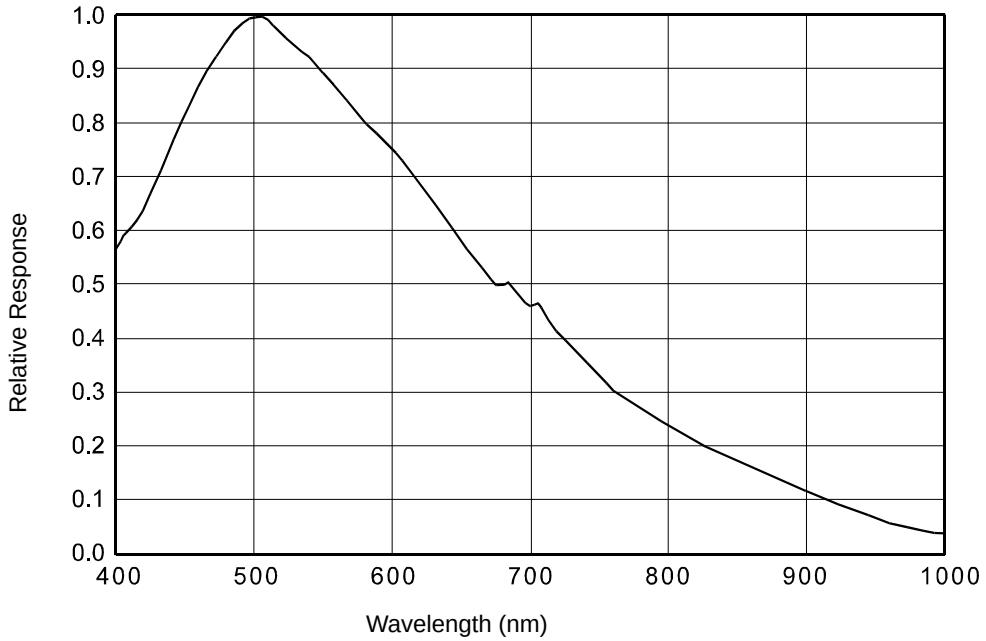


Fig. 10: acA1600-20gm Spectral Response (From Sensor Data Sheet)

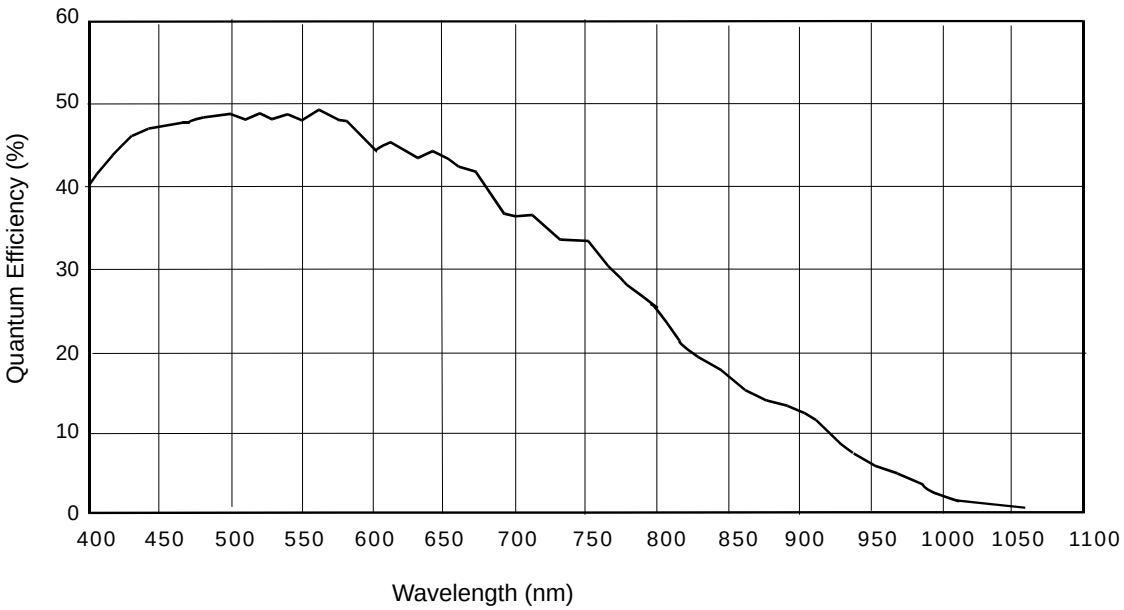


Fig. 11: acA1600-60gm Spectral Response (From Sensor Data Sheet)

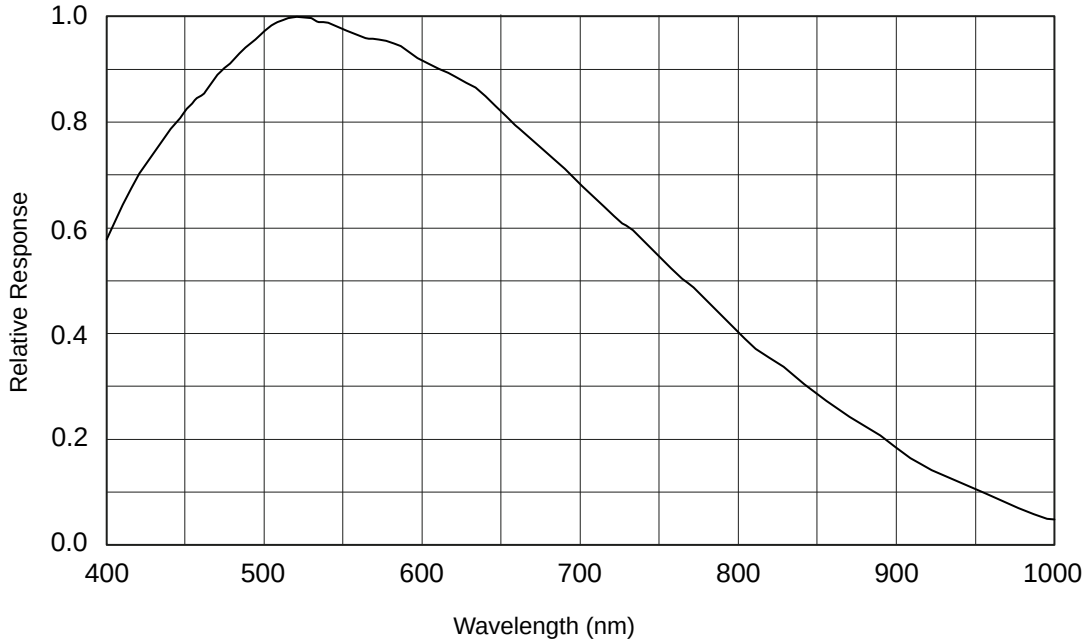


Fig. 12: acA1920-40gm, acA1920-50gm Spectral Response (From Sensor Data Sheet)

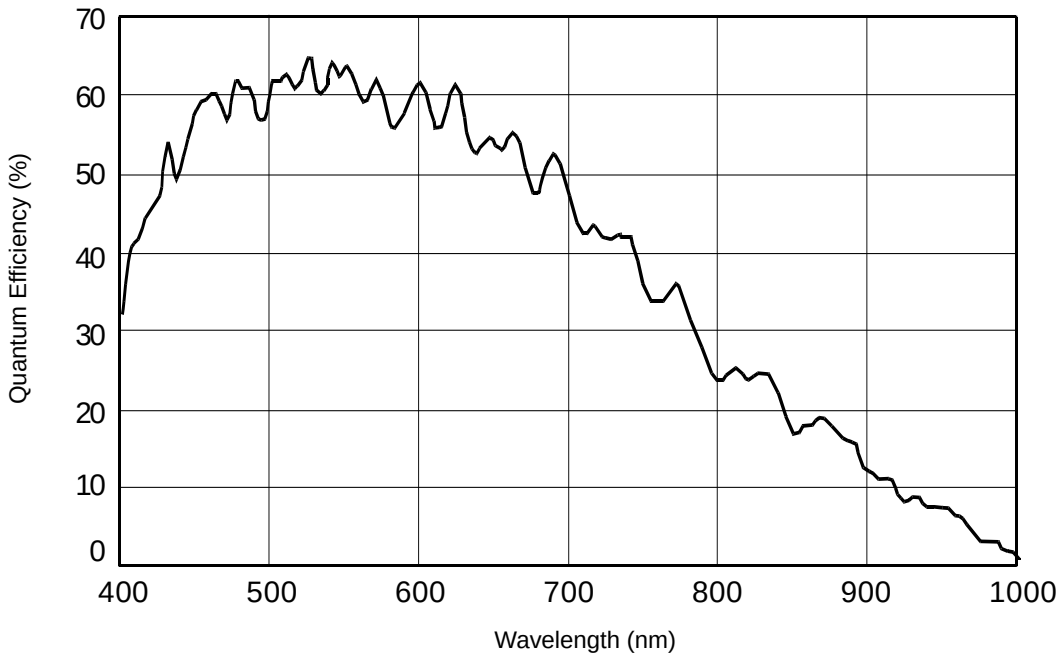


Fig. 13: acA2000-50gm, acA2040-25gm Spectral Response (From Sensor Data Sheet)

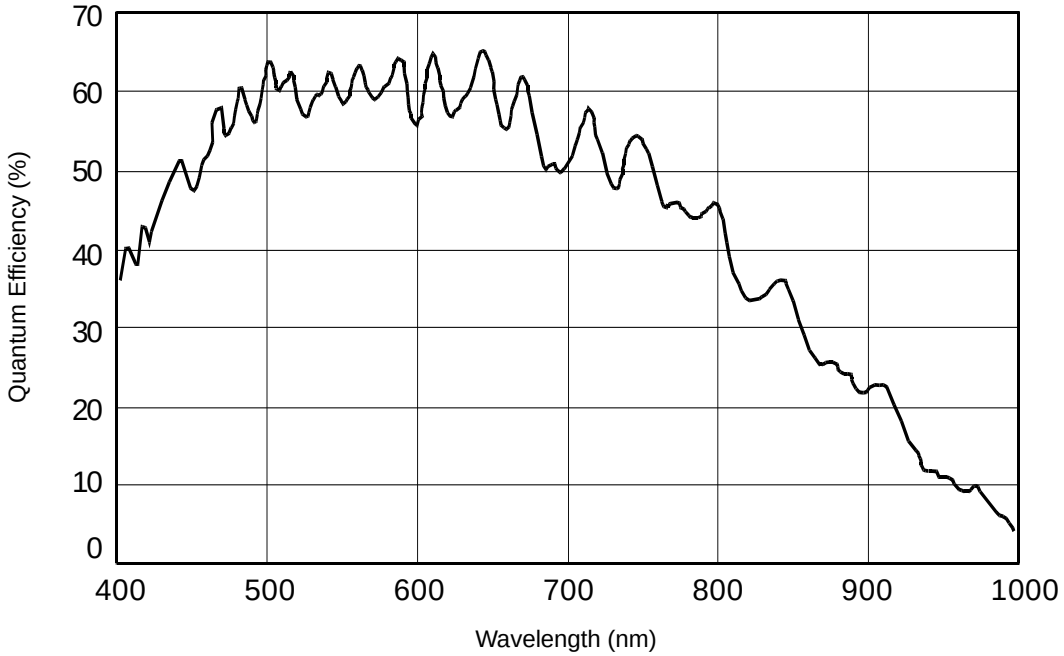


Fig. 14: acA2000-50gmNIR, acA2040-25gmNIR Spectral Response (From Sensor Data Sheet)

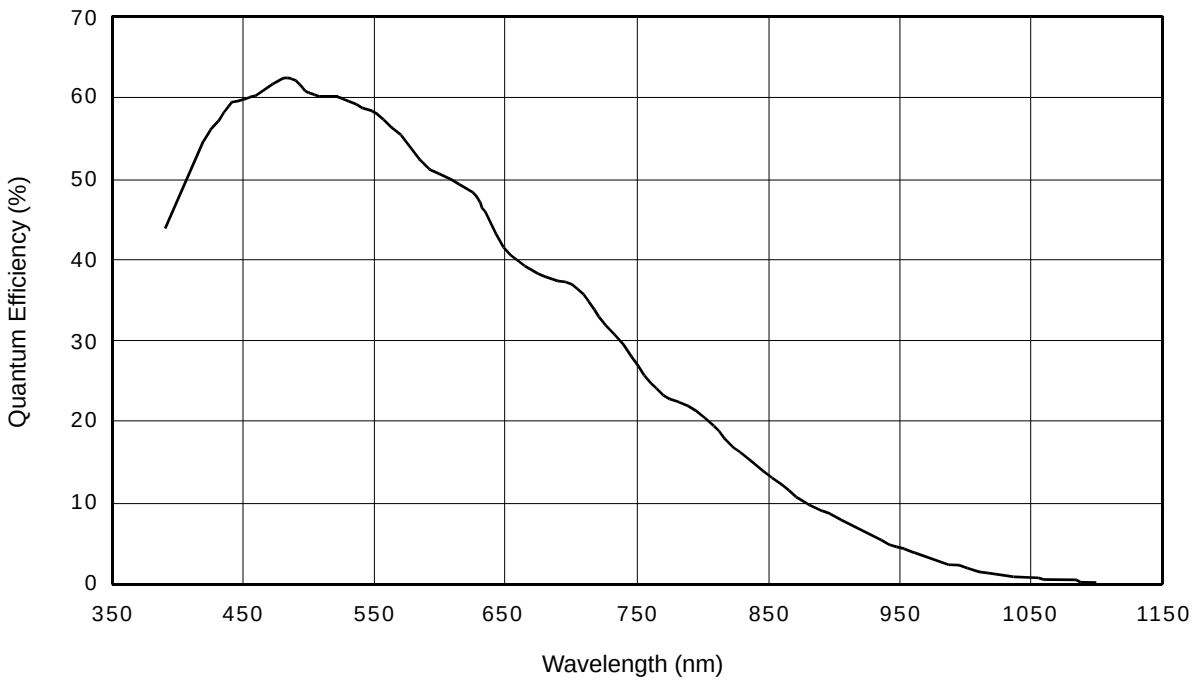


Fig. 15: acA2500-14gm, acA1920-25gm Spectral Response (From Sensor Data Sheet)

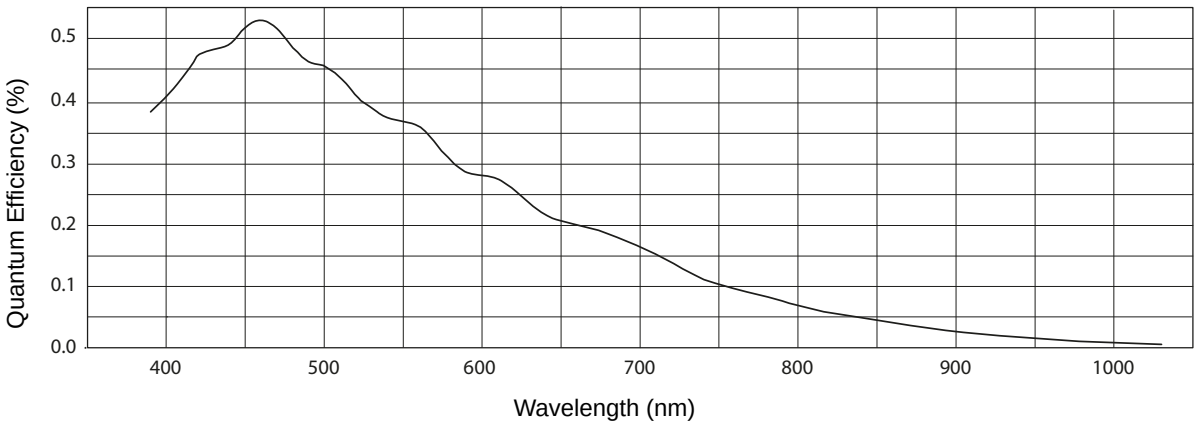


Fig. 16: acA3800-10gm Spectral Response (From Sensor Data Sheet)

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.

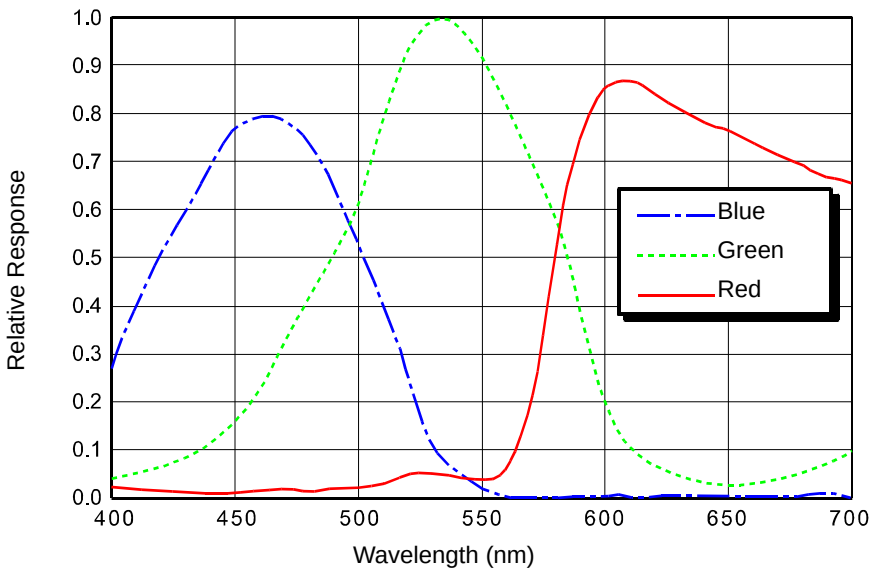


Fig. 17: acA640-90gc Spectral Response (From Sensor Data Sheet)

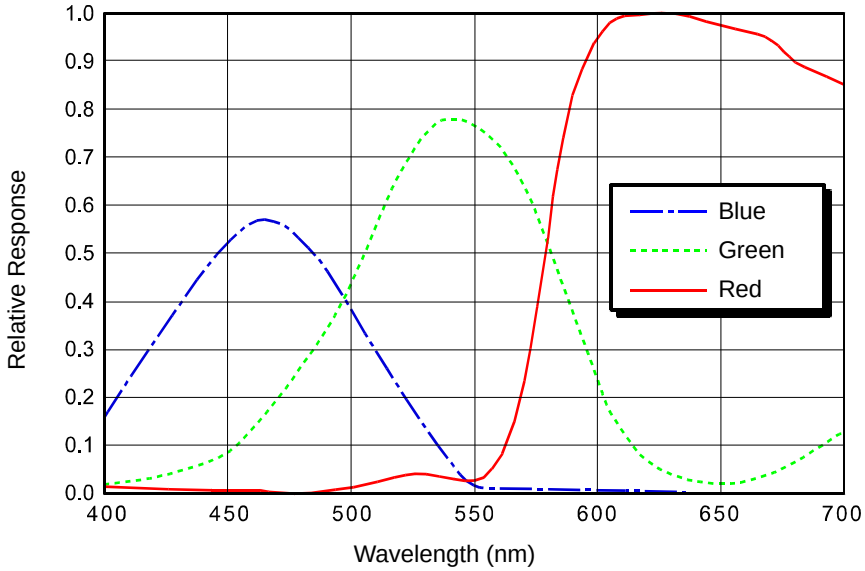


Fig. 18: acA640-120gc Spectral Response (From Sensor Data Sheet)

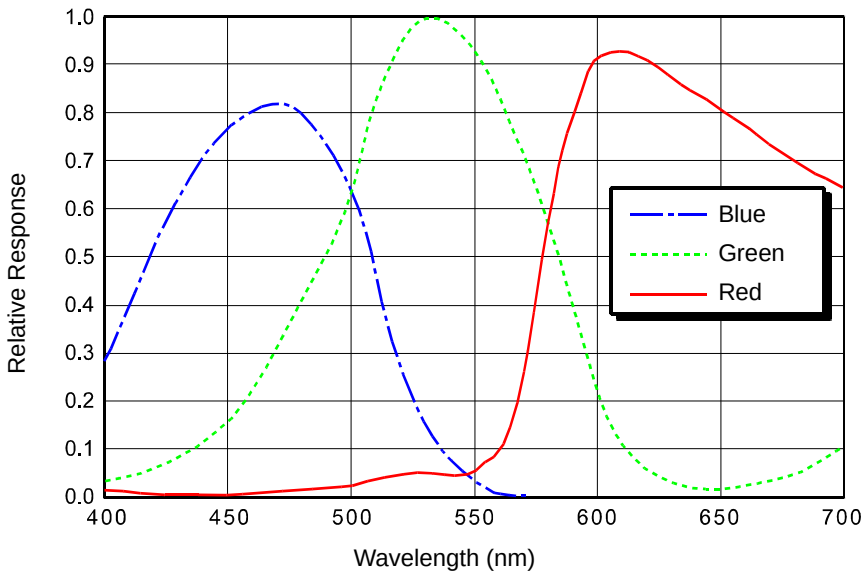


Fig. 19: acA645-100gc Spectral Response (From Sensor Data Sheet)

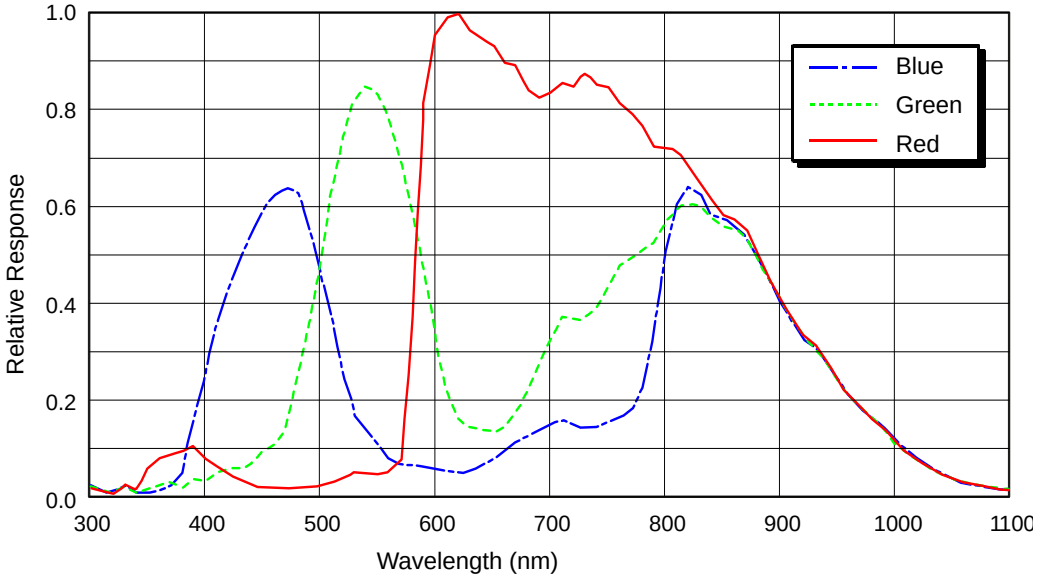


Fig. 20: acA640-300gc, acA800-200gc, acA1300-75gc, Spectral Response (From Sensor Data Sheet)

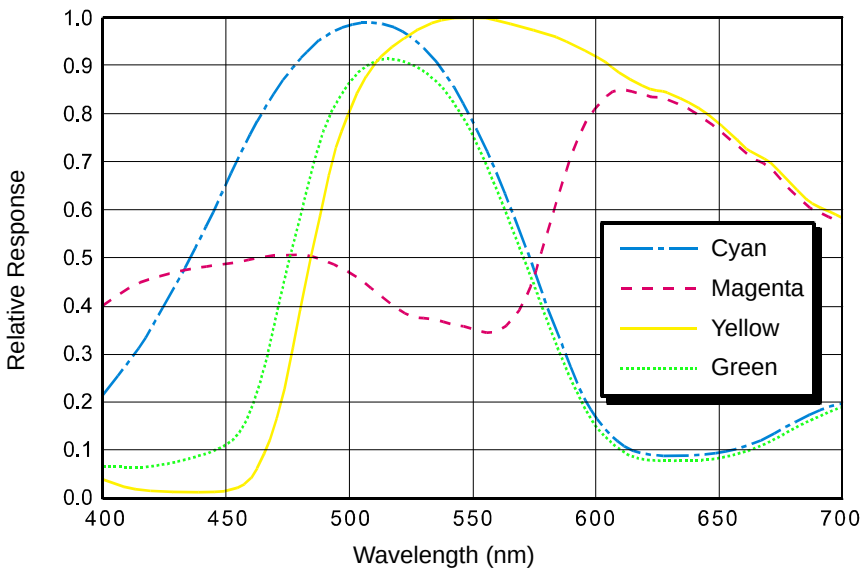


Fig. 21: acA750-30gc Spectral Response (From Sensor Data Sheet)

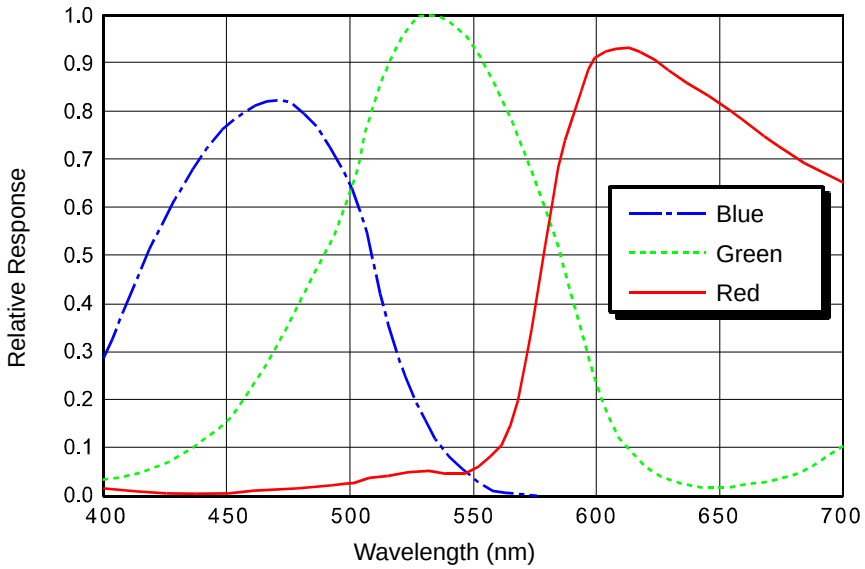


Fig. 22: acA780-75gc Spectral Response (From Sensor Data Sheet)

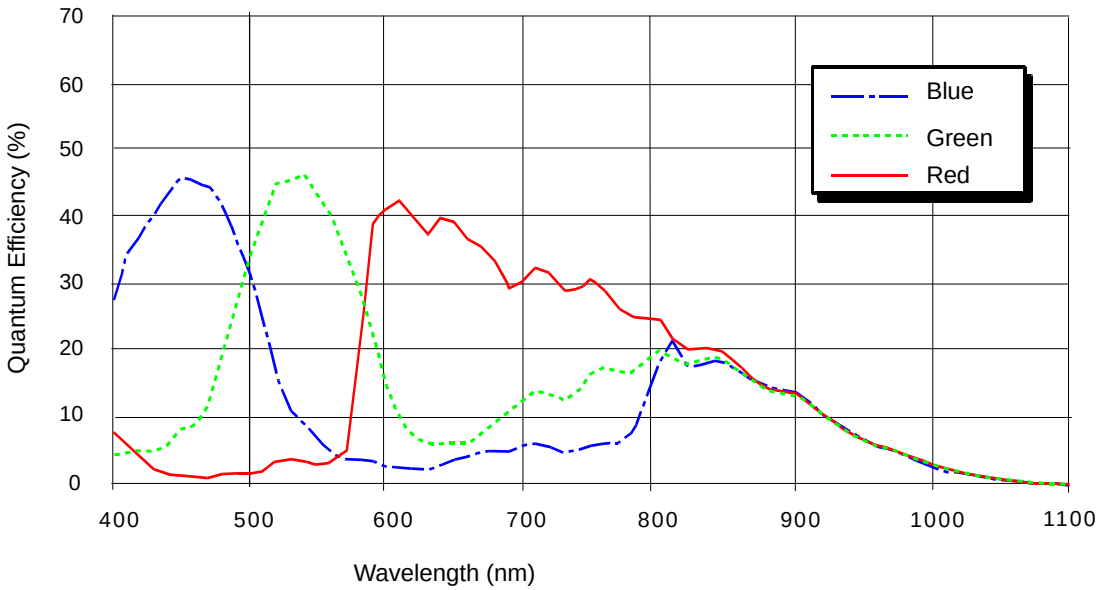


Fig. 23: acA1280-60gc, acA1300-60gc Spectral Response (From Sensor Data Sheet)

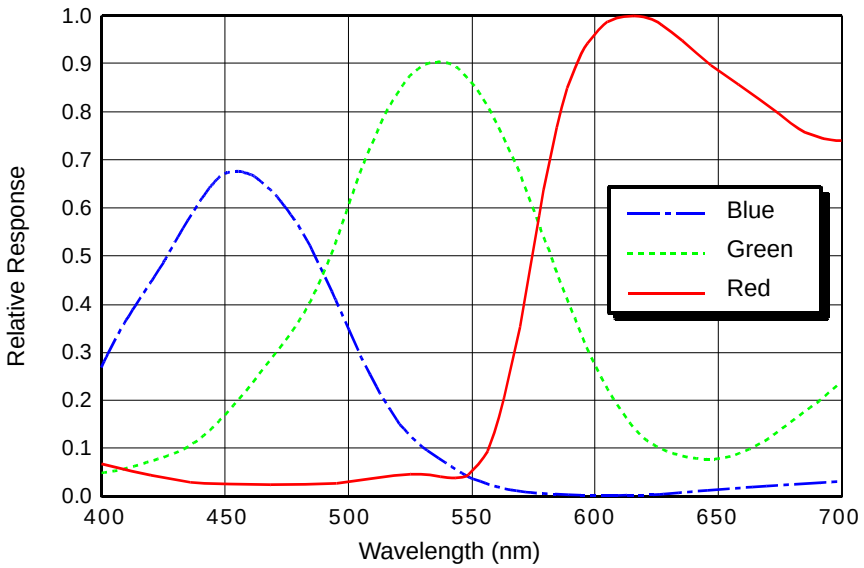


Fig. 24: acA1300-22gc, acA1300-30gc Spectral Response (From Sensor Data Sheet)

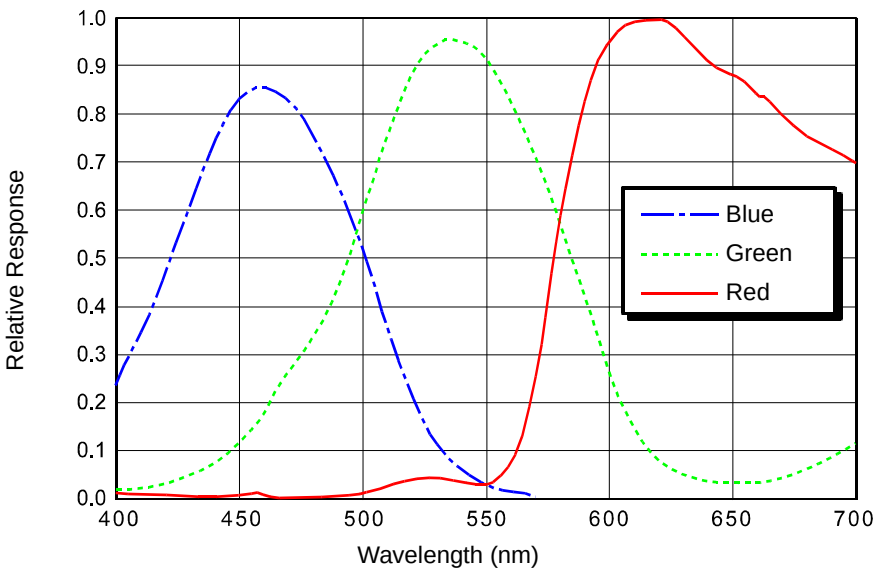


Fig. 25: acA1600-20gc Spectral Response (From Sensor Data Sheet)

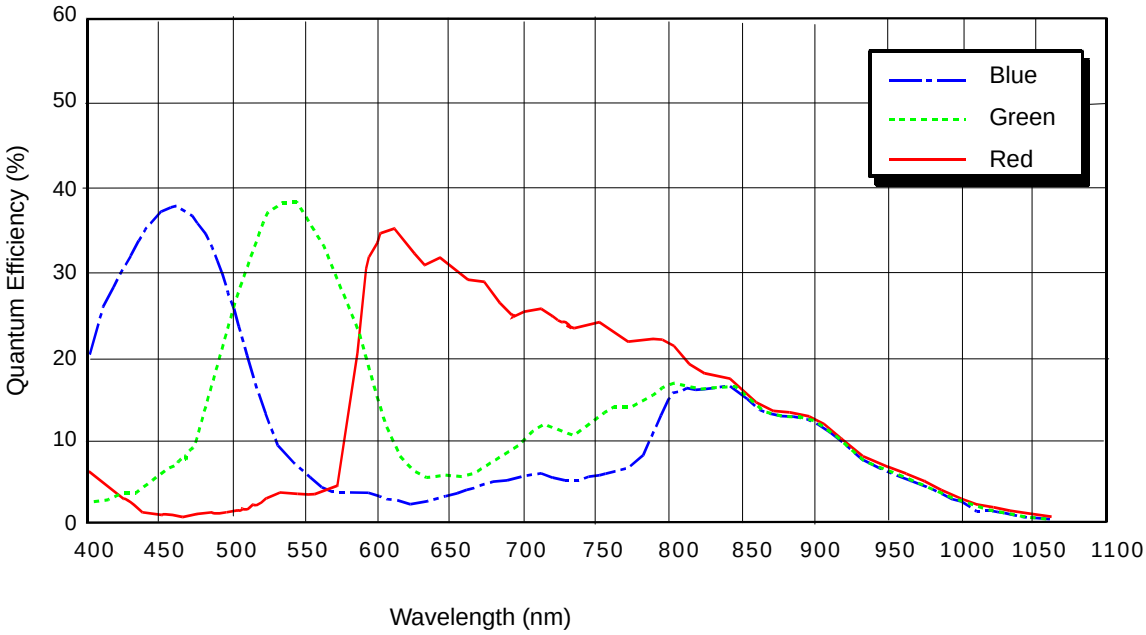


Fig. 26: acA1600-60gc Spectral Response (From Sensor Data Sheet)

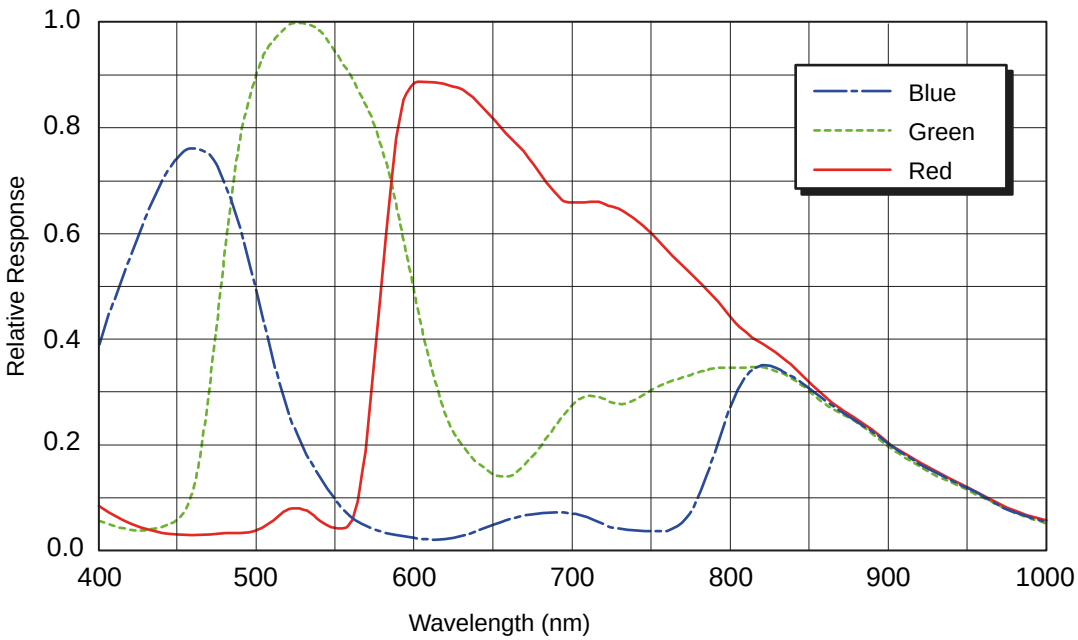


Fig. 27: acA1920-40gc, acA1920-50gc Spectral Response (From Sensor Data Sheet)

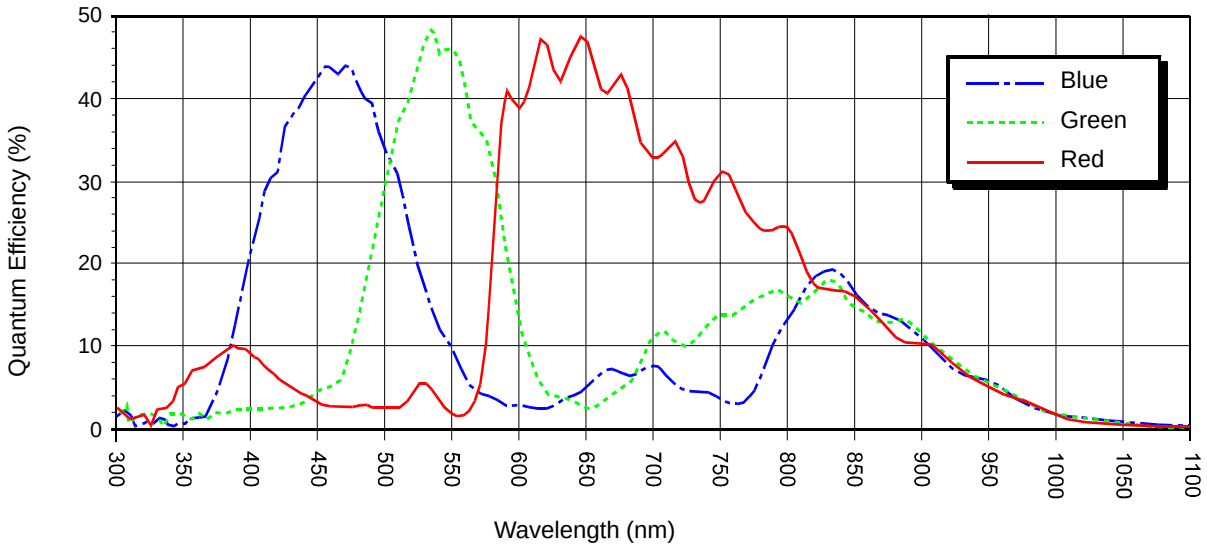


Fig. 28: acA2000-50gc, acA2040-25gc Spectral Response (From Sensor Data Sheet)

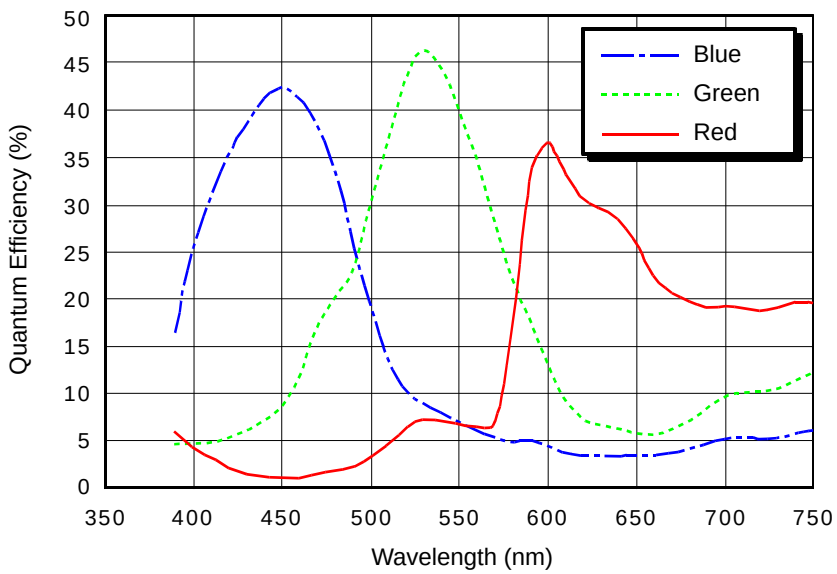


Fig. 29: acA1920-25gc, acA2500-14gc, Spectral Response (From Sensor Data Sheet)

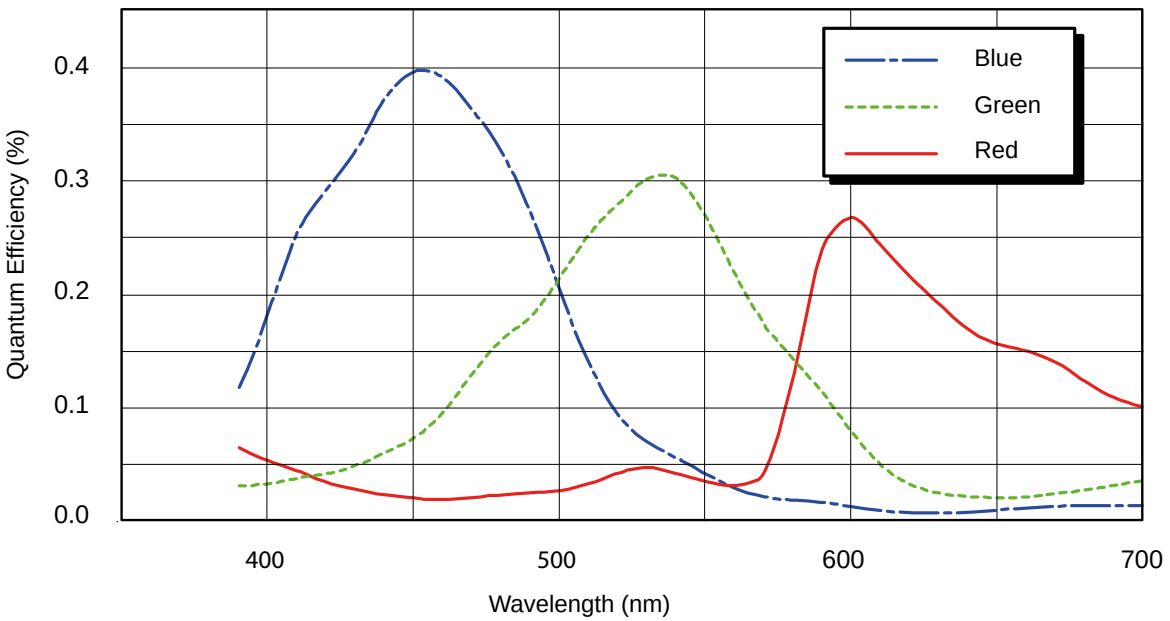


Fig. 30: acA3800-10gc Spectral Response (From Sensor Data Sheet)

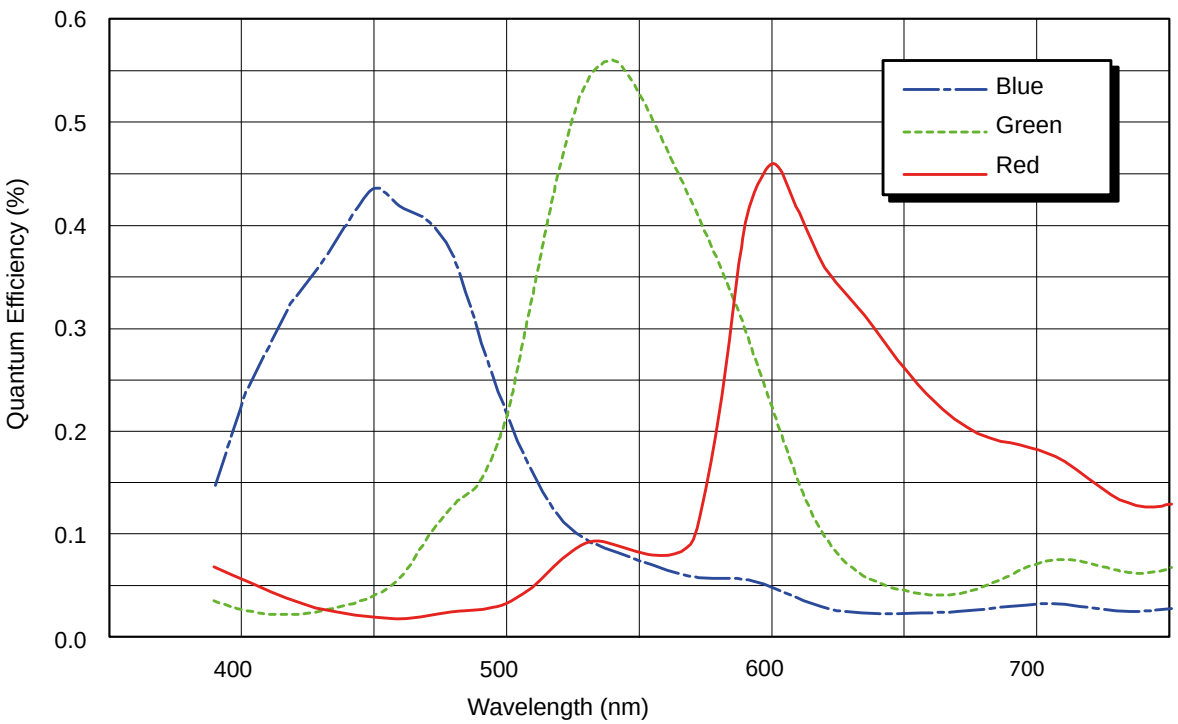


Fig. 31: acA4600-7gc Spectral Response (From Sensor Data Sheet)

1.4 Mechanical Specifications

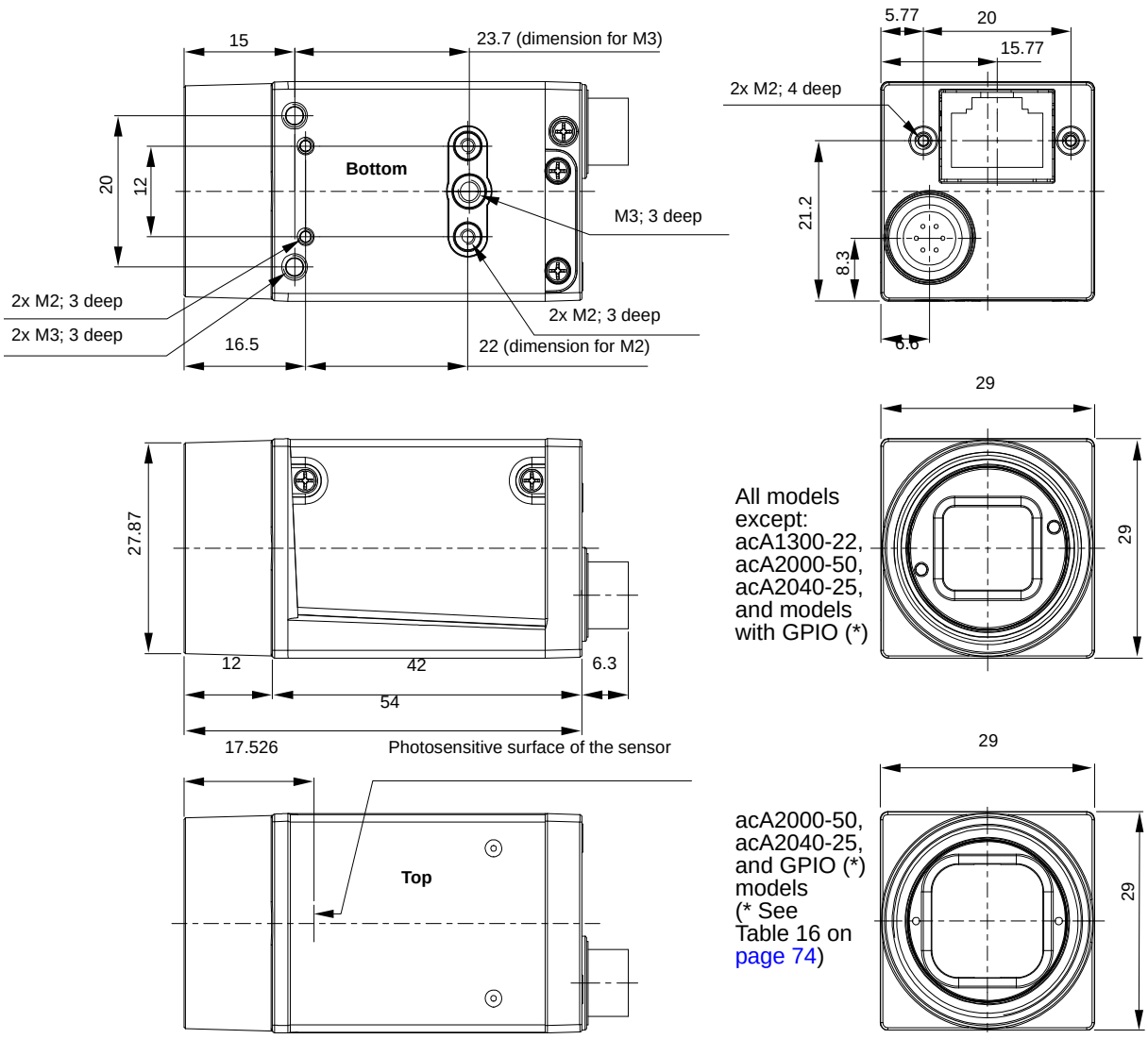
The camera housing conforms to protection class IP30 assuming that the lens mount is covered by a lens or by the protective plastic cap that is shipped with the camera.

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 32.
- with a CS-mount lens adapter are as shown in Figure 33.

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



Not to scale

Fig. 32: Mechanical Dimensions (in mm) for Cameras with the C-mount Lens Adapter

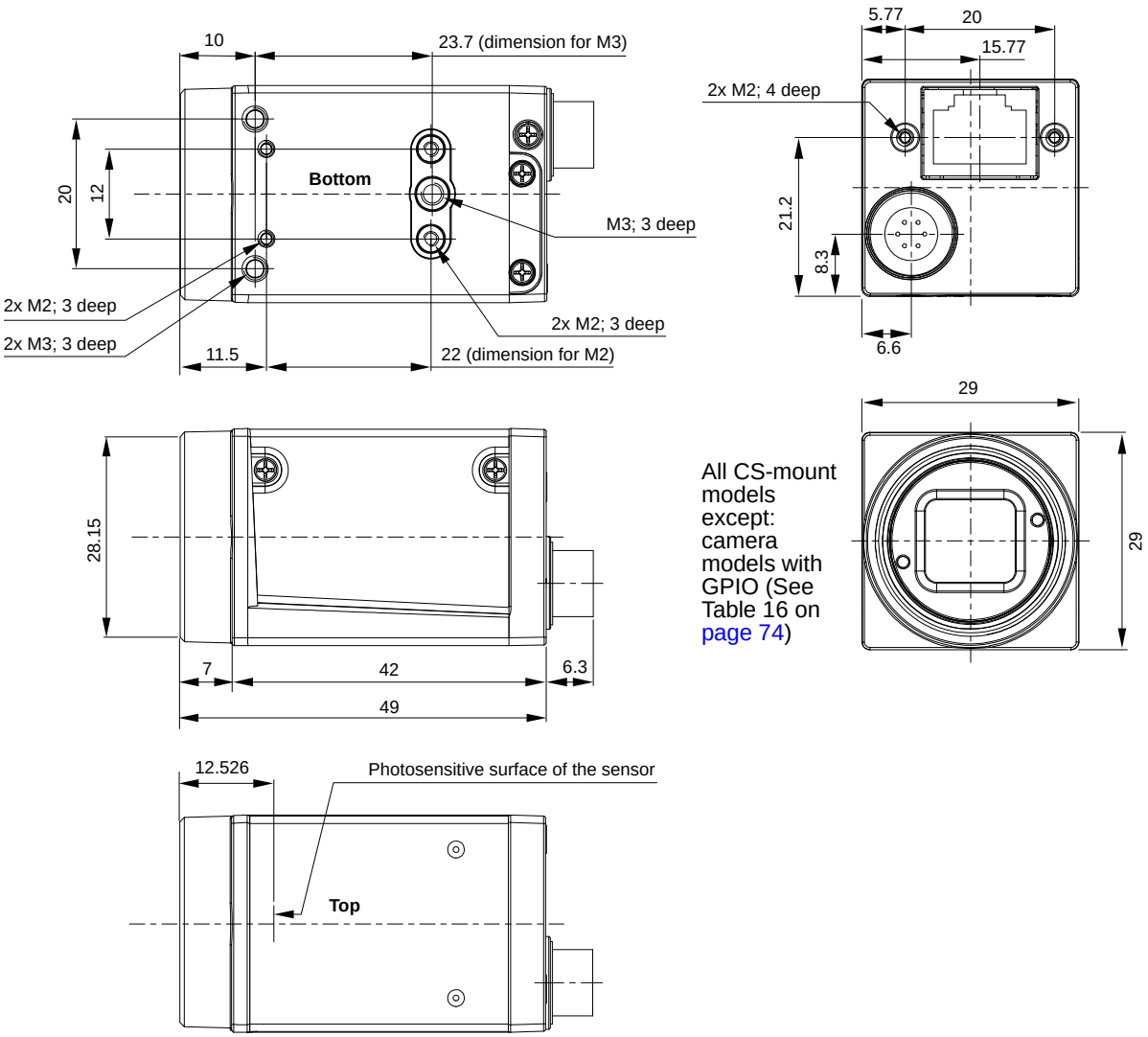


Fig. 33: Mechanical Dimensions (in mm) for Cameras with the CS-mount Lens Adapter

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 34):

  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 35):

  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.

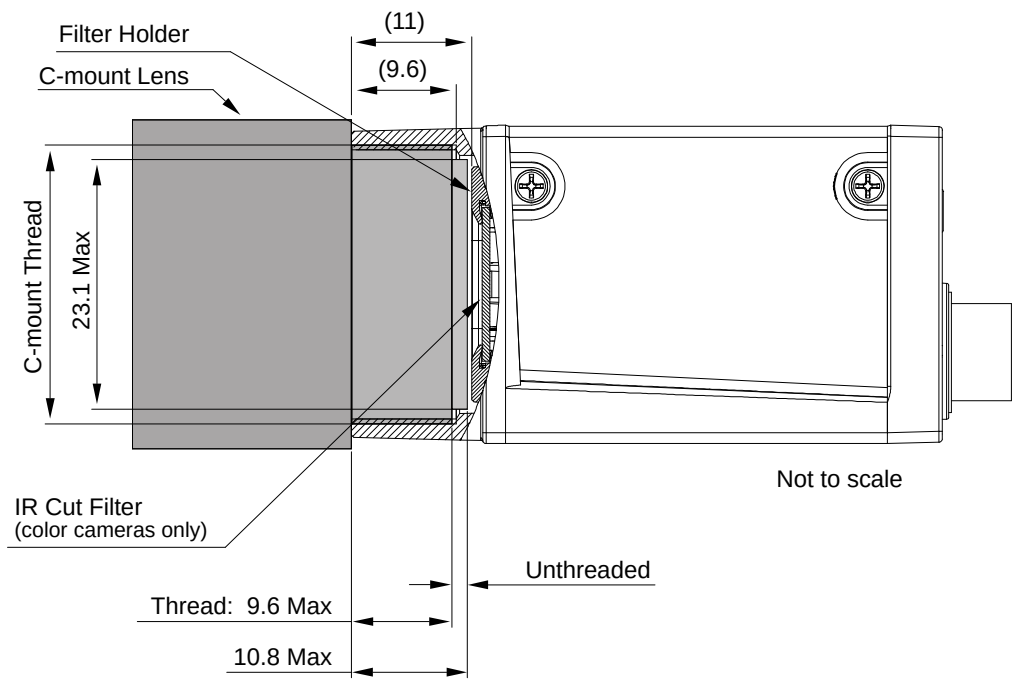


Fig. 34: Maximum Lens Thread Length (Dimensions in mm) for Cameras with the C-mount Lens Adapter

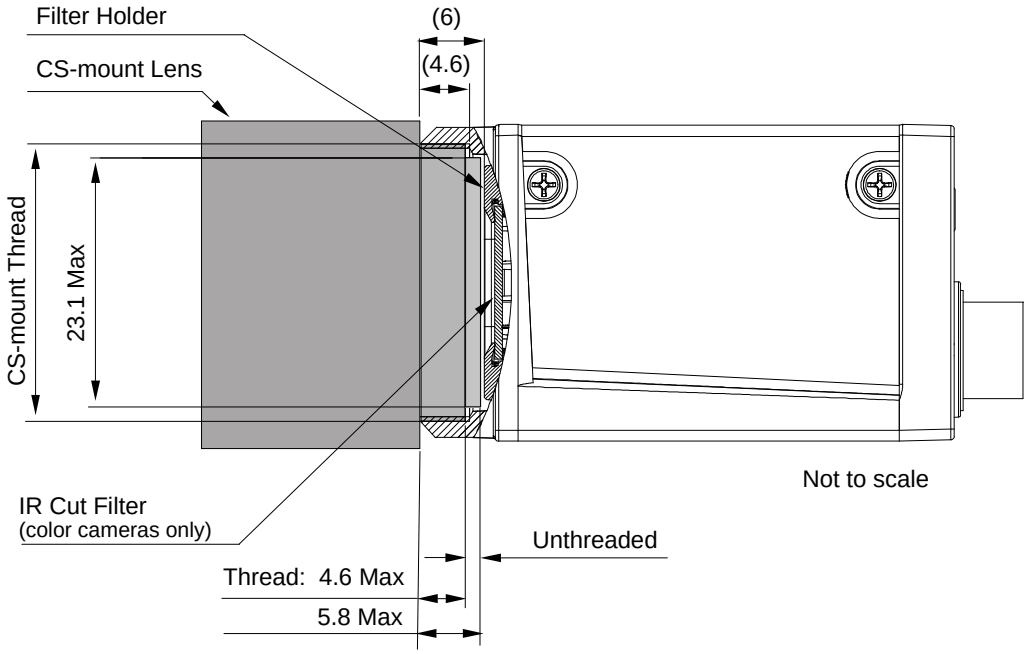


Fig. 35: Maximum Lens Thread Length (Dimensions in mm) for Cameras with the CS-mount Lens Adapter

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 15: 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

1.5.1 LWIP TCP/IP Licensing

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

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

1.5.2 LZ4 Licensing

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

LZ4 - Fast LZ compression algorithm

Copyright (C) 2011-2013, Yann Collet.

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

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

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

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

1.6 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 the Basler 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:

- 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.
- 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.

To ensure good image quality, we recommend not to operate the camera at elevated temperatures.

1.7.3 Temperature Monitoring and Over Temperature Detection

1.7.3.1 Coreboard Temperature

Available for
acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

The cameras from the table above are equipped with a temperature sensor mounted on the coreboard. The temperature sensor lets you read the current temperature of the camera's coreboard in degrees C.

Reading the Coreboard Temperature

You can use the pylon API to read the coreboard temperature in degrees C from within your application software.

To read the coreboard temperature:

1. Select the coreboard board temperature sensor.
2. Read the temperature.

The following code snippet illustrates using the API to read the temperature in degrees C:

```
// Select the coreboard temperature sensor
Camera.TemperatureSelector.SetValue (TemperatureSelector_Coreboard);
// Read the coreboard temperature
double d = Camera.TemperatureAbs.GetValue();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3 on page 63.

1.7.3.2 Coreboard Temperature Conditions and Over Temperature Idle Mode

The temperature sensor is used to monitor the temperature of the camera's coreboard. The camera also has coreboard over temperature protection. An over temperature condition is detected, if the temperature of the coreboard rises above certain temperatures. Two stages can be distinguished:

- the critical temperature of 77 °C (+170.6 °F) and
- the over temperature of 80 °C (+172.4 °F).

When the camera heats up and reaches the internal critical temperature of 77 °C (+170.6 °F) a critical temperature event message is transmitted. It warns that the camera is about to reach the internal over temperature condition.

When the camera's internal temperature increases beyond the over temperature of 80 °C (+176 °F), an over temperature condition is reached and the camera enters the over temperature idle mode (availability, see table below). In this mode, the camera will no longer acquire images but display test image 2. In addition, an over temperature event will be transmitted.

To obtain the event messages, event notification must have been enabled (see Section 9.16 on page 350). For more information about event notification, see Section 9.16 on page 350.

For acA1920-40 and acA1920-50, the following functionalities are not available yet:

- No event notification for these camera models, when the critical or over temperature is reached.
- No temperature idle mode
- The following functionalities not available yet:
  - `camera.CriticalTemperature.GetValue();`
  - `bool b = camera.OverTemperature.GetValue()`
  - `TemperatureStateEnums e = camera.TemperatureState.GetValue();`

NOTICE

When the camera is in the over temperature mode, irreversible damage to the camera can occur.

- Provide sufficient heat dissipation (see Section 1.7.2 on page 55) to quickly decrease the camera's internal temperature and exit the over temperature idle mode.
- Provide sufficient heat dissipation to ensure that the camera will ideally never reach the internal over temperature condition.

When the camera's internal temperature is decreased below 77 °C (+170.6 °F), the camera will automatically resume operation. The camera will operate as it did, before entering the over temperature idle mode. A critical temperature event message will not be transmitted as the internal temperature is decreased.

- Note that regular operation of the camera requires both, that
  - the camera's internal temperature is below the internal critical temperature **and** that
  - the housing temperature stays within the specified range of "housing temperature during operation" (see Section 1.7.1 on page 55).
- Note that elevated temperatures will worsen image quality and shorten the camera's lifetime. The latter effect will also become more severe as the number of high-temperature conditions increases.

You can use the pylon API to obtain the critical temperature and over temperature values in degrees C from within your application software.

```
bool b = camera.CriticalTemperature.GetValue();
bool b = camera.OverTemperature.GetValue();
```

If you want to get informed about the temperature status of the camera's coreboard, you can also use the pylon API:

```
TemperatureStateEnums e = camera.TemperatureState.GetValue();
```

The following temperature states can be read:

Possible values from temperature readout	Meaning: The internal temperature on the coreboard ...
Ok	... is below the critical temperature. Normal operation condition
Critical	... has reached the critical temperature of 77 °C. See notice on previous page.
Error	... has reached the over temperature 80 °C. See notice on previous page.

1.7.3.3 Getting Information about the Temperature Status via Events

Event Notification ailable for ...
acA640-300, acA800-200, acA1300-75

You can set up event notification to get informed about the different temperature conditions in the camera. Two "temperature" events can be generated:

- CriticalTemperatureEvent
  When the camera models indicated above reach a temperature of 77 °C (+170.6 °F), the so-called critical temperature, the camera can issue the CriticalTemperature event.
- OverTemperatureEvent
  When the camera models indicated above reach a temperature of 80 °C (+176 °F), the camera can issue the OverTemperature event.

In both cases you should sufficiently decrease the camera's temperature to prevent the camera from overheating. For information about heat dissipation, see Section 1.7.2 on page 55.

You can enable the "temperature" events in the software. For information about event reporting, see Section 9.16 on page 350.

1.8 Precautions

DANGER

Electric Shock Hazard

Unapproved power supplies may cause electric shock. Serious injury or death may occur.

- You must use camera power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the SELV and LPS requirements.

WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

- You must use camera power supplies which meet the Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the LPS requirements.

NOTICE

On all cameras, the lens thread length is limited.

All cameras (mono, color, and mono NIR) 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 49.

NOTICE

Voltage outside of the specified range can cause damage.

- If you are supplying camera power via Power over Ethernet (PoE), the power must comply with the IEEE 802.3af specification.
- 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 +10.8 VDC, the camera may operate erratically.
- The ace GigE cameras must only be connected to other limited power sources (LPS) / Safety Extra Low Voltage (SELV) circuits that do not represent any energy hazards.

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.

- 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.
- 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.

NOTICE

Avoid dust on the sensor.

The camera is shipped with a plastic cap on the lens mount. To avoid collecting dust on the camera's IR cut filter (color cameras) or sensor (mono and mono NIR cameras), make sure that you always put the plastic cap in place when there is no lens mounted on the camera.

To avoid collecting dust on the camera's IR cut filter (color cameras) or sensor (mono cameras), make sure to observe the following:

- Always put the plastic cap in place when there is no lens mounted on the camera.
- Make sure that the camera is pointing down every time you remove or replace the plastic cap, a lens or a lens adapter.
- Never apply compressed air to the camera. This can easily contaminate optical components, particularly the sensor.

NOTICE

Cleaning of the sensor and the housing

Sensor

Avoid cleaning the surface of the camera's sensor if possible. If you must clean it:

- Before starting, disconnect the camera from camera power and I/O power.
- Use a soft, lint-free cloth dampened with a small amount of high-quality window cleaner.
- Because electrostatic discharge can damage the sensor, you must use a cloth that won't generate static during cleaning (cotton is a good choice).
- Make sure the window cleaner has evaporated after cleaning, before reconnecting the camera to the power.

Housing

To clean the surface of the camera housing:

- Do not use solvents or thinners; they can damage the surface.
- Use a soft, dry cloth that won't generate static during cleaning (cotton is a good choice).
- To remove tough stains, use a soft cloth dampened with a small amount of neutral detergent; then wipe dry.
- Make sure the detergent has evaporated after cleaning, before reconnecting the camera to the power.

2 Installation

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

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

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

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.

DANGER

Electric Shock Hazard

Unapproved power supplies may cause electric shock. Serious injury or death may occur.

- You must use camera power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the SELV and LPS requirements.

WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

- You must use camera power supplies which meet the Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the LPS requirements.

3 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 Camera Software Suite 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 API.

3.1 Basler pylon Camera Software Suite

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

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

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

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

3.1.1 pylon Viewer

The pylon Viewer is included in the Basler pylon Camera Software Suite. It is a standalone application that lets you view and change most of the camera's parameter settings via a GUI-based interface. Using the pylon Viewer is a very convenient way to get your camera up and running quickly during your initial camera evaluation or doing a camera design-in for a new project.

For more information about using the viewer, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

3.1.2 Basler pylon IP Configurator

The pylon IP Configurator is included in the Basler pylon Camera Software Suite. The pylon IP Configurator is a standalone application that lets you change the IP configuration of the camera via a GUI. The tool will detect all Basler GigE cameras attached to your network and let you make changes to a selected camera.

For more information about using the IP Configurator, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

3.1.3 pylon SDKs

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

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

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

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

Available for	
acA640-90, acA640-120, acA645-100, acA750-30, acA780-75,	acA1300-22, acA1300-30, acA1600-20

Cameras with CCD sensor provide 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.
- 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 36 on page 66 for cameras with a progressive scan sensor and Figure 37 on page 67 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.

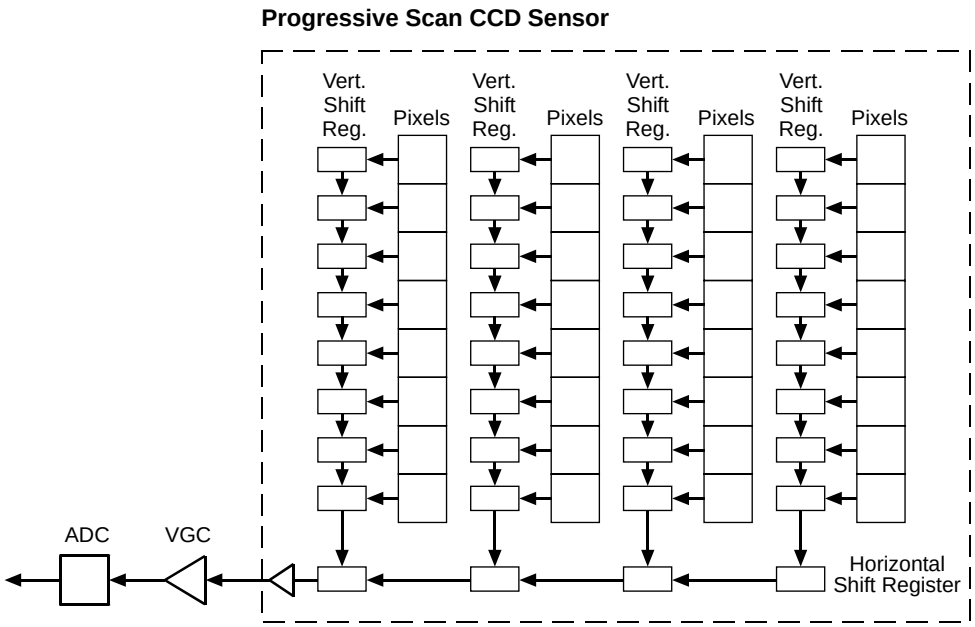


Fig. 36: CCD Sensor Architecture - Progressive Scan Sensors

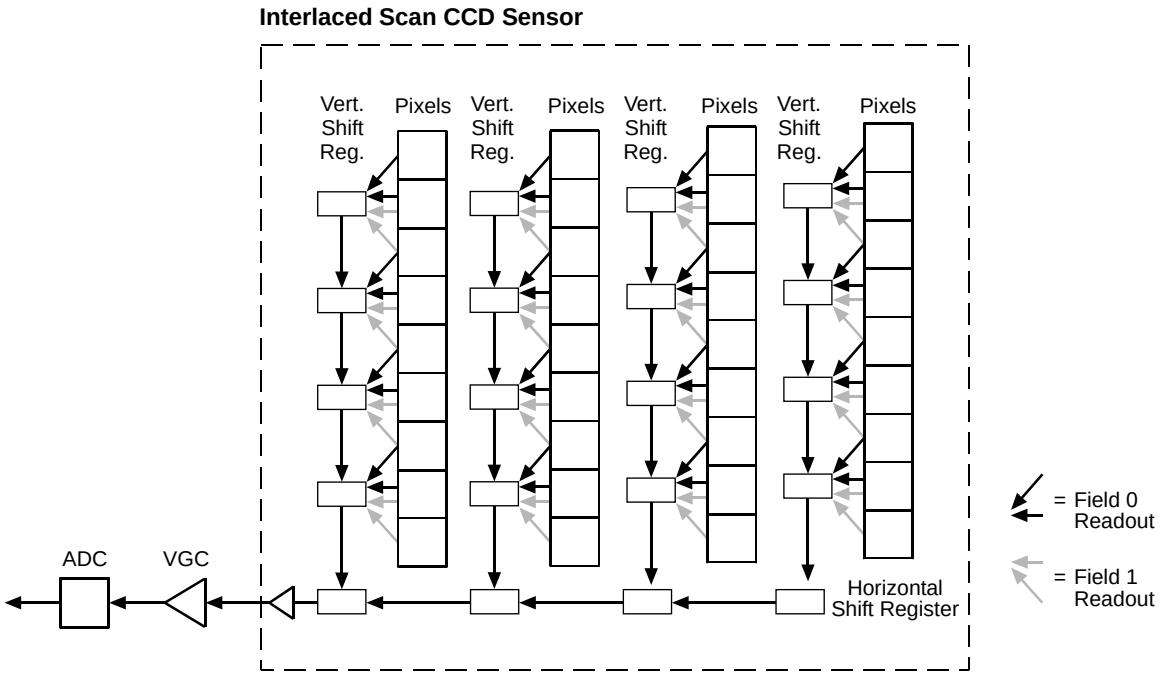


Fig. 37: CCD Sensor Architecture - Interlaced Scan Sensors

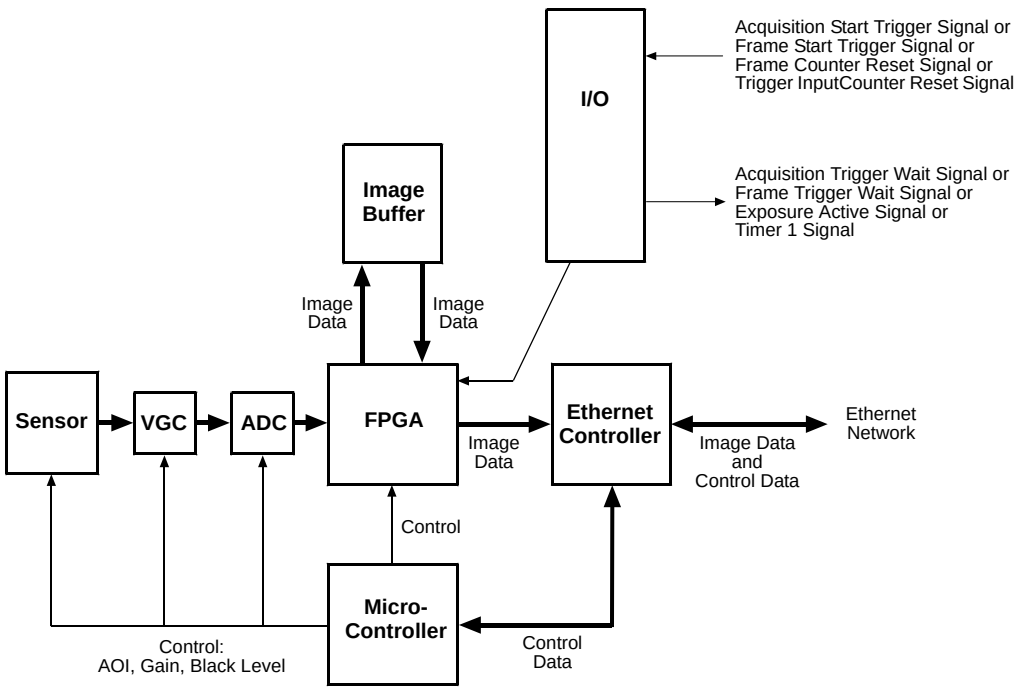


Fig. 38: Camera Block Diagram

4.2 Overview Global Shutter with CMOS Sensor

Available for		
acA640-300, acA800-200, acA1300-60 (*), acA1300-75	acA1600-60 (*), acA1920-40, acA1920-50,	acA2000-50, acA2040-25
(*) The camera models marked with an asterik (*) have a switchable shutter mode. For information, see Section 4.4 on page 72 and Section 6.7 on page 159.		

Cameras with CMOS sensor provide 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.
- 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 39 on page 69). 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.

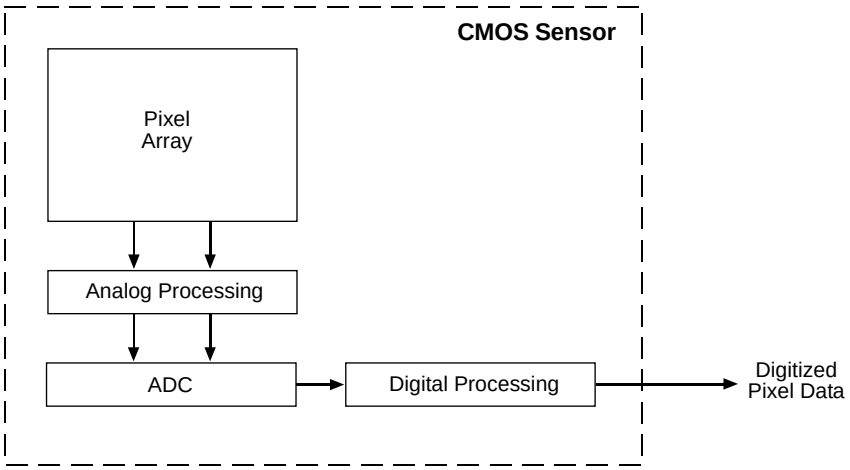


Fig. 39: CMOS Sensor Architecture

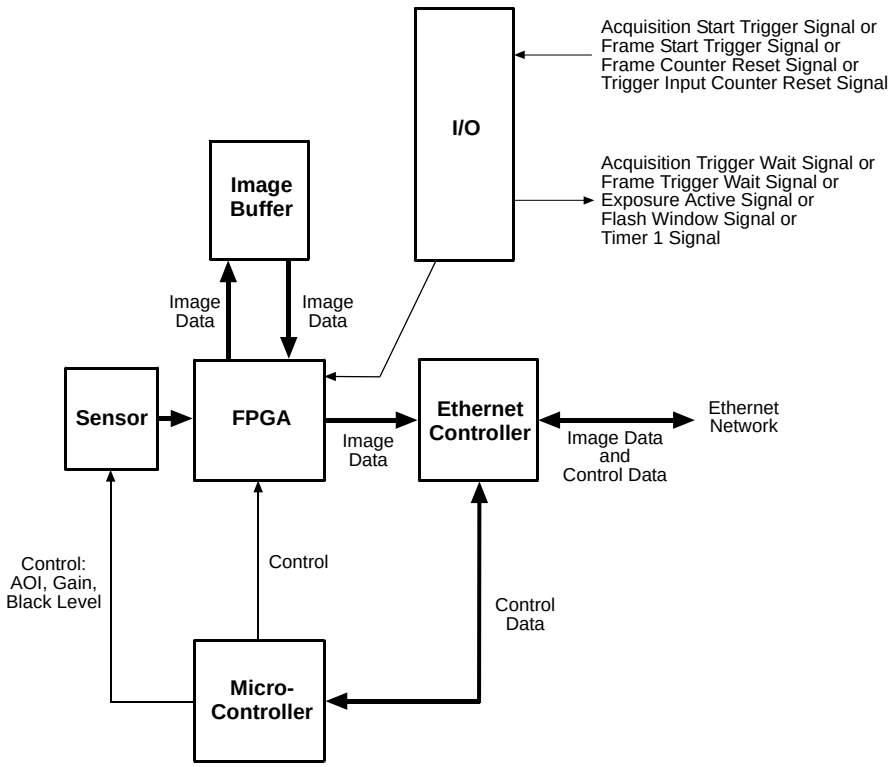


Fig. 40: Camera Block Diagram

4.3 Overview Rolling Shutter with CMOS Sensor

Available for	
acA1280-60, acA1300-60 (*), acA1600-60 (*), acA1920-25 (*),	acA2500-14 (*), acA3800-10 (*), acA4600-7 (*)
(*) The camera models marked with an asterik (*) have a switchable shutter mode. For information, see Section 4.4 on page 72 and Section 6.7 on page 159.	

Cameras with rolling shutter provide 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.
- 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 41 on page 71). 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.

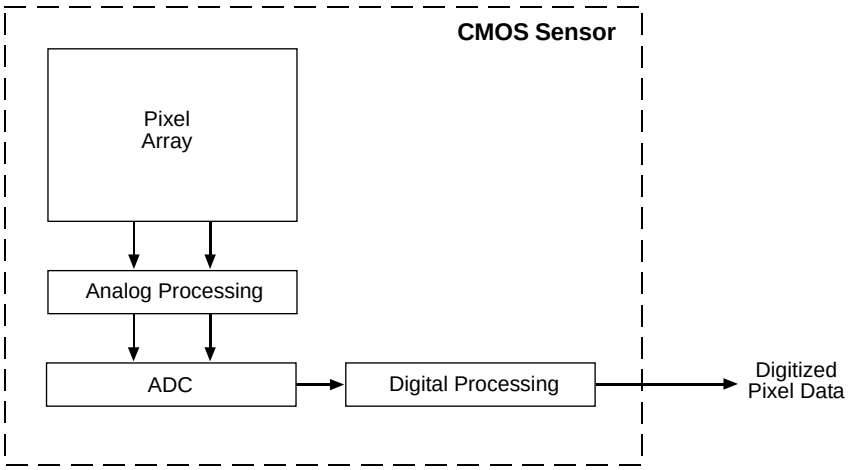


Fig. 41: CMOS Sensor Architecture

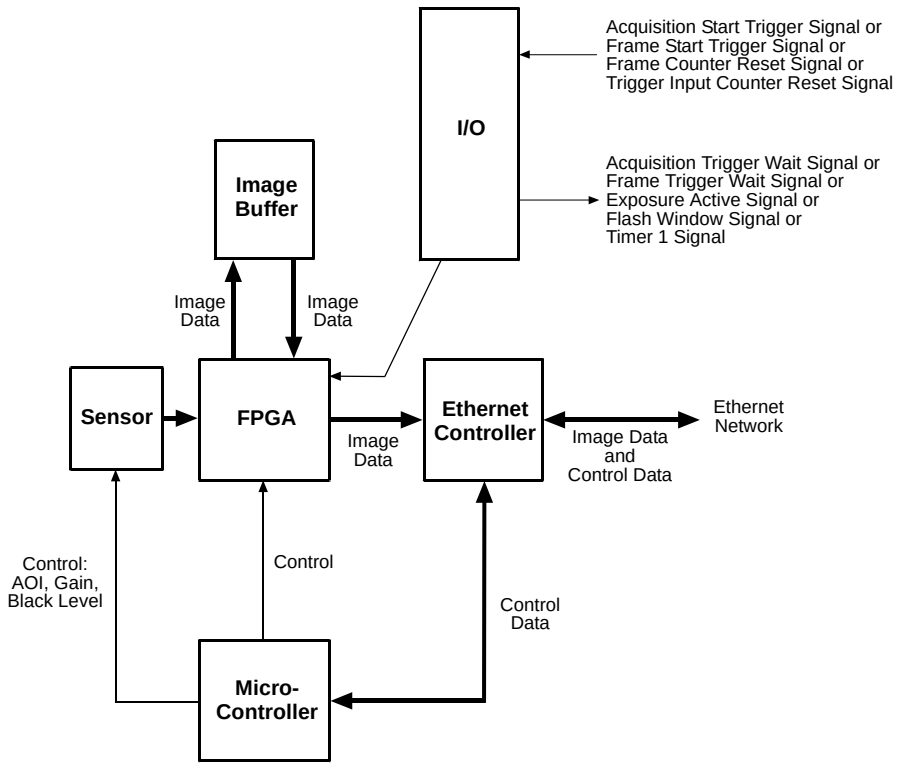


Fig. 42: Camera Block Diagram

4.4 Cameras with Switchable Shutter Mode

4.4.1 Cameras that can Switch Between Rolling and Global Shutter Mode

Available for
acA1300-60, acA1600-60

For these cameras, the following two shutter modes are available.
The cameras can be operated in the **global shutter** or **rolling shutter** mode.

By default, the shutter mode is set to global shutter mode.

Depending on your requirements you can set the camera to the desired shutter mode.

4.4.2 Cameras that can Switch Between ERS and GRR Mode

Available for
acA1300-60, acA1600-60, acA1920-25, acA2500-14, acA3800-10, acA4600-7

For these cameras, the following two rolling shutter modes are available.
The cameras can be operated in

- the **electronic rolling shutter (ERS)** or in
- the **global reset release shutter (GRR)** mode.

By default, the shutter mode is set to ERS mode.

Depending on your requirements you can set the camera to the desired shutter mode.

For information about

- the sensor architecture and global shutter mode, see Section 4.2 on page 68
- the sensor architecture and rolling shutter mode, see Section 4.3 on page 70
- electronic shutter operation in detail, see Section 6.7 on page 159
- setting the shutter mode, see page 166

5 Physical Interface and I/O Control

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

5.1 Camera Connector Types

I/O Connector (6-pin Connector)	Ethernet Connector (8-pin RJ-45 Jack)
▪ Power supply (if PoE is not used) ▪ Access to I/O lines	▪ 100/1000 Mbit/s Ethernet connection to the camera ▪ Power over Ethernet (PoE), (if power is not supplied via 6-pin connector
▪ 6-pin connector on the camera: Hirose micro receptacle (part number HR10A-7R-6PB) or equivalent. ▪ Recommended mating connector: Hirose micro plug (part number HR10A-7P-6S) or equivalent. For more information, see Section 5.4.2 on page 76.	Recommended mating connector: ▪ Any standard 8-pin RJ-45 plug (snap-in) or ▪ 8-pin RJ-45 plug with locking screws To ensure that you order cables with the correct connectors, note the horizontal orientation of the screws before ordering.

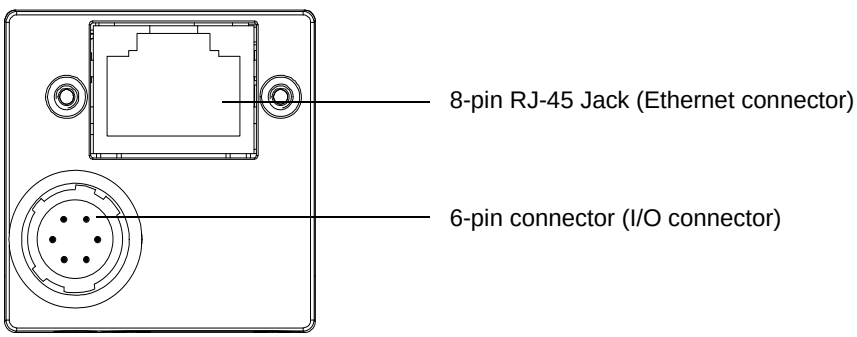


Fig. 43: Camera Connectors

5.2 Which Camera Model Has GPIO?

GPIO = General Purpose I/O

Depending on the camera model, pin 3 of the 6-pin connector can either be used as GPIO line, or it is not used:

- Most ace GigE camera models don’t have any GPIO line. They have one opto-isolated input line and one opto-isolated output line. These camera models do not use pin 3 of the 6-pin connector.
- For some camera models you can use pin 3 of the 6-pin connector (I/O) as a direct-coupled GPIO line.

The following tables shows which camera model has a GPIO line or which model has no GPIO line.

Camera Models with GPIO Line [Pin 3 used as GPIO line]	Camera Models without GPIO Line [Pin 3 not used]
acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50	All other models

Table 16: Camera Models with or without GPIO Line

5.3 Camera Connector Pin Numbering and Assignments

5.3.1 I/O Connector Pin Numbering and Assignments

Pin	Designation	Function for Cameras without GPIO
1	-	+12 VDC Camera Power
2	Line1	Opto-isolated IN
3	-	Not Connected
4	Out1	Opto-isolated OUT
5	-	Opto-isolated I/O Ground
6	-	DC Camera Power Ground

Table 17: Pin Assignments for the I/O Connector (Cameras without GPIO)

Pin	Designation	Function for Cameras with GPIO
1	-	+12 VDC Camera Power
2	Line1	Opto-isolated IN
3	Line3	GPIO (direct-coupled General Purpose I/O)
4	Line2	Opto-isolated OUT
5	-	Opto-isolated I/O Ground
6	-	DC Camera Power Ground and GPIO Ground

Table 18: Pin Assignments for the I/O Connector (Cameras with GPIO)

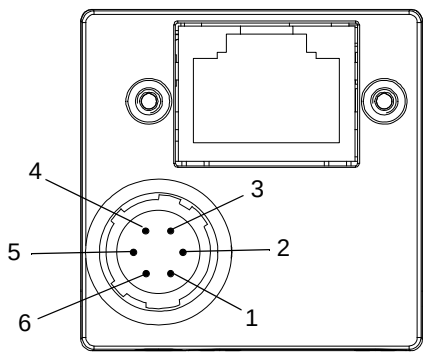


Fig. 44: Pin Numbering for the I/O Connector

5.3.2 Ethernet Connector Pin Numbering and Assignments

The Ethernet connector is an 8-pin RJ-45 jack.
Pin numbering and assignments adhere to the Ethernet standard and IEEE 802.3af.

5.4 Camera Cabling Requirements

5.4.1 Ethernet Cable

Use

- high-quality Ethernet cables.
  Use of shielded CAT 5E or better cables with S/STP shielding is recommended.
- either straight-through (patch) or a cross-over Ethernet cable to connect the camera directly to a GigE network adapter in a PC or to a GigE network switch.

As a general rule, applications with longer cables or applications in harsh EMI conditions require higher category cables.

Close proximity to strong magnetic fields should be avoided.

5.4.2 I/O Cable

Recommendations for the I/O cable.

- The I/O cable must be shielded.
- The I/O cable must have at least 0.14 mm^2 (close to AWG26).
- You should use a twisted pair wire.
- Maximum recommended cable length: 10 m
- Cable end: Hirose micro plug (part number HR10A-7P-6S) or equivalent
- Pin assignment (see Table 16 and Table 17 on page 75)
- You have to observe the applicable voltage levels in Table 19 on page 81.
- Close proximity to strong magnetic fields should be avoided.

Depending on the particular application, using different cables may lead to voltage drops, signal distortion, and EMI/ESD problems which in turn may cause the camera to malfunction.

If you are supplying power to the camera via Power over Ethernet, the power and I/O cable will not be used to supply power to the camera, but 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 Ethernet connector (PoE) **or** via the camera's I/O connector.

	Direct-coupled GPIO lines have the advantage of working with very short delays compared to opto-isolated I/O lines. The direct-coupled GPIO is, however, distinctly more susceptible to EMI than the opto-isolated I/Os. Under harsh EMI conditions, the GPIO can turn out not to be usable at all. We therefore strongly recommend to only use the direct-coupled GPIO line when significant electromagnetic interference will not occur. For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.

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

	Basler offers suitable plugs and cables. Contact your Basler sales representative to order connectors or cables.

5.5 Camera Power

Via PoE (Power over Ethernet)	Power via power and I/O cable
▪ Via Ethernet cable plugged into camera's Ethernet connector (RJ-45 connector). ▪ Power must adhere to the requirements specified in IEEE 802.3af.	▪ From a power supply via a cable plugged into the camera's I/O connector. ▪ Nominal operating voltage: 12 VDC (± 10%) with less than one percent ripple
▪ Power consumption -> see specification tables in Section 1 of this manual. ▪ Close proximity to strong magnetic fields should be avoided.	

DANGER

Electric Shock Hazard

Unapproved power supplies may cause electric shock. Serious injury or death may occur.

- You must use camera power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the SELV and LPS requirements.

WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

- You must use camera power supplies which meet the Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the LPS requirements.

NOTICE

Voltage outside of the specified range can cause damage.

- If you are supplying camera power via Power over Ethernet (PoE), the power must comply with the IEEE 802.3af specification.
- 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 +10.8 VDC, the camera may operate erratically.
- The ace GigE cameras must only be connected to other limited power sources (LPS) / Safety Extra Low Voltage (SELV) circuits that do not represent any energy hazards.

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 5.6.1 on page 80 and Section 5.7.1 on page 83).
- for Basler ace GigE cameras can differ from the recommended voltage range for camera power for other Basler cameras.

5.6 Opto-isolated Input (Pin 2)

The camera is equipped with one dedicated opto-isolated input line. The designation depends on the camera model; see the table below.

	Camera Models without GPIO Line	Camera Models with GPIO Line
Designation of the input line	Line1	Line1
For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.		

The input line is accessed via the I/O connector on the back of the camera (see Figure 44 on page 75).

5.6.1 Electrical Characteristics

DANGER

Electric Shock Hazard

Unapproved power supplies may cause electric shock. Serious injury or death may occur.

- You must use camera power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the SELV and LPS requirements.

WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

- You must use camera power supplies which meet the Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the LPS requirements.

Basler offers suitable and tested power supplies for PoE as well as power over the I/O connector.

NOTICE

Voltage outside of the specified range can cause damage.

The recommended voltage range

- for the opto-isolated input line differs from the recommended voltage ranges for the output line (see Section 5.7.1 on page 83).
- 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.

The following voltage requirements apply to the camera's I/O input line (pin number, see Figure 44 on page 75):

Input Voltage	Significance
+30.0 VDC	Absolute maximum. The absolute maximum must never be exceeded. Otherwise, the camera can be damaged and the warranty becomes void.
+0 to +24 VDC	Safe operating I/O input voltage range.
+0 to +1.4 VDC	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.

Table 19: Voltage Requirements and Information for the I/O Input Line

If the camera is connected to a PLC device, we recommend using a special cable that adjusts the voltage level of a PLC to the camera.

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.As shown in Figure 45, the input line is opto-isolated. See the previous section for input voltages and their significances. The current draw for each input line is between 5 mA and 15 mA.

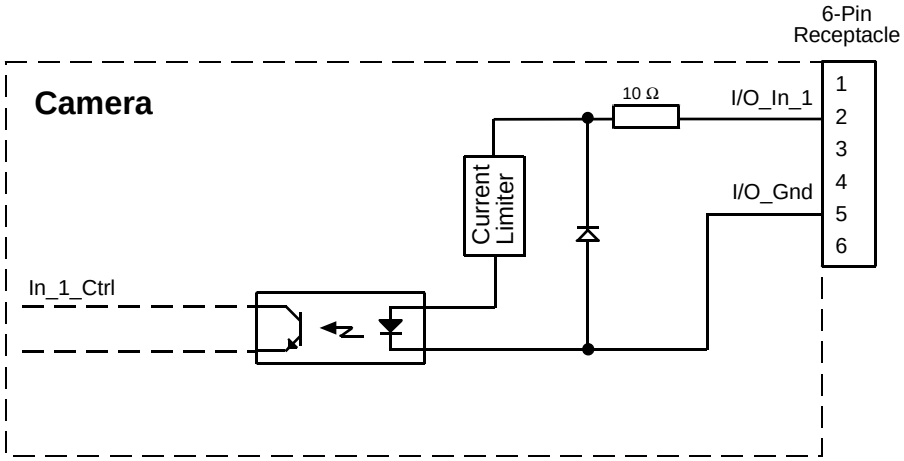


Fig. 45: Input Line Schematic (Simplified)

Figure 46 shows an example of a typical circuit you can use to input a signal into the camera.

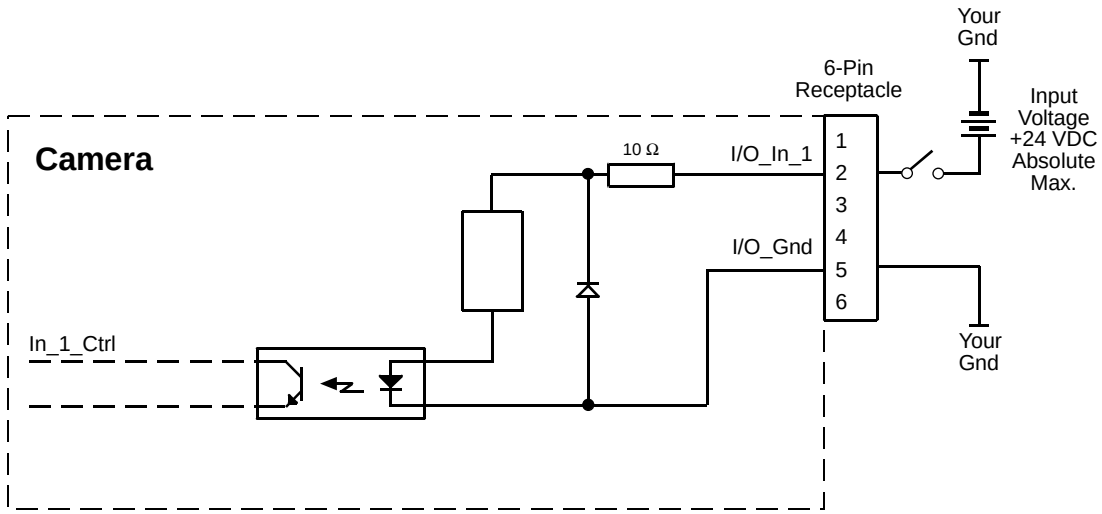


Fig. 46: Typical Input Circuit (Simplified)

For more information about

- input line pin numbering and pin assignments, see Section 5.3 on page 75.
- how to use an externally generated frame start trigger (ExFSTrig) signal to control acquisition start, see Section 6.4.3 on page 142.
- configuring the input line, see Section 5.10 on page 98.

5.7 Opto-isolated Output (Pin 4)

The camera is equipped with one physical output line. The designation depends on the camera model; see the table below.

	Camera Models without GPIO Line	Camera Models with GPIO Line
Designation of the output line	Out1	Line2
For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.		

The output line is accessed via the I/O connector (6-pin connector) on the back of the camera.

5.7.1 Electrical Characteristics

DANGER

Electric Shock Hazard

Unapproved power supplies may cause electric shock. Serious injury or death may occur.

- You must use camera power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the SELV and LPS requirements.

WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

- You must use camera power supplies which meet the Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the LPS 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 78) and for the input line (see Section 5.6.1 on page 80). You must supply power within the specific voltage range.
- 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 number, see Figure 44 on page 75):

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

Table 20: Voltage Requirements and Information for the I/O Output

As shown in Figure 47, the output line is opto-isolated. See the previous section for the recommended operating voltages. The **maximum continuous 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.

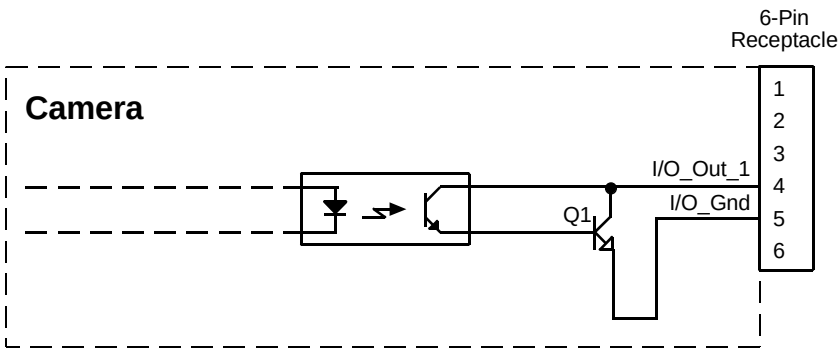


Fig. 47: Output Line Schematic (Simplified)

On early production cameras, 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 5.11.7 on page 113.

Figure 48 shows a typical circuit you can use to monitor the output line with a voltage signal.

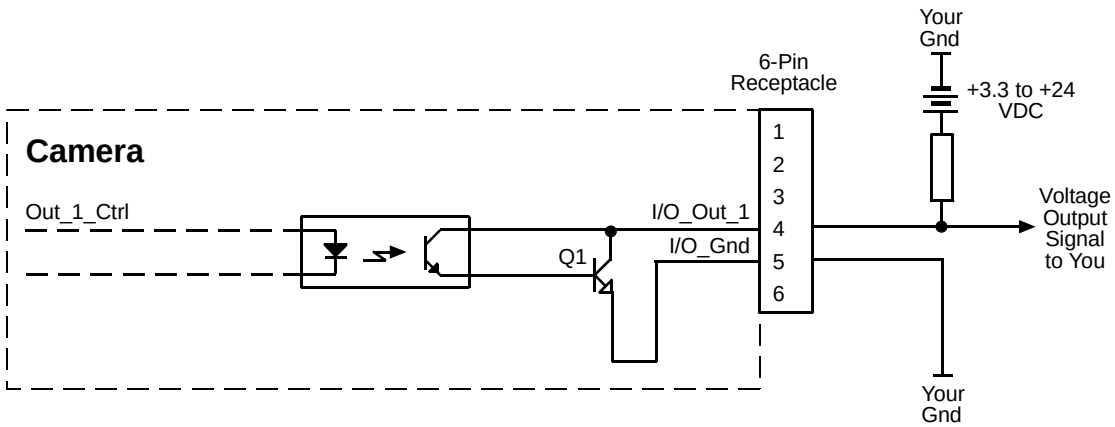


Fig. 48: Typical Voltage Output Circuit (Simplified Example)

Figure 49 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.

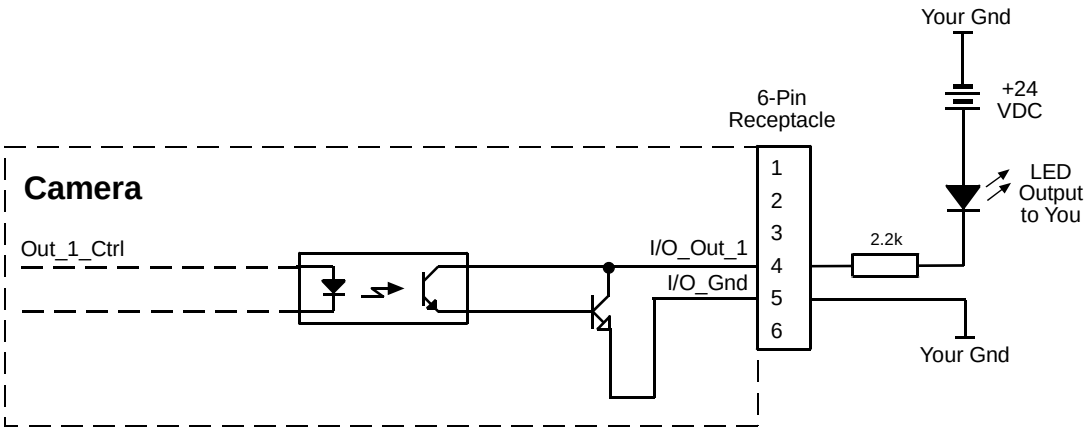


Fig. 49: Typical LED Output Signal at +24 VDC for the External Circuit (Simplified Example)

By default, the camera's Exposure Active signal is assigned to the opto-isolated output line:

- Out1 --> for cameras without GPIO
- Line2 --> for cameras with GPIO

The assignment of a camera output signal to Out1 (cameras with GPIO: Line2) can be changed by the user. For more information about assigning camera output signals to an output line, see Section 5.11.1 on page 102.

For more information about

- output line pin assignments and pin numbering, see Section 5.3 on page 75.
- the Exposure Active signal, see Section 6.10 on page 177.

5.8 General Purpose I/O (Only Available for Certain Cameras)

For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.

By default, the GPIO line is set to operate as input to the camera.

5.8.1 Introduction

Certain Basler ace GigE cameras have one direct-coupled GPIO line that is accessed via pin 3 of the 6-pin connector on the back of the camera (see Figure 44 on page 75).

The GPIO line

- can be set to operate as an input to the camera or to operate as an output.
- is designated as Line 3 (see also Section 5.3.1 on page 75).
- is a direct-coupled GPIO line and is compatible with TTL signals.

The next sections describe the differences in the GPIO electrical functionality when the line is set to operate as input and when it is set to operate as output.

DANGER

Electric Shock Hazard

Unapproved power supplies may cause electric shock. Serious injury or death may occur.

- You must use camera power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the SELV and LPS requirements.

WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

- You must use camera power supplies which meet the Limited Power Source (LPS) requirements.
- If you use a powered hub or powered switch, they must meet the LPS requirements.

NOTICE

Applying incorrect electrical signals to the camera's GPIO line can severely damage the camera.

1. Before you connect any external circuitry to the GPIO line, we strongly recommend that you set the GPIO line to operate as an input or as an output (according to your needs).
2. Once the line is properly set, make sure that you only apply electrical signals to the line that are appropriate for the line's current setting.

Direct-coupled GPIO lines have the advantage of working with very short delays compared to opto-isolated I/O lines.

The direct-coupled GPIO is, however, distinctly more susceptible to EMI than the opto-isolated I/Os. Under harsh EMI conditions, the GPIO can turn out not to be usable at all.

We therefore strongly recommend to only use the direct-coupled GPIO line when significant electromagnetic interference will not occur.

For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.

For more information about

- GPIO pin assignments and pin numbering, see Section 5.3.1 on page 75.
- setting the GPIO line operation, see Section 5.8.2 on page 89 and Section 5.8.3 on page 91.

5.8.2 Operation as an Input

5.8.2.1 Electrical Characteristics

NOTICE
Voltage outside of the safe operating voltage range can cause damage. You must supply power within the safe operating voltage range.

The following I/O supply voltage requirements apply to the direct-coupled GPIO line when the line is set as an input.

Voltage	Significance
+30.0 VDC	Absolute maximum. The absolute maximum must never be exceeded. Otherwise, the camera can be damaged and the warranty becomes void.
+0 to + 5.0 VDC	Safe operating input voltage range (the minimum external pull up voltage is 3.3 VDC).
+0 to +0.8 VDC	The voltage indicates a logical 0.
> +0.8 to +2.0 VDC	Region where the transition threshold occurs; the logical status is not defined in this region.
> +2.0 VDC	The voltage indicates a logical 1.

Table 21: Voltage Requirements for the Direct-coupled GPIO Line Set as an Input

Your application must be able to accept 2 mA (sink current) from the direct-coupled GPIO input line without exceeding +0.8 VDC, the upper limit of the low status. The current draw for high-level input current is < 100 µA.

Figure 50 shows the applicable electrical circuit when a GPIO line is set to operate as an input.

The figure shows, as an example, the use of a TTL or CMOS logic gate in the external circuit. A different example for an external circuit is shown in Figure 51.

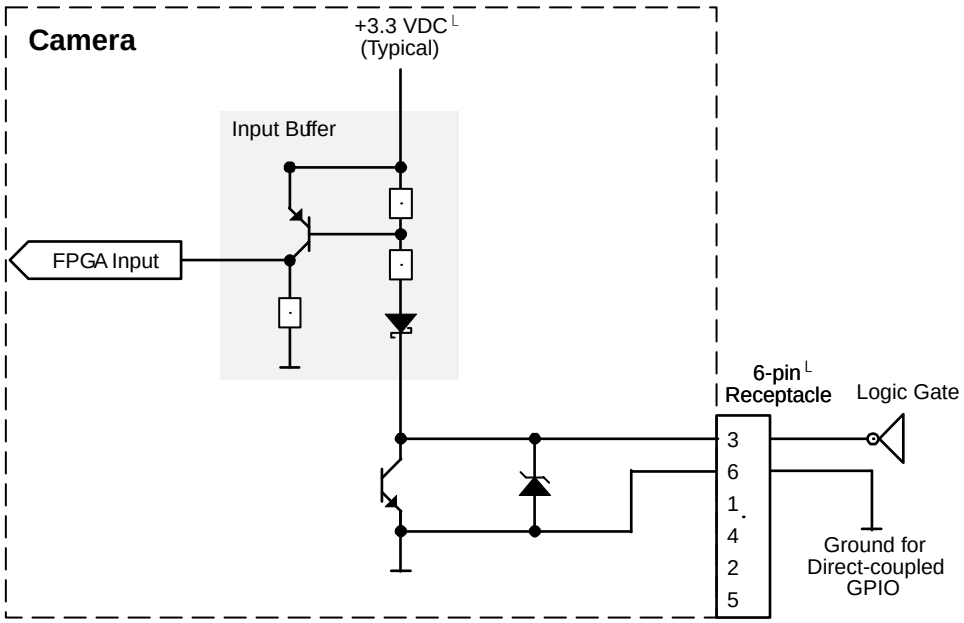


Fig. 50: Direct-coupled GPIO Line Schematic with the GPIO Line Set as an Input and with a Typical External Circuit Using a Logic Gate (Simplified)

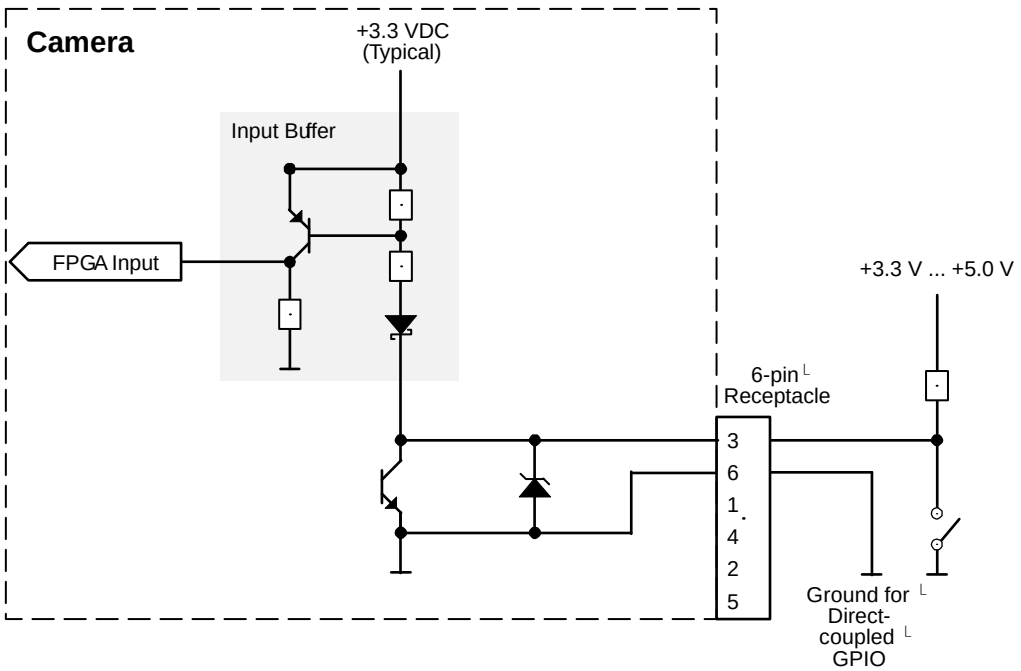


Fig. 51: Direct-coupled GPIO Line Schematic with the GPIO Line Set as an Input and with a Typical External Circuit (Simplified)

For more information about

- GPIO pin assignments and pin numbering, see Section 5.3.1 on page 75.
- setting the GPIO line operation, see Section 5.8.2 on page 89 and Section 5.8.3 on page 91.

5.8.3 Operation as an Output

5.8.3.1 Electrical Characteristics

NOTICE
Voltage outside of the specified range can cause damage. You must supply power within the specified voltage range.

	To ensure that the specified voltage levels for signals transmitted out of the camera will be reached even under less than favorable conditions (e.g. for long cable lengths, harsh EMI environment, etc.), we recommend to generally use an external pull up resistor or to connect a "high side load".

- The following I/O supply voltage requirements apply to the direct-coupled GPIO line when it is set as an output and when it is in the **"off"** state:

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

Table 22: Voltage Requirements for the Direct-coupled GPIO Line Set as an Output

- The following applies to the direct-coupled GPIO line when it s set as an output and when it is in the **"on"** state:

  The camera uses an open collector with only a weak internal pull-up resistor (approximately 2 kΩ). It is therefore likely that many applications will have to provide an additional pull-up resistor.

  The residual voltage will typically be approximately 0.4 V at 50 mA and 25 °C housing temperature. The actual residual voltage, however, depends on camera operating temperature, load current, and production spread.

Note: The maximum current allowed through the output circuit is **50 mA**.

Currents

- The leakage current in the "off" state should usually not exceed approximately 60 µA and will typically be much lower (e.g. approximately 4 µA at 25 °C (+77 °F) housing temperature). The actual leakage current depends on camera operating temperature and production spread of electronic components.
- The **maximum** load current allowed through the output circuit is **50 mA**.
- There is no specific minimum load current but you need to consider several facts:
  - the leakage current will have stronger effect when load currents are low
  - the propagation delay of the output increases as load currents decrease
  - higher-impedance circuits tend to be more susceptible to EMI
  - higher currents yield higher voltage drop on long cables.

As shown in Figure 52, shows the applicable electrical circuit when a GPIO line is set to operate as an output.

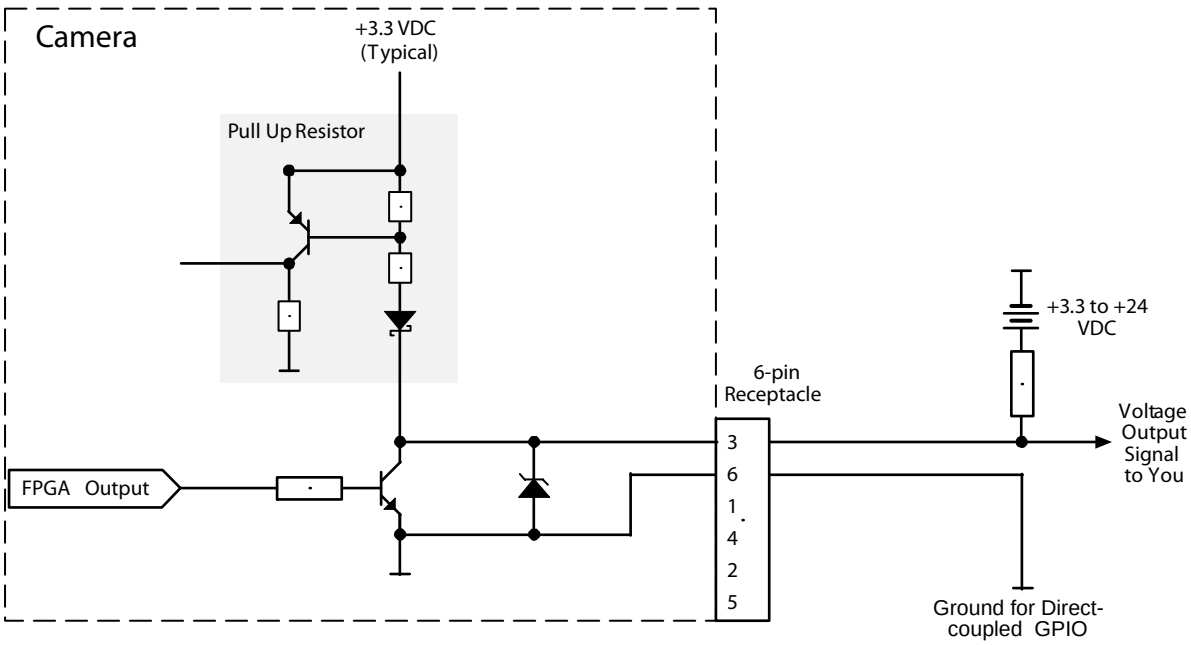


Fig. 52: Direct-coupled GPIO Line Schematic with the GPIO Line Set as an Output and with a Typical Voltage Output Circuit (Simplified)

For more information about

- GPIO pin assignments and pin numbering, see Section 5.3.1 on page 75.
- setting the GPIO line operation, see Section 5.10 on page 98 and Section 5.11 on page 102.

5.9 Temporal Performance of I/O Lines

This section describes delays ("propagation delays") resulting from the operation of the camera's input and output lines. For image acquisition, the propagation delays must be added to the delays described in Section 6 on page 121.

You will need the information included in this section most likely only if you need microsecond accuracy when controlling camera operation via I/O lines.

All examples in this section assume that the I/O line inverters are disabled.

5.9.1 Introduction

As indicated in Section 5.3 on page 75, the camera provides two different kinds of I/O lines:

- opto-isolated I/O lines
- a direct-coupled General Purpose I/O (GPIO) line.
  Only available on some cameras; see Table 16 on page 74.

The related electrical characteristics and circuit schematics are given in Section 5.6 through Section 5.8.

With regard to use, the two kinds of I/O lines differ mainly in these respects:

- The opto-isolated I/O lines have the advantage of being distinctly more robust against EMI than the GPIO line.
- The propagation delays ("response times") differ between the two kinds of I/O lines.

  A propagation delay is the time that elapses between the moment when a signal voltage passes through the transition threshold and the moment when the related line status changes – or vice versa (see Figure 53 for camera input and Figure 54 for camera output).

The following important characteristics are apparent from Figure 53 and Figure 54:

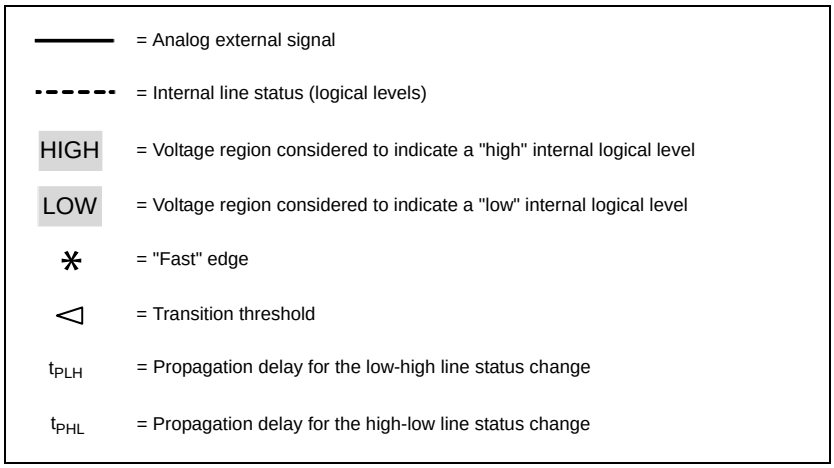
- The propagation delays for the opto-isolated I/O lines are in most cases longer than for the GPIO line. In other words, the opto-isolated I/O lines are usually "slower" than the GPIO line.
- For each analog signal, the rising edge and the falling edge are associated with different propagation delays. The edge with the shorter propagation delay (the "fast" edge) is indicated by an asterisk.

Note: In order to avoid loosing an external trigger signal make sure its pulse width will be long enough to provide sufficient time for the camera's input circuit to react:

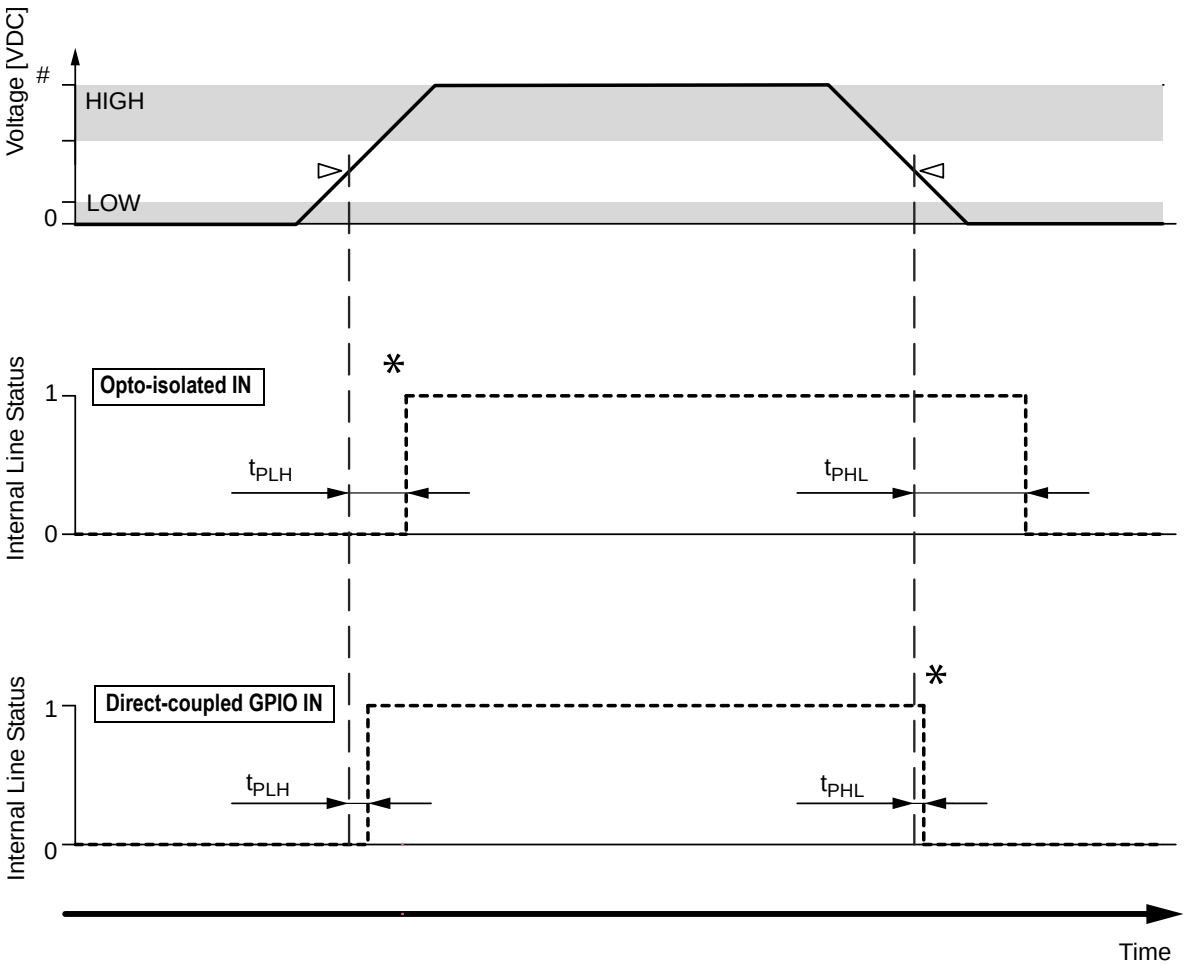
The minimum required pulse width will be longer for the

- opto-isolated input line compared to a GPIO line and for a
- trigger signal using the active low state for triggering compared to a trigger signal using the active high state.

As a general rule of thumb, an external trigger pulse width of 100 µs should be long enough for most cases.



Drawing not to scale



#: 3.3 - 24 VDC for opto-isolated input, >2.0 - 5.0 VDC for direct-coupled GPIO IN

Fig. 53: Analog External Signal and Associated Internal Line Status with Propagation Delays for Opto-isolated Input and Direct-coupled GPIO Inputs (Line Inverters Disabled)

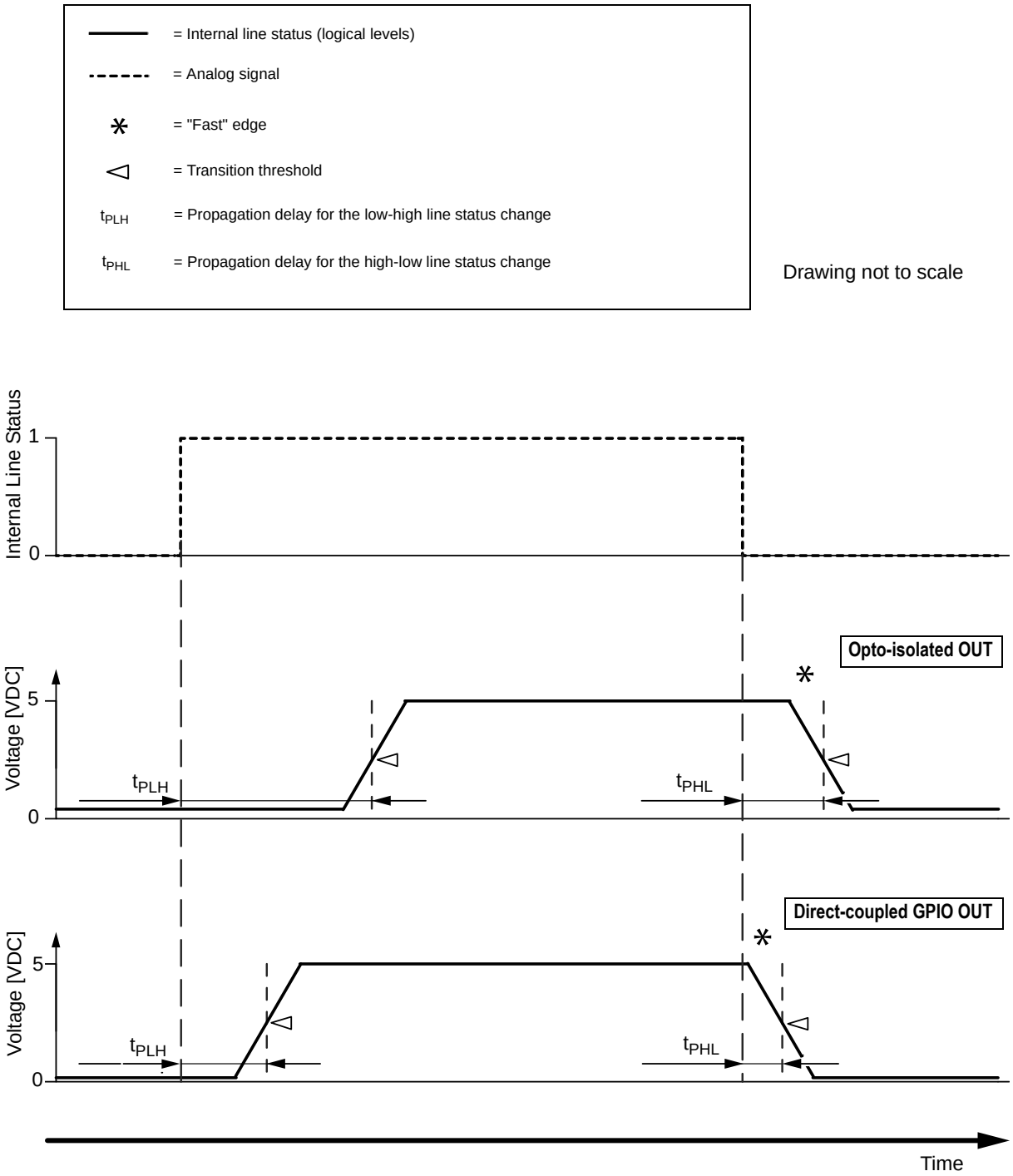


Fig. 54: Internal Line Status and Associated Output Signals with Propagation Delays for Opto-isolated Output and Direct-coupled GPIO Output (Line Inverters Disabled)

5.9.2 Factors Determining I/O Temporal Performance

A number of factors control the exact durations of propagation delays. The influence for some of the factors is, however, ill constrained or unknown. As a consequence, **generally valid and exact quantitative predictions of propagation delays are impossible**.

The following factors apply:

Factors Influencing Camera I/O Propagation Delays	Input		Output	
	Opto-isolated IN	Direct-coupled GPIOs	Opto-isolated OUT	Direct-coupled GPIOs
Operating temperature: Unknown but temperature must be within specified range; see Section 1.7.1 on page 55.	•	o	•	o
Production spread: Unknown	•	o	•	o
Aging (optocouplers): Unknown	•		•	
External I/O supply voltage: Depends on application but must be within specified ranges; see Section 5.6 through Section 5.8.	•		•	o
Load resistance: Depends on application			•	o
Load current: Depends on application but must be within specified ranges; see Section 5.6 through Section 5.8.			•	o

Table 23: Factors Influencing Camera I/O Propagation Delays (• = major influence, o = minor influence)

5.9.3 Recommendations for Using Camera I/Os

Adhering to the following recommendations will help you to achieve efficient and stable camera operation when using the camera's I/O lines. When reading the recommendations, also see Figure 53 and Figure 54.

Opto-isolated I/Os and Direct-coupled GPIOs

- Use the "fast" edge of a signal for tight temporal control and to minimize unwanted influence on propagation delays in general.

  The propagation delays for a "fast" edge will rarely exceed 15 µs for an opto-isolated I/O line, and rarely 1 µs for a direct-coupled GPIO line. Under very unfavorable conditions, propagation delays related to "slow" edges can take milliseconds.

  - To minimize propagation delays related to a "fast" edge, increase the load resistance.
  - To minimize propagation delays related to a "slow" edge, use an I/O supply voltage between 3.3 VDC and 5 VDC and decrease the load resistance such that a load current between 30 mA and 40 mA will result.
- Use the direct-coupled GPIO line when you need to minimize propagation delays but mind their greater susceptibility to EMI compared to the opto-isolated I/Os.

Opto-isolated I/Os

When you apply current to the input and output lines for extended periods or even for most of the time you will promote aging of the optocouplers. Keep the times when current flows to a minimum to preserve stable propagation delays.

	Signal edge-to-edge variation (jitter) resulting from I/O operation itself is negligible but can be introduced by your trigger signal. To avoid jitter, make sure the slopes of your trigger signals are short, preferably < 500 ns. The camera's inherent jitter is less than 100 ns, peak to peak.

5.10 Configuring the Input Line

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

The camera is equipped with one physical input line. The designation depends on the camera model; see the table below.

	Camera Models without GPIO Line	Camera Models with GPIO Line
Designation of the input line	Line1	Line1
For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.		

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

Camera Function	If the input line is selected for the camera function, whenever a proper electrical signal is applied to the line, ...
Acquisition Start Trigger	... the camera will recognize the signal as an acquisition start trigger signal., For detailed information, see Section 6.3 on page 128
Frame Start Trigger	...the camera will recognize the signal as a frame start trigger signal. For detailed information, see Section 6.4.3 on page 142.
Frame Counter Reset	...the counter value for the frame counter chunk feature will be reset. For detailed information, see Section 10.3 on page 368
Trigger Input Counter Reset	...the counter value for the trigger reset counter chunk feature will be reset. For detailed information, see Section 10.3 on page 368

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 the electrical characteristics of the input line, see Section 5.6 on page 80.

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

5.10.2 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 55 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.

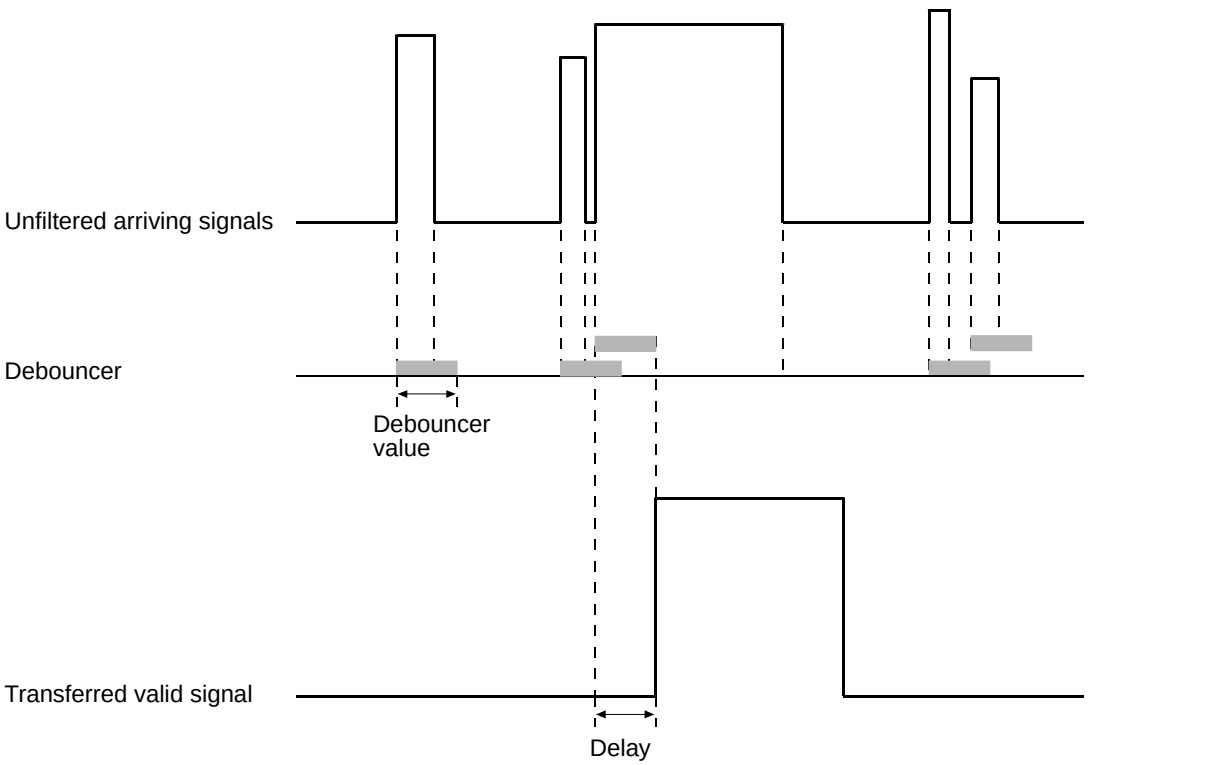


Fig. 55: Filtering of Input Signals by the Debouncer

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:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select input line1. 2. Set the value of the LineDebouncerTimeAbs parameter.	1. Use the LineSelector parameter to select Line1. 2. Set the LineMode parameter to Line1. 3. Set the value of the LineDebouncerTimeAbs parameter.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
// Select the input line
Camera.LineSelector.SetValue(LineSelector_Line1);

// Set the parameter value to 10 microseconds
Camera.LineDebouncerTimeAbs.SetValue(10.0);
```

For Models with GPIO

```
// Select the input line
Camera.LineSelector.SetValue(LineSelector_Line1);

// Set the line mode
Camera.LineMode.SetValue(LineMode_Input);

// Set the parameter value to 10 microseconds
Camera.LineDebouncerTimeAbs.SetValue(10.0);
```

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

5.10.3 Setting the Input Line for Invert

You can set the input line and the GPIO line to invert or not to invert the incoming electrical signal.

To set the invert function on the input line:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select the input line. 2. Set the value of the LineInverter parameter to true to enable inversion on the selected line or to false to disable inversion.	1. Use the LineSelector parameter to select Line1. 2. Set the LineMode to Input. 3. Set the value of the LineInverter parameter to true to enable inversion on the selected line or to false to disable inversion.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

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

For Models with GPIO

```
// Select the input line
Camera.LineSelector.SetValue(LineSelector_Line1);

// Set the line mode
Camera.LineMode.SetValue(LineMode_Input);

// Enable the inverter on line 1
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 on page 63.

5.11 Configuring the Output Line

5.11.1 Selecting a Source Signal for the Output Line

The camera is equipped with one physical output line. The designation depends on the camera model; see the table below.

	Camera Models without GPIO Line	Camera Models with GPIO Line
Designation of the output line	Out1	Line2
For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.		

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 output line 1(Line2). The camera has five standard output signals available. The designation depends on the camera model:

Camera Models without GPIO Line	Camera Models with GPIO Line
Exposure Active	
Frame Trigger Wait	
User Output	User Output x
Acquisition Trigger Wait	
Flash Window	
Timer Active	Timer 1 Active
Sync User Output	Sync User Output x
X: 1, 2, or 3	

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 set a camera output signal as the source signal/ to set a line as user settable:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select Output Line 1. 2. Set the value of the LineSource parameter to one of the available output signals or to user settable. This will set the source signal for the output line.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Set the value of the LineSource parameter to one of the available output signals or to user settable. This will set the source signal for the output line.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
Camera.LineSelector.SetValue(LineSelector_Out1);
Camera.LineSource.SetValue(LineSource_ExposureActive);
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
Camera.LineSource.SetValue(LineSource_ExposureActive);
```

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

By default, the camera's Exposure Active signal is assigned to the opto-isolated output line: For

- cameras without GPIO --> Out1
- cameras with GPIO --> Line2

For more information about

- the pylon API and the pylon Viewer, see Section 3.1.1 on page 63.
- the acquisition trigger and frame trigger wait signals, see Section 6.10.4 on page 183.
- the exposure active signal, see Section 6.10.1 on page 177.
- the flash window signal, see Section 6.7.2 on page 162 and Section 6.10.3 on page 182.
- working with a timer output signal, see Section 5.11.8 on page 114
- setting the state of a user settable output line, see Section 5.11.3 on page 106.
- the sync user output signal, see Section 5.11.5 on page 109.
- the electrical characteristics of the output line, see Section 5.7 on page 83.

5.11.2 Minimum Output Pulse Width

Available for	Not Available for
All models	acA640-90, acA640-100, acA750-30, acA1300-30, acA1600-20, acA2500-14

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).
- 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.

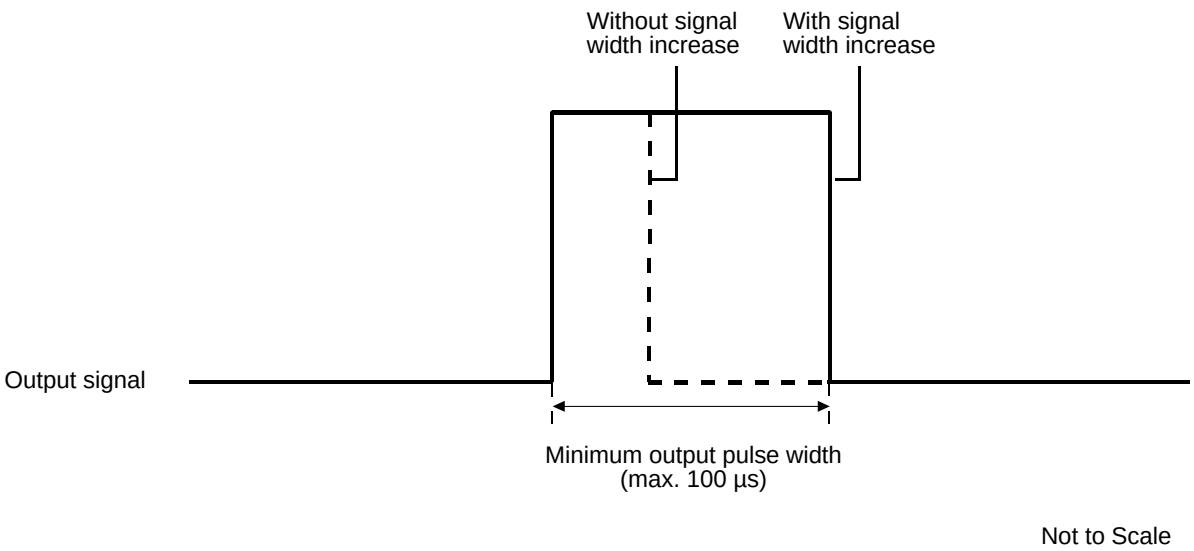


Fig. 56: Increasing the Signal Width of an Output Signal

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:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Set the value of the MinOutPulseWidthAbs parameter.	1. Use the LineSelector parameter to the desired output line (e.g. Line2). For line 2 the LineMode parameter is automatically set to Output. 2. Set the value of the MinOutPulseWidthAbs parameter. If you want to use the GPIO line, you will have to set the LineMode parameter before setting the MinOutPulseWidthAbs parameter.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
// Select the output line
Camera.LineSelector.SetValue(LineSelector_Out1);

// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);
```

For Models with GPIO

```
// Select the output line
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);

// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);
```

If you want to use the GPIO line (line 3), you will have to set the LineMode parameter before setting the MinOutPulseWidthAbs parameter.

```
// Select the output line
Camera.LineSelector.SetValue(LineSelector_Line3);
Camera.LineMode.SetValue(LineMode_Output);

// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);
```

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 63.

5.11.3 Setting the State of a User Settable Output Line

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. This means that you can assign a state (high or low) to a line using the User Output Value parameter.

You can use this to control external events or devices, e.g. a light source. Note that this is a non-sequence parameter and therefore its values can't be changed using the sequencer feature.

If you want to achieve the same result with the sequencer, you need to set the Sync User Output Value parameter instead (see Section 5.11.5 on page 109).

To set the state of a user settable output line:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select Output Line 1. 2. Set the UserOutputValue parameter to true (1) or false (0). This will set the state of the output line.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Set the UserOutputValue parameter to true (1) or false (0). This will set the state of the output line. *N* = 1, 2 ...

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
// 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( );
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
Camera.LineSource.SetValue(LineSource_UserOutput1);
// 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 63.

5.11.4 Setting and Checking the State of All User Settable Output Lines

The UserOutputValueAll parameter reports the current state of all user settable output signals. The parameter value is expressed as a hexadecimal number by the Basler pylon Viewer and as a 32-bit word in the Basler pylon API. The number is derived from bits 0 and 12 of a 32-bit word and is based on all current binary UserOutputValue parameter values.

As shown in Figure 58, certain bits are associated with certain lines. The states of those lines are expressed by the related binary UserOutputValue parameter values.

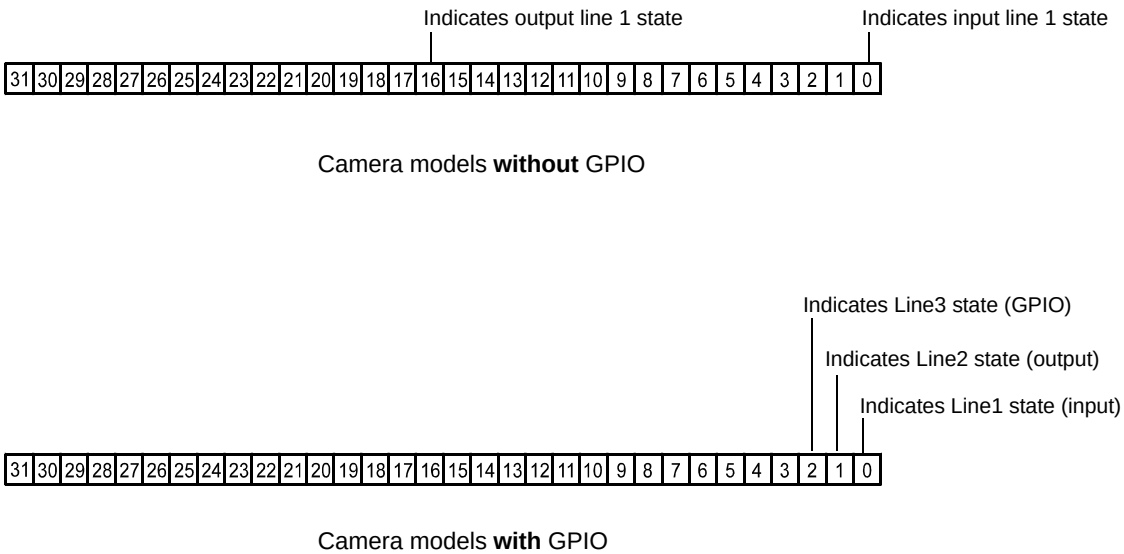


Fig. 57: Bit Field of the SyncUserOutputValueAll Parameter: Bit Numbers and Assignment of the Output Line

To determine all current UserOutputValue parameter values in a single step, check the hexadecimal number of the UserOutputValueAll parameter value. This contains the current state of all user settable output signals.

For example, if a UserOutputValueAll parameter value of 0x1 is reported while all line inverters are disabled, this can be translated into the following states:

- the UserOutputValue parameter value for Line 1 is currently 1, indicating that the signal state is currently high and
- the UserOutputValue parameter value for the GPIO line is currently 0, indicating that the signal line state is currently low.

Setting and Checking the State Using Basler pylon

To set the state of all user settable output signals in a single step:

1. Set the value of the UserOutputValueAll parameter to set all output signals.

To check the state of all user settable output signals in a single step:

1. Read the value of the UserOutputValueAll parameter to determine the current settings of all output signals.

You can set and read the UserOutputValueAll parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set and read the parameter value. In this example, the UserOutputValueAll parameter value is set to 0.

```
// Setting all output signal values in a single step
camera.UserOutputValueAll.SetValue(0);
// Reading all output signal values in a single step
int64_t i = camera.UserOutputValueAll.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 63.

5.11.5 Setting the State of a User Settable Synchronous Output Signal

Available for	Not Available for
acA640-90, acA640-120, acA645-100, acA750-30, acA780-75, acA1280-60, acA1300-22, acA1300-30, acA1300-60, acA1600-20, acA1600-60, acA1920-25, acA2000-50, acA2040-25, acA2500-14, acA3800-10, acA4600-7	acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

User settable output lines, in this case the camera's output line, can be set to supply output signals synchronous to the frame start trigger. This works in a similar fashion to the User Output Value parameter described in Section 5.11.3 on page 106 with the difference that this parameter value can be changed with the sequencer feature.

Using the synchronous output signal, you can control external events and devices, e.g. a light source, when you're using the sequencer feature. To do this, you save the state of an output line, high or low, in a sequence set, so when a frame start trigger is received and the sequencer advances from one set to the next, the output signal is either high or low. This can be used to turn a lamp on or off to account for different lighting requirements depending on the sequence set for example. Put in other words, the turning on or off of a lamp is synchronized with the frame start trigger of the sequences.

For more information about the sequencer feature, see Section 9.8 on page 273.

Setting the State Using Basler pylon

To set the state of a synchronous output signal:

For Models without GPIO
1. Use the SyncUserOutputSelector parameter to select output line 1. 2. Set the SyncUserOutputValue parameter to true (1) or false (0). This will set the state of the output line.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
// Set the output line to provide a synchronous user settable output
// signal
Camera.LineSelector.SetValue(LineSelector_Out1);
Camera.LineSource.SetValue(LineSource_SyncUserOutput);
// Select the user settable line and set the state of the synchronous
// user output signal
camera.SyncUserOutputSelector.SetValue(
   SyncUserOutputSelector_SyncUserOutput1);
camera.SyncUserOutputValue.SetValue(true);
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
Camera.LineSource.SetValue(LineSource_SyncUserOutput);
// Select the user settable line and set the state of the synchronous
// user output signal
camera.SyncUserOutputSelector.SetValue(
   SyncUserOutputSelector_SyncUserOutput1);
camera.SyncUserOutputValue.SetValue(true);
```

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 63.

5.11.6 Setting and Checking the State of All User Settable Synchronous Output Signals

The SyncUserOutputValueAll parameter reports the current state of all user settable synchronous output signals. The parameter value is expressed as a hexadecimal number by the Basler pylon Viewer and as a 32-bit word in the Basler pylon API. The number is derived from bits 0 and 12 of a 32-bit word and is based on all current binary SyncUserOutputValue parameter values.

As shown in Figure 58, certain bits are associated with certain lines. The states of those lines are expressed by the related binary SyncUserOutputValue parameter values.

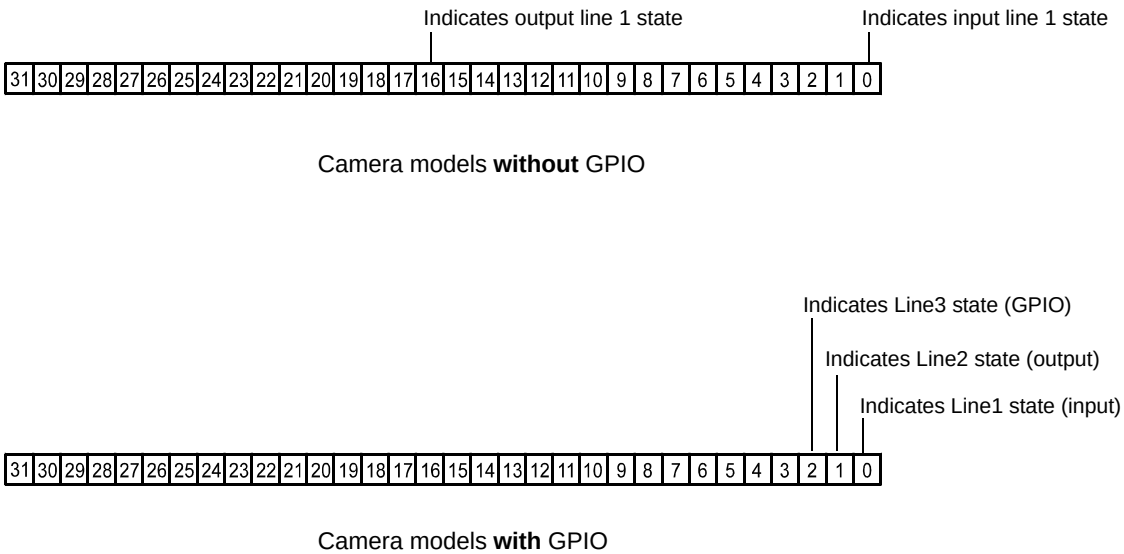


Fig. 58: Bit Field of the SyncUserOutputValueAll Parameter: Bit Numbers and Assignment of the Output Line

To determine all current SyncUserOutputValue parameter values in a single step, check the hexadecimal number of the SyncUserOutputValueAll parameter value. This contains the current state of all user settable synchronous output signals.

For example, if a SyncUserOutputValueAll parameter value of 0x1 is reported while all line inverters are disabled, this can be translated into the following states:

- the SyncUserOutputValue parameter value for Line 1 is currently 1, indicating that the signal state is currently high and
- the SyncUserOutputValue parameter value for the GPIO line is currently 0, indicating that the signal line state is currently low.

Setting and Checking the State Using Basler pylon

To set the state of all user settable synchronous output signals in a single step:

1. Set the value of the SyncUserOutputValueAll parameter to set all synchronous output signals.

To check the state of all user settable synchronous output signals in a single step:

1. Read the value of the SyncUserOutputValueAll parameter to determine the current settings of all synchronous output signals.

You can set and read the SyncUserOutputValueAll parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set and read the parameter value. In this example, the SyncUserOutputValueAll parameter value is set to 0.

```
// Setting all synchronous output signal values in a single step
camera.SyncUserOutputValueAll.SetValue(0);
// Reading all synchronous output signal values in a single step
int64_t i = camera.SyncUserOutputValueAll.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 63.

5.11.7 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 (also see Figure 59):

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

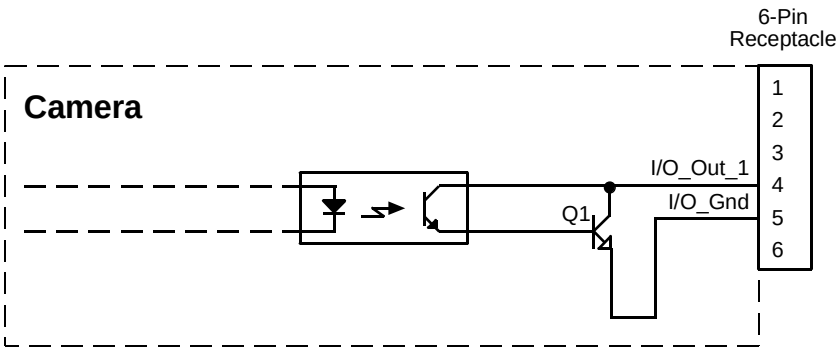


Fig. 59: Output Line Schematic (Simplified)

Setting the State Using Basler pylon

To set the invert function on the output line:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Set the LineInverter parameter to true to enable inversion on the selected line or to false to disable inversion.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Set the LineInverter parameter to true to enable inversion on the selected line or to false to disable inversion.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

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

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
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 63.

5.11.8 Working with the Timer Output Signal

The source signal for the output line can be set to TimerActive (camera models with GPIO: TimerActive1). The camera has one timer designated as "timer 1". When you set the source signal for the output line to TimerActive (cameras with GPIO: TimerActive1), 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.

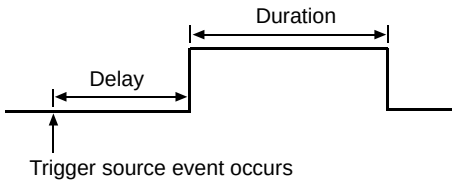


Fig. 60: Timer Signal

Currently, the only trigger source event available to start the timer is ExposureActive. The event is generated on the rising edge of the 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, 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 5.11.1 on page 102.

5.11.8.1 Setting the Trigger Source for the Timer

To set the trigger source for the timer:

1. Use the TimerSelector parameter to select timer 1.
2. Set the value of the TimerTriggerSource parameter to ExposureActive. This will set the selected timer to use the start of exposure to begin the timer.

You can set the TriggerSelector and the TimerTriggerSource 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:

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
Camera.LineSource.SetValue (LineSource_TimerActive);
Camera.TimerSelector.SetValue(TimerSelector_Timer1);
Camera.TimerTriggerSource.SetValue(TimerTriggerSource_ExposureStart);
```

For Models with GPIO

```
Camera.LineSource.SetValue (LineSource_Timer1Active);
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
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 63.

5.11.8.2 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 TimerDelayRaw parameter, and the second element is the TimerDelayTimeBase. The delay time is the product of these two elements:

Delay time = (TimerDelayRaw value) x (TimerDelayTimebaseAbs value)

By default, the TimerDelayTimebaseAbs is fixed at 1 µs. Typically, the delay time is adjusted by setting the TimerDelayRaw parameter value.

The TimerDelayRaw 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:

1. Use the TimerSelector parameter to select Timer1.
2. Set the value of the TimerDelayRaw parameter.

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 TimerDelayTimebaseAbs is fixed at 1 µs (minimum value), and the timer delay is normally adjusted by setting the value of the TimerDelayRaw parameter. However, if you require a delay time that is longer than what you can achieve by changing the value of the TimerDelayRaw parameter alone, the TimerDelayTimebaseAbs parameter can be used to change the delay time base.

The TimerDelayTimebaseAbs 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 TimerDelayTimebaseAbs 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.

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 TimerDelayAbs 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:

1. Use the TimerSelector to select timer 1.
2. Set the value of the TimerDelayAbs parameter.

You can set the TimerSelector and the TimerDelayAbs 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 TimerDelayAbs parameter to set the delay time, the camera accomplishes the setting change by automatically changing the TimerDelayRaw parameter to achieve the value specified by the TimerDelayAbs setting.
This leads to the following limitation: You must set the TimerDelayAbs parameter to a value that is equivalent to a setting you could achieve by using the TimerDelayRaw and the current TimerDelayTimebaseAbs parameters.
For example, if the time base was currently set to 50 µs, you could use the TimerDelayAbs parameter to set the delay to 50 µs, 100 µs, 150 µs, etc.

Note that, if you set the TimerDelayAbs parameter to a value that you could not achieve by using the TimerDelayRaw and current TimerDelayTimebaseAbs parameters, the camera will

automatically change the setting for the TimerDelayAbs parameter to the nearest achieveable value.

You should also be aware that, if you change the delay time using the raw settings, the TimerDelayAbs 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 63.

5.11.8.3 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: TimerDurationRaw parameter, and TimerDurationTimebaseAbs. The duration time is the product of the following two elements:

Duration Time = (TimerDurationRaw parameter value) x (TimerDurationTimebaseAbs)

By default, the TimerDurationTimebaseAbs is fixed at 1 µs. Typically, the duration time is adjusted by setting only the TimerDurationRaw parameter value.

Range of TimerDurationRaw 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 TimerSelector to select a timer.
2. Set the value of the TimerDurationRaw parameter.

You can set the TimerSelector and the TimerDelayRaw 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);
```

Changing the Duration Time Base

By default, the TimerDurationTimebaseAbs parameter is fixed at 1 µs, and the timer duration is normally adjusted by setting the value of the TimerDelayRaw parameter. However, if you require a duration time that is longer than what you can achieve by changing the value of the TimerDurationRaw parameter alone, the TimerDurationTimebaseAbs parameter can be used to change the duration time base.

The TimerDurationTimebaseAbs 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 TimerDurationTimebaseAbs 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.

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 TimerDurationAbs 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 TimerSelector to select timer 1.
2. Set the value of the TimerDurationAbs parameter.

You can set the TimerSelector and the TimerDurationAbs 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(100.0);
```

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

When you use the TimerDurationAbs parameter to set the duration time, the camera accomplishes the setting change by automatically changing the TimerDurationRaw parameter to achieve the value specified by the TimerDurationAbs setting. This leads to a limitation that you must keep in mind, if you use TimerDurationAbs parameter to set the duration time. That is, you must set the TimerDurationAbs parameter to a value that is equivalent to a setting you could achieve by using the TimerDurationRaw and the current TimerDurationTimebaseAbs parameters. For example, if the time base was currently set to 50 µs, you could use the TimerDurationAbs parameter to set the duration to 50 µs, 100 µs, 150 µs, etc.

If you read the current value of the TimerDurationAbs parameter, the value will indicate the product of the TimerDurationRaw parameter and the TimerDurationTimebaseAbs. In other words, the TimerDurationAbs 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 TimerDurationAbs 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 63.

5.12 Checking the State of the I/O Lines

5.12.1 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:

For Models without GPIO	For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Read the value of the LineStatus 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.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Read the value of the LineStatus parameter to determine the current state of the line.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

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

For Models with GPIO

```
// Select output line 1 and read the state
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
bool Line2State = 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 63.

5.12.2 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:

1. Read the value of the LineStatusAll parameter.

You can read the LineStatusAll 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 LineStatusAll parameter is a 32-bit value. As shown in Figure 61 and Figure 62, 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.

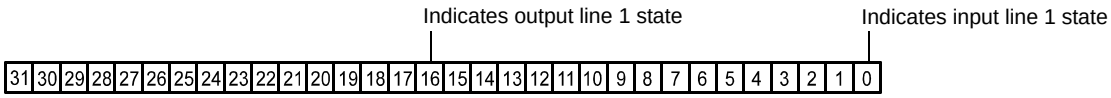


Fig. 61: Line Status All Parameter Bits (Camera models **without** GPIO)

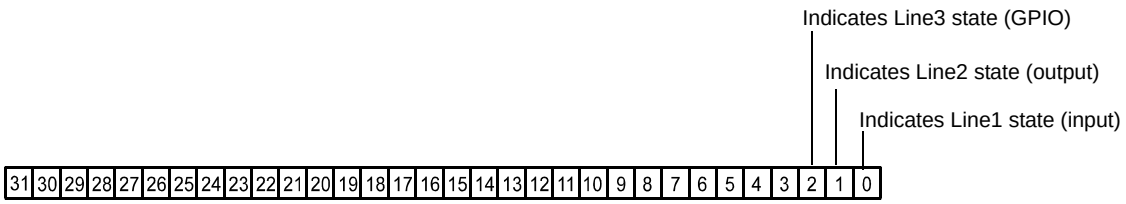


Fig. 62: Line Status All parameter bits (Camera models **with** GPIO)

6 Image Acquisition Control

	When configuring the image acquisition control parameters, keep in mind that for some camera models a GPIO line is available in addition to the input and output line. The configuration of the I/O lines for cameras with GPIO is different compared to the configuration of cameras without GPIO. For information about the availability of a GPIO line in the different camera models, see Table 16 on page 74.

	The sample code included in this section represents "low level" code that is actually used by the camera. Many tasks, however, can be programmed more conveniently with fewer lines of code when employing the Instant Camera classes, provided by the Basler pylon C++ API. For information about the Instant Camera classes, see the *C++ Programmer's Guide and Reference Documentation* delivered with the Basler pylon Camera Software Suite.

This chapter provides information about controlling image acquisition. You will find information about

- triggering image acquisition,
- setting the exposure time for acquired images,
- controlling the camera's image acquisition rate, and
- how the camera's maximum allowed image acquisition rate can vary depending on the current camera settings.

6.1 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:

- AcquisitionStart and AcquisitionStop commands and the AcquisitionMode 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 63 on page 124 and to the use case diagrams in Section 6.11 on page 192. 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.

AcquisitionStart and AcquisitionStop Commands and the AcquisitionMode Parameter

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

A parameter called the AcquisitionMode has a direct bearing on how the AcquisitionStart command operates.

If the AcquisitionMode parameter is set to

- **SingleFrame**, you can only acquire one frame after executing an AcquisitionStart command. When one frame has been acquired, the AcquisitionStart command will expire. Before attempting to acquire another frame, you must execute a new AcquisitionStart command.
- **Continuous**, an AcquisitionStart command does not expire after a single frame is captured. Once an AcquisitionStart command has been executed, you can acquire as many frames as you like. The AcquisitionStart command will remain in effect until you execute an AcquisitionStop command. Once an AcquisitionStop command has been executed, the camera will not be able to acquire frames until a new AcquisitionStart command is executed.

The Trigger Selector

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 TriggerMode parameter, that is used to set the mode for both of these triggers. Also, the TriggerSoftware command can be executed for either the acquisition start trigger or the frame start trigger.

Whenever you want to work with a specific type of trigger, your first step is to set the TriggerSelector parameter to the trigger you want to work with; either AcquisitionStart or the FrameStart. At that

point, the changes you make to the TriggerMode, TriggerSource, etc., will be applied to the selected trigger only.

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 **TriggerMode** 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.
- **On**, the initial acquisition status of the camera will be "waiting for acquisition start trigger" (see Figure 63 on page 124).
  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 AcquisitionFrameCount. 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 TriggerMode parameter is set to On, the AcquisitionFrameCount parameter is set to 3, 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 63 on page 124). 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 **TriggerMode** parameter for the **frame start trigger** is

- **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.
- **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.

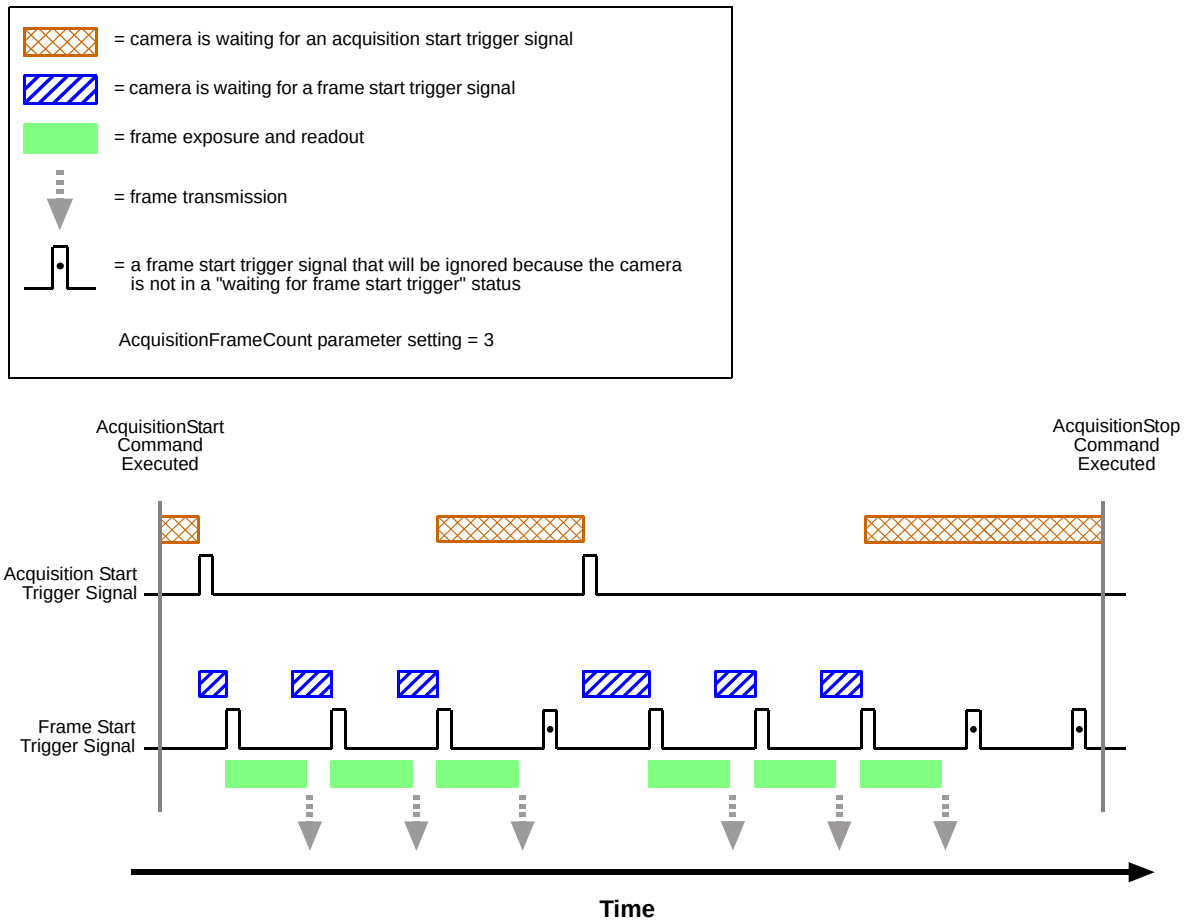


Fig. 63: Acquisition Start and Frame Start Triggering

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 TriggerSoftware command is executed, the selected trigger signal will be applied to the camera.
- via **hardware**, you must first select the acquisition start or the frame start trigger and indicate that input line 1 (or if input line 3 is used as an input on cameras with GPIO: Line3) 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 (or input line 3), an occurrence of the selected trigger signal will be recognized by the camera.

Exposure Time Control

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 ExposureTimeAbs will determine the exposure time for each frame.
- **hardware frame start triggering**, there are two modes of operation: "Timed" and "TriggerWidth".
  - With the "**Timed**" mode, the ExposureTimeAbs parameter will determine the exposure time for each frame.
  - With the "**TriggerWidth**" mode, the way that you manipulate the rise and fall of the hardware signal will determine the exposure time. The "TriggerWidth" mode is especially useful,if you want to change the exposure time from frame to frame.

6.2 AcquisitionStart and AcquisitionStop Commands and the AcquisitionMode

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

Executing an AcquisitionStop command terminates the camera's ability to acquire frames. When the camera receives an AcquisitionStop 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 AcquisitionMode parameter has two settings: SingleFrame and Continuous. The use of AcquisitionStart and AcquisitionStop commands and the camera's AcquisitionMode parameter setting are related.

If the camera's Acquisition Mode parameter is set to

- **SingleFrame**, after an AcquisitionStart command has been executed, a single frame can be acquired. When acquisition of one frame is complete, the camera will execute an AcquisitionStop command internally and will no longer be able to acquire frames. To acquire another frame, you must execute a new AcquisitionStart command.
- **ContinuousFrame**, after an AcquisitionStart 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 AcquisitionStop command is executed. Once the AcquisitionStop command is received, the camera will no longer be able to acquire frames.

When the camera's acquisition mode is set to SingleFrame, 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 6.11 on page 192.

Setting the Acquisition Mode and Issuing Start/Stop Commands

You can set the AcquisitionMode parameter value and you can execute AcquisitionStart or AcquisitionStop commands from within your application software by using the Basler pylon API. The code snippet below illustrates using the API to set the AcquisitionMode parameter value and to execute an AcquisitionStart 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 63.

6.3 The Acquisition Start Trigger

When reading this section, it is helpful to refer to Figure 63 on page 124.

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 TriggerMode parameter for the acquisition start trigger is set to on. This feature is explained in greater detail in the following sections.

6.3.1 Acquisition Start Trigger Mode

The main parameter associated with the acquisition start trigger is the TriggerMode parameter. The TriggerMode parameter for the acquisition start trigger has two available settings: Off and On.

6.3.1.1 Acquisition Start Trigger Mode = Off

When the TriggerMode 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.

6.3.1.2 Acquisition Start Trigger Mode = On

When the TriggerMode 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

AcquisitionFrameCount 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 TriggerMode parameter for the acquisition start trigger is set to On, you must select a source signal to serve as the acquisition start trigger. The TriggerSource parameter specifies the source signal. The available selections for the TriggerSource parameter are:

- **Software** - When the source signal is set to Software, you apply an acquisition start trigger signal to the camera by executing a TriggerSoftware command for the acquisition start trigger on the host PC.
- **Line1** - 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.
- **Line3** - Analogous to line 1. The GPIO line line 3 must be configured for input.

If the TriggerSource parameter for the acquisition start trigger is set to Line1 or Line3, you must also set the TriggerActivation parameter. The available settings for the TriggerActivation parameter are:

- **RisingEdge** - specifies that a rising edge of the electrical signal will act as the acquisition start trigger.
- **FallingEdge** - specifies that a falling edge of the electrical signal will act as the acquisition start trigger.

When the TriggerMode parameter for the acquisition start trigger is set to On, the camera’s AcquisitionMode parameter must be set to Continuous.

6.3.2 Acquisition Frame Count

When the TriggerMode parameter for the acquisition start trigger is set to On, you must set the value of the camera’s AcquisitionFrameCount parameter. The value of the AcquisitionFrameCount 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 AcquisitionFrameCount 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.

6.3.3 Setting the Acquisition Start Trigger Mode and Related Parameters

You can set the TriggerMode and TriggerSource parameters for the acquisition start trigger and also set the AcquisitionFrameCount parameter value from within your application software by using the Basler pylon API.

The following code snippet illustrates using the API to set the TriggerMode to On, the TriggerSource to Software, and the AcquisitionFrameCount 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 TriggerMode to On, the TriggerSource to Line1, the TriggerActivation to RisingEdge, and the AcquisitionFrameCount 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 63.

6.3.4 Using a Software Acquisition Start Trigger

6.3.4.1 Introduction

If the TriggerMode parameter for the acquisition start trigger is set to On and the TriggerSource 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 TriggerSelector parameter to AcquisitionStart.
- Executing a TriggerSoftware 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 AcquisitionFrameCount 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 TriggerSelector parameter is set to AcquisitionStart, a software acquisition start trigger will be applied to the camera each time a TriggerSoftware command is executed.

6.3.4.2 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 63.

6.3.5 Using a Hardware Acquisition Start Trigger

6.3.5.1 Introduction

If the TriggerMode parameter for the acquisition start trigger is set to On and the TriggerSource parameter is set to Line1 or Line3 (if the GPIO line is configured as an input), an externally generated electrical signal injected into the input line 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 TriggerActivation parameter is used to select rising edge or falling edge triggering.

When the TriggerMode 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 the selected input line (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 AcquisitionFrameCount parameter setting, the camera will return to the "waiting for acquisition start trigger" acquisition status. When a new ExASTrig signal is applied to the input line, 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 6.3.5.2.
- the electrical requirements for the input line(s), see Section 5.6 on page 80.
- which camera model has GPIO, see Section 5.2 on page 74.

6.3.5.2 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 63.

6.4 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 AcquisitionMode parameter must be set correctly.
- A proper AcquisitionStart command must be applied to the camera.
- A proper acquisition start trigger signal must be applied to the camera (if the TriggerMode parameter for the acquisition start trigger is set to On).

For more information about

- the AcquisitionMode parameter and about AcquisitionStart and AcquisitionStop commands, see Section 6.1 on page 122 and Section 6.2 on page 126.
- the acquisition start trigger, and about the acquisition status, see Section 6.1 on page 122 and Section 6.3 on page 128.

Referring to the use case diagrams that appear in Section 6.11 on page 192 can help you understand the explanations of the frame start trigger.

6.4.1 Trigger Mode

The main parameter associated with the frame start trigger is the TriggerMode parameter. The TriggerMode parameter for the frame start trigger has two available settings: Off and On.

6.4.1.1 Frame Start Trigger Mode = Off (Free Run)

When the TriggerMode parameter for the frame start 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 TriggerMode set to Off, the way that the camera will operate the frame start trigger depends on the setting of the camera's AcquisitionMode parameter. If the AcquisitionMode parameter is set to

- **SingleFrame**, the camera will automatically generate a single frame start trigger signal whenever it receives an AcquisitionStart command.
- **Continuous**, the camera will automatically begin generating frame start trigger signals when it receives an AcquisitionStart command. The camera will continue to generate frame start trigger signals until it receives an AcquisitionStop command.

  The rate at which the frame start trigger signals are generated may be determined by the camera's AcquisitionFrameRateAbs 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.

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 6.1 on page 122 and Section 6.3 on page 128.

Exposure Time Control with the TriggerMode Set to Off

When the TriggerMode 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 ExposureTimeAbs parameter.

For more information about the camera's ExposureTimeAbs parameter, see Section 6.5 on page 150.

6.4.1.2 TriggerMode = On (Software or Hardware Triggering)

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

Do not trigger frame acquisition at a rate that exceeds

- the maximum allowed for the current camera settings. If you apply frame start trigger signals to the camera when it is not ready to receive them, the signals will be ignored.
  For more information about determining the maximum allowed frame rate, see Section 6.12 on page 196.
- the host computer's capacity limits for data transfer or storage or both. If you try to acquire more images than the host computer is able to process, frames may be dropped. For more information about bandwidth optimization, see the *Installation and Setup Guide for Cameras Used with Basler pylon for Windows* (AW000611).

The TriggerSource parameter specifies the source signal that will act as the frame start trigger signal. The available selections for the TriggerSource 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.
- **Line1** - 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.
- **Line3** - Analogous to line 1. The GPIO line line 3 must be configured for input.

If the TriggerSource parameter is set to Line1 or Line3, you must also set the TriggerActivation parameter. The available settings for the TriggerActivation parameter are:

- **RisingEdge** - specifies that a rising edge of the electrical signal will act as the frame start trigger.
- **FallingEdge** - 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 6.4.2 on page 139.
- using a hardware trigger to control frame acquisition start, see Section 6.4.3 on page 142.

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

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 6.1 on page 122 and Section 6.3 on page 128.

Exposure Time Control with the TriggerMode Set to On

When the TriggerMode parameter for the frame start trigger is set to On and the TriggerSource parameter is set to

- **Software**, the exposure time for each frame acquisition is determined by the value of the camera's ExposureTimeAbs parameter.
- **Line1** or **Line3**, the exposure time for each frame acquisition can be controlled
  - with the ExposureTimeAbs 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 6.4.2 on page 139.
- when using a hardware trigger, see Section 6.4.3 on page 142.

6.4.1.3 Setting The Frame Start Trigger Mode and Related Parameters

You can set the TriggerMode 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 TriggerMode for the frame start trigger to On and the TriggerSource to Line1:

```
// 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 AcquisitionMode to Continuous, the TriggerMode 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( );
```

6.4.2 Using a Software Frame Start Trigger

6.4.2.1 Introduction

If the TriggerMode parameter for the frame start trigger is set to On and the TriggerSource 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 64 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 ExposureMode parameter must be set to Timed. The exposure time for each acquired frame will be determined by the value of the camera's ExposureTimeAbs parameter.

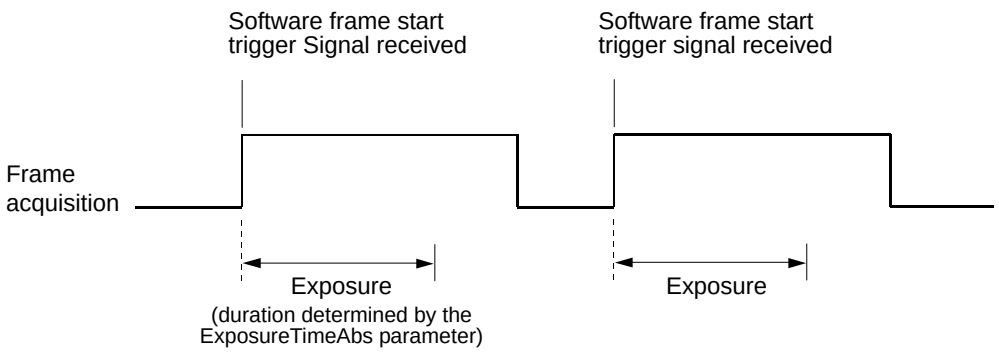


Fig. 64: Frame Acquisition with a Software Frame Start Trigger

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. Software frame start trigger signals that are applied to the camera when it is not ready to receive them will be ignored.
There is a detailed explanation about the maximum allowed frame rate at the end of this chapter.

Section 6.4.2.2 on page 140 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 6.12 on page 196.

6.4.2.2 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 TriggerMode 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 63.

6.4.3 Using a Hardware Frame Start Trigger

6.4.3.1 Introduction

If the TriggerMode parameter for the frame start trigger is set to On and the TriggerSource parameter is set to Line1 or Line3, an externally generated electrical signal injected into physical input line 1 or into GPIO line line 3 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 TriggerActivation 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 6.4.3.4 on page 148.
- the electrical requirements for line 1, see Section 5.6 on page 80.
- determining the maximum allowed frame rate, see Section 6.12 on page 196.

6.4.3.2 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 exposure mode and
- trigger width exposure mode.

Depending on the camera models there are differences in the exposure modes.

Timed Exposure Mode

When the timed mode is selected, the exposure time for each frame acquisition is determined by the value of the camera's ExposureTimeAbs parameter.

If the camera is set for

- **rising edge** triggering, the exposure time starts when the ExFSTrig signal rises.
- **falling edge** triggering, the exposure time starts when the ExFSTrig signal falls.

Figure 65 illustrates timed exposure with the camera set for rising edge triggering.

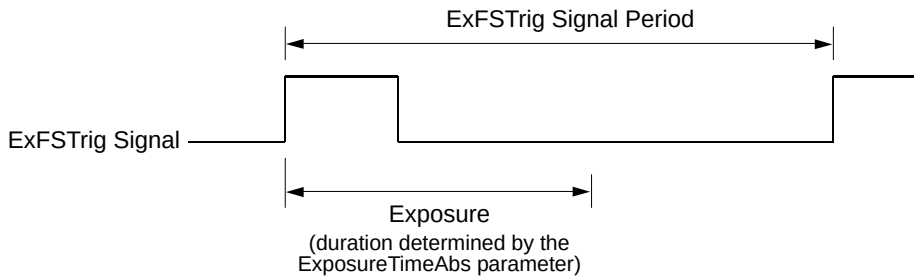


Fig. 65: Timed Exposure with Rising Edge Triggering

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 66 for rising edge triggering.

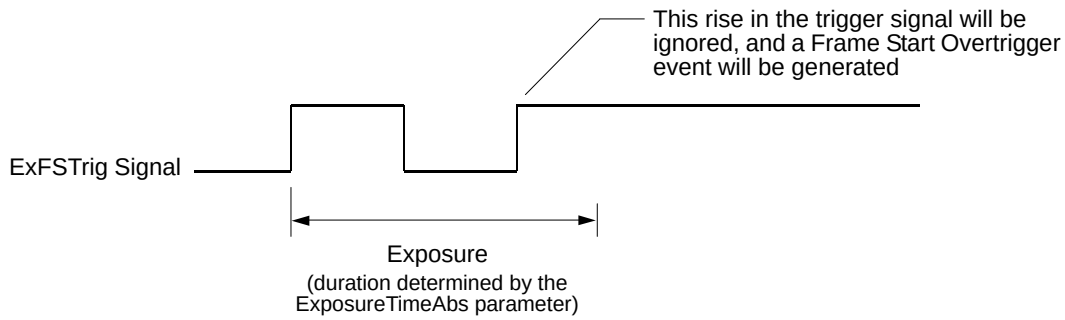


Fig. 66: Overtriggering with Timed Exposure

For more information about

- the Frame Start Overtrigger event, see Section 9.16 on page 350.
- the camera's ExposureTimeAbs parameter, see Section 6.5 on page 150.

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure for each frame acquisition will be directly controlled by the ExFSTrig signal.
Depending on the camera model, there are differences how the trigger width exposure mode is configured.

Trigger Width Exposure Mode without Offset	Trigger Width Exposure Mode with Offset	No Trigger Width Exposure Mode Available
acA640-90*, acA640-120*, acA645-100*, acA750-30*, acA780-75*, acA1300-22*, acA1300-30*, acA1600-20* For information, see page 144.	acA640-300 **, acA800-200 **, acA1300-75 **, acA1920-40 **, acA1920-50 **, acA2000-50 **, acA2040-25 ** For information, see page 145.	acA1280-60, acA1300-60, acA1600-60, acA1920-25, acA2500-14, acA3800-10, acA4600-7

Trigger Width Exposure Mode (without Exposure Time Offset)

For the camera models marked with an Asterik * in the table above the trigger width exposure is realized as follows:

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 67 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.

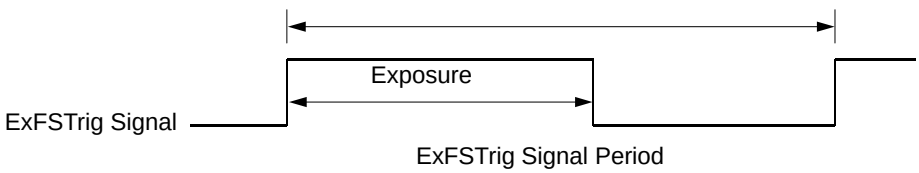


Fig. 67: Trigger Width Exposure with Rising Edge Triggering

When you operate the camera in trigger width exposure mode, you must also set the camera's ExposureOverlapTimeMaxAbs parameter. This parameter setting will be used by the camera to operate the FrameTriggerWait signal.

You should set the ExposureOverlapTimeMaxAbs 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 ExposureOverlapTimeMaxAbs parameter to 3000 µs.

Trigger Width Exposure Mode with Special Exposure Time Offset

For the camera models marked with two Asteriks ** in the table on page 144, a special exposure time offset must be taken into account.

When the trigger width exposure mode is selected, the exposure time for each frame acquisition will be the sum of two individual time periods (see Figure 68):

- The first time period is the exposure time that is controlled by the ExFSTrig signal:
  If the camera is set for rising edge triggering, the first time period - and therewith the exposure time - begins when the ExFSTrig signal rises. The first time period ends when the ExFSTrig signal falls.

  If the camera is set for falling edge triggering, the first time period begins when the ExFSTrig signal falls. The first time period ends when the ExFSTrig signal rises.
- The second time period is the exposure time offset, C_4. It is automatically added to the first time period by the camera's sensor. See Table 24 on page 145.

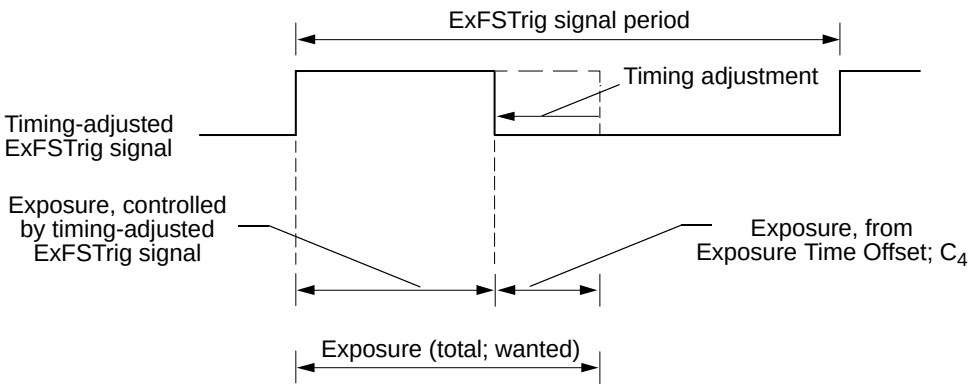


Fig. 68: Trigger Width Exposure with Adjusted Rising Edge Triggering; (Exposure Start Delays Is Omitted)

Camera Models	Exposure Time Offset C_4
acA2000-50, acA2040-25	37 µs
acA640-300*, acA800-200*, acA1300-75 *	64 µs
acA1920-40*, acA1920-50 *	14 µs

Table 24: Exposure Time Offset Values

Camera Models	Exposure Time Offset C_4
* These models have an additional ExposureOverlapTimeMode parameter that can be set to Automatic or Manual; see below.	

Table 24: Exposure Time Offset Values

To obtain a certain wanted exposure time with trigger width exposure mode you will have to adjust the ExFSTrig signal in order to compensate for the automatically added exposure time offset, C_4: Subtract C_4 from the wanted exposure time. Use the resulting adjusted time as the high time for the ExFSTrig signal if the signal is not inverted or as the low time if the signal is inverted.
Note that the C_4 exposure time does **not** affect the moment of exposure start.

When you operate the camera in trigger width exposure mode, you must also set the following parameters:

- ExposureOverlapTimeMode parameter:
  This parameter is **only available** for cameras with a GPIO line (see * in the table above). The parameter can be set to Manual or Automatic. If the ExposureOverlapTimeMode parameter is set to
  - Automatic, the value of the ExposureOverlapTimeMaxAbs parameter is automatically set to the maximum possible value. In this case you cannot modify the ExposureOverlapTimeMaxAbs parameter.
  - Manual, you can adapt the ExposureOverlapTimeMaxAbs parameter to your requirements.
- ExposureOverlapTimeMaxAbs parameter.

The parameter setting will be used by the camera to operate the FrameTriggerWait signal.

You should set the ExposureOverlapTimeMaxAbs 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 ExposureOverlapTimeMaxAbs parameter to 3000 µs.

For more information about

- the FrameTriggerWait signal and the ExposureOverlapTimeMaxAbs parameter, see Section 6.10.4 on page 183.
- which camera model has a GPIO line, see Section 5.2 on page 74.

Setting the Parameters Related to the Trigger Width Exposure Mode

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

If the camera is a camera with GPIO line it is important to set the exposure mode **after** the trigger mode and the trigger source have been set. Otherwise the ExposureMode_TriggerWidth parameter is not available.

```
// Set the trigger selector to frame start.
Camera.TriggerSelector.SetValue(TriggerSelector_FrameStart);

// Set the trigger mode.
Camera.TriggerMode.SetValue(TriggerMode_On);
// Set the trigger source.
Camera.TriggerSource.SetValue(TriggerSource_Line1);

// Set the exposure mode.
Camera.ExposureMode.SetValue(ExposureMode_TriggerWidth);
```

For information about which camera model has a GPIO line or not, see Section 5.2 on page 74.

6.4.3.3 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:

1. Set the camera's TriggerSelector parameter to FrameStart.
2. Set the value of the TriggerDelayAbs parameter.

The frame start trigger delay will not operate, if the TriggerMode parameter for the frame start trigger is set to Off or if you are using a software frame start trigger.

6.4.3.4 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 TriggerMode 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 TriggerMode 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);
// If the camera model is a camera with GPIO line:
// Set the exposure overlap time mode
Camera.ExposureOverlapTimeMode.SetValue(ExposureOverlapTimeMode_Manual);
// 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 63.

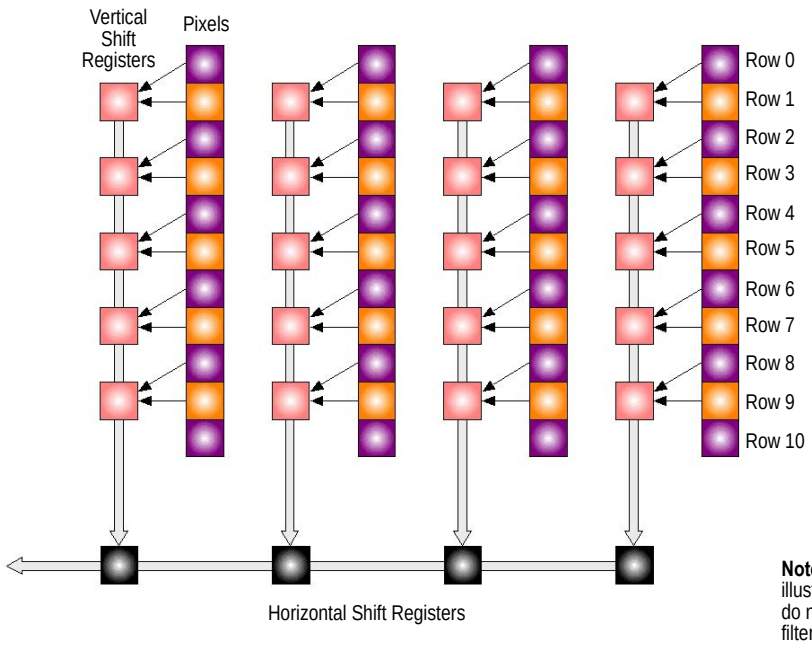
6.5 acA750 - Acquisition Control Differences

6.5.1 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 69, 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.



Note: The colors used in this drawing are designed to illustrate how the camera's output modes work. They do not represent the actual colors used in the color filter on acA750-30gc cameras.

Fig. 69: Field 0 Readout

As shown in Figure 70, 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.

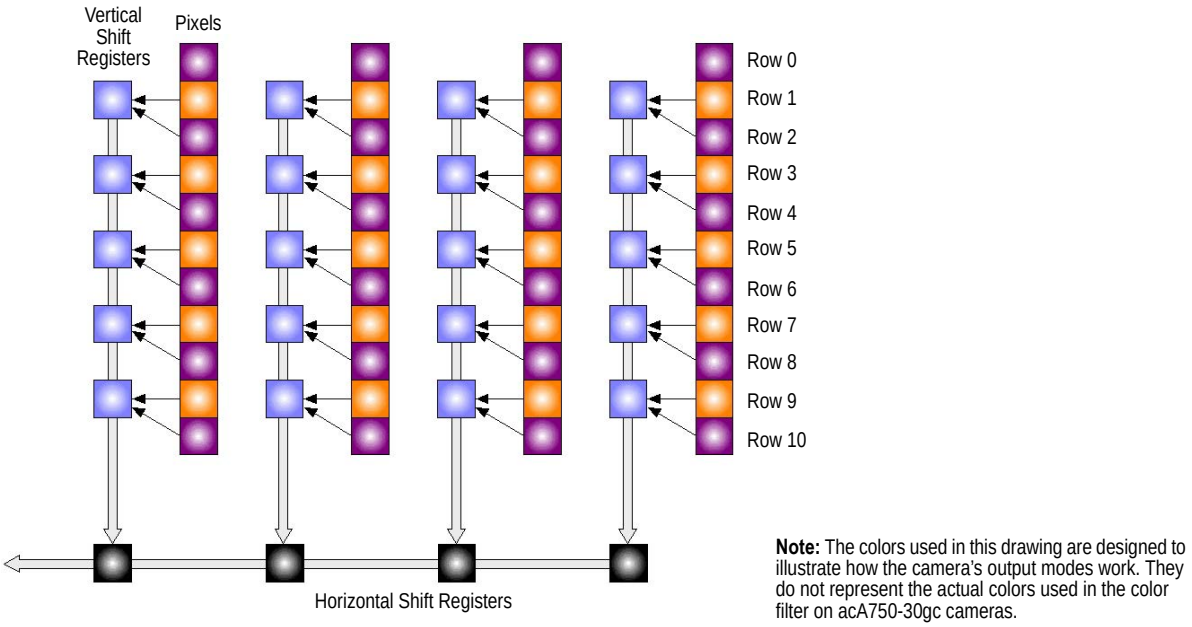


Fig. 70: Field 1 Readout

6.5.2 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 6.5.1 on page 150. 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.

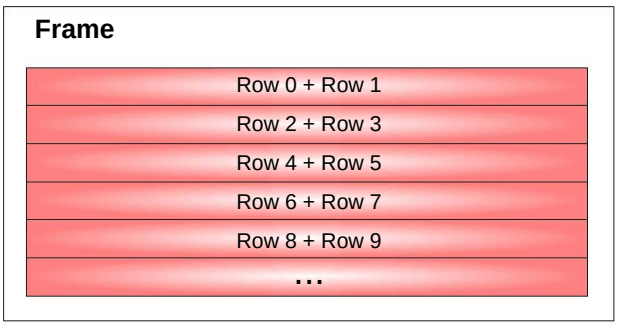


Fig. 71: Field 0 Output Mode

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 6.5.1 on page 150. 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.

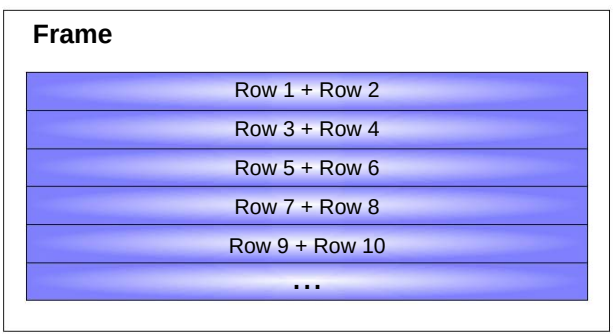


Fig. 72: Field 1 Output Mode

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 73, 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.

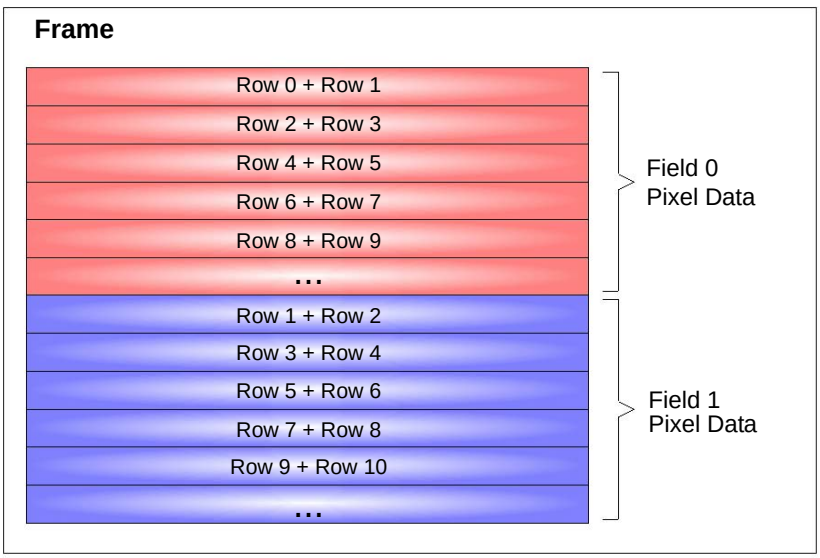


Fig. 73: Concatenated New Fields Output Mode

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 74, 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.

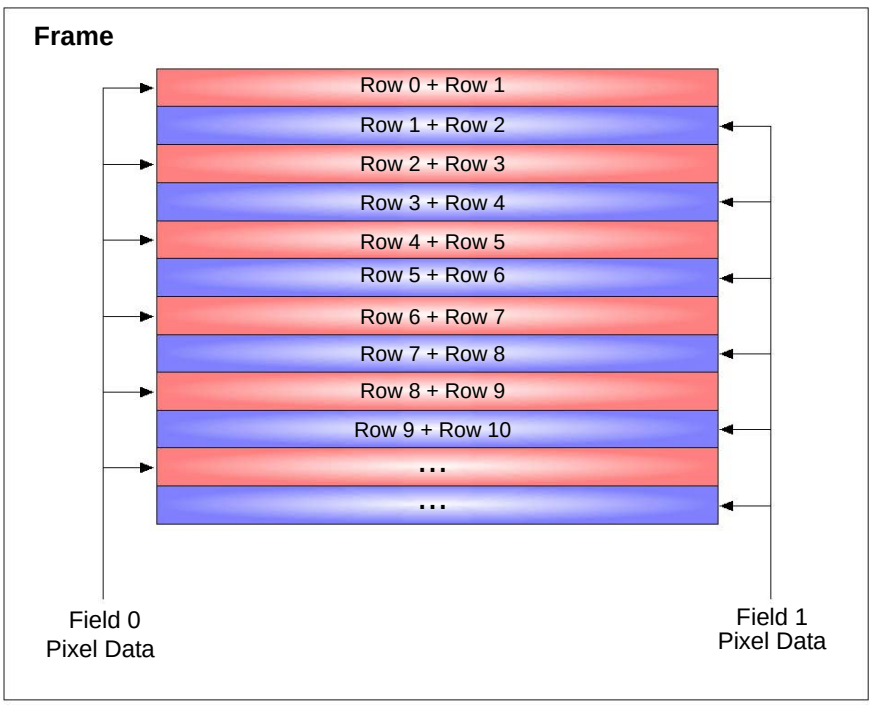


Fig. 74: Deinterlaced New Fields Output Mode

6.5.3 Setting the Field Output Mode

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

```
// 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 63.

6.6 Setting the Exposure Time

This section (Section 6.6) 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 9.14.1 on page 328.
- the Exposure Auto function in particular, see Section 9.14.5 on page 337.

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

- TriggerMode parameter for the frame start trigger is set to Off.
- TriggerMode parameter for the frame start trigger is set to On and TriggerSource is set to Software.
- TriggerMode parameter for the frame start trigger is set to On, TriggerSource is set to Line1 or Line3, and the ExposureMode is set to Timed.

The ExposureTimeAbs parameter must not be set below a minimum specified value. The minimum and maximum settings for each camera model are shown in the following tables.
As some cameras can be operated either with global shutter or with rolling shutter the possible ExposureTime parameters depend on the selected shutter mode.

Table 25 on page 157 shows the values for cameras operated with global shutter.

Table 26 on page 158 shows the values for cameras operated with rolling shutter.

Global Shutter Operation: Exposure Times [µs]			
Camera Model	**Minimum Allowed Exposure Time**	**Maximum Possible Exposure Time**	**Can be set in increments of ...**
acA640-90gm/gc	17	1000000	1
acA640-120gm/gc	4	1000000	
acA645-100gm/gc	20	10000000	
acA640-300gm/gc**	112	10000000	
acA750-30gm/gc	30	1000000	
acA780-75gm/gc	20	10000000	
acA800-200gm/gc**	112	10000000	
acA1300-22gm/gc, acA1300-30gm/gc	16	10000000	
acA1300-60gm/gc (*), acA1300-60gmNIR (*)	10	916000	
acA1300-75gm/gc**	112	1000000	
acA1600-20gm/gc	25	1000000	
acA1600-60gm/gc (*)	10	840000	
acA1920-40gm/gc**	8-bit pixel format: 76 12-bit pixel format: 90	10000000	
acA1920-50gm/gc**	63	10000000	
acA2000-50gm/gc,** acA2000-50gmNIR**, acA2040-25gm/gc**, acA2040-25gmNIR**	24	10000000	

* Switchable shutter mode. See Table 28 on page 159.

** The minimum allowed exposure time values indicated above already include the exposure time offset. For information about the exposure time offset on these camera models, see page 145.

Table 25: Minimum and Maximum Allowed Exposure Time Setting (µs) for Global Shutter Operation

Rolling Shutter Operation [µs]			
Camera Model	**Minimum Allowed Exposure Time**	**Maximum Possible Exposure Time**	**Can be set in increments of ...**
acA1280-60gm/gc/, acA1300-60gm/gc (*), acA1300-60gmNIR (*)	15	896000	1
acA1600-60gm/gc (*)	35	840000	
acA1920-25gm/gc, acA2500-14gm/gc (*)	35	9999990	35
acA3800-10gm/gc (*)	35	1600000	1
acA4600-7gc (*)	35	1460000	
* Switchable shutter mode. See Table 28 on page 159.			

Table 26: Minimum and Maximum Allowed Exposure Time Setting (µs) for Rolling Shutter Operation

You can use the Basler pylon API to set the ExposureTimeAbs 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 63.

6.7 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**. For rolling shutter, there are two sub-types: electronic rolling shutter (ERS) and global reset release mode (GRR).

The following table shows what kind of shutter can be used in the different camera models.

Camera Model	Gobal Shutter	Rolling Shutter	
		ERS Mode (default)	GRR Mode
All models; exceptions see below	x	-	-
acA1280-60	-	x	-
acA1920-25 (*)	-	x	x
acA2500-14 (*)	-	x	x
acA1300-60 (*)	x (default)	x	x
acA1600-60 (*)	x (default)	x	x
acA3800-10 (*)	-	x	x
acA4600-7 (*)	-	x	x
* For the cameras with an asterix (*), you can switch between the indicated shutter modes.			

Table 27: Camera Models and Possible Shutter Modes

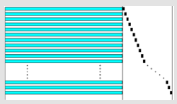
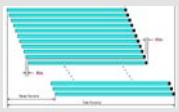
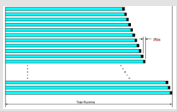
Global Shutter	Electronic Rolling Shutter (ERS)	Global Reset Release Mode (GRR)
		
For moving objects	▪ For stationary objects/**not** moving objects ▪ Lower ambient noise ▪ If used for moving objects: Use of flash lighting and flash window recommended	▪ For stationary objects/**not** moving objects ▪ Use of flash lighting and flash window is a must.

Table 28: Overview of Shutter Modes

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

6.7.1 Global Shutter

Available for	Not Available for
All models **Note** Only valid for acA1300-60 and 1600-60, if they are operated in the global shutter mode. (*)	acA1280-60, acA1920-25, acA2500-14, acA3800-10(*) (**), acA4600-7 (*) (**)
* The camera models marked with an asterik (*) have a switchable shutter mode (see Section 4.4 on page 72 and Section 6.7 on page 159). ** The camera models marked with (**) don't have an exposure active signal. You can set a flash window for these cameras. For information about the flash window, see Section 6.7.2.1 on page 167.	

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.

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.

In general, cameras that operate in the global shutter mode, 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 ReadoutTimeAbs parameter.

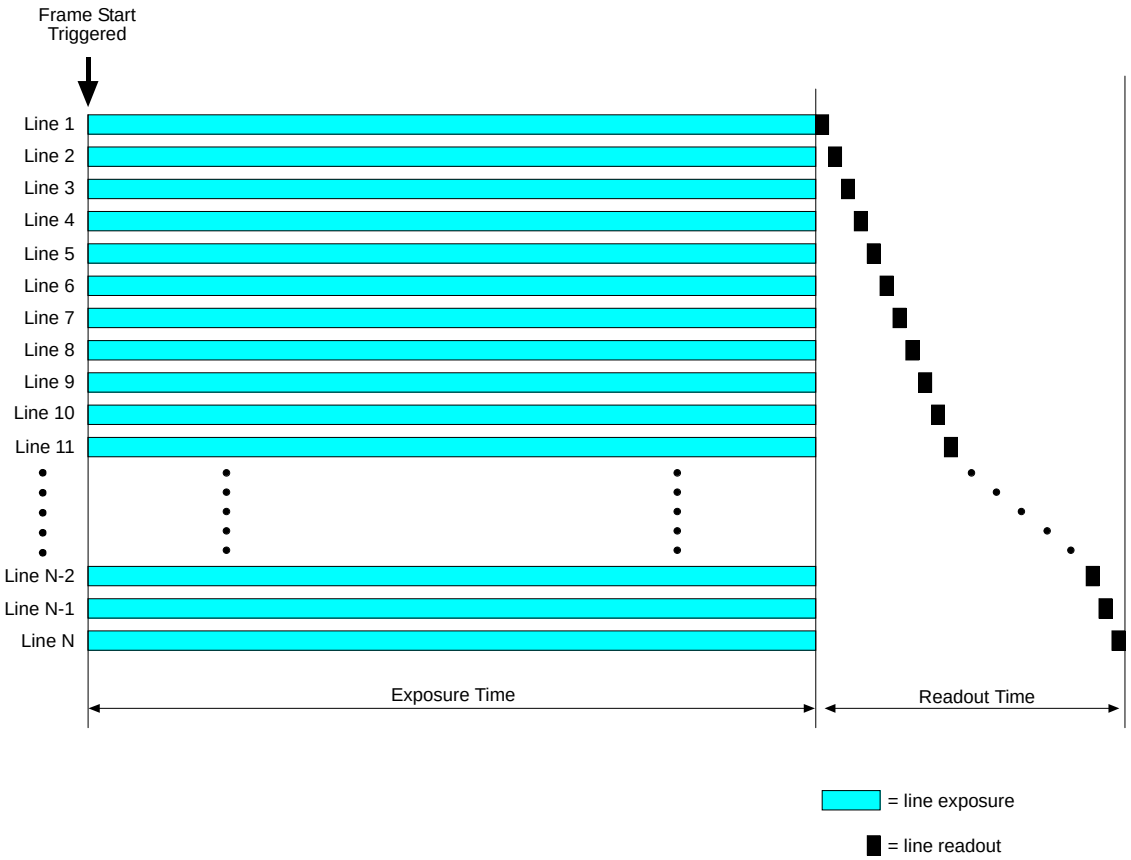


Fig. 75: Global Shutter

For more information about

- the exposure active output signal, see Section 6.10.1 on page 177.
- the ReadoutTimeAbs parameter, see Section 6.11 on page 192.

6.7.2 Rolling Shutter

Available for	Not Available for
acA1280-60, acA1920-25, acA2500-14, acA3800-10, and acA4600-7 (*) acA1300-60 (*) and acA1600-60 (*): Only valid if they are operated in the rolling shutter mode.	All other models
(*) The camera models marked with an asterik (*) have a switchable shutter mode (see Section 4.4 on page 72 and Section 6.7 on page 159).	

The 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 (ERS mode) and
- global reset release mode (GRR mode).

Electronic Rolling Shutter Mode (ERS Mode)

When the shutter is in the electronic rolling shutter 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. And so on until the bottom line of pixels is reached (see Figure 76).

The exposure time is the same for all lines and is determined by the ExposureTimeAbs 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.

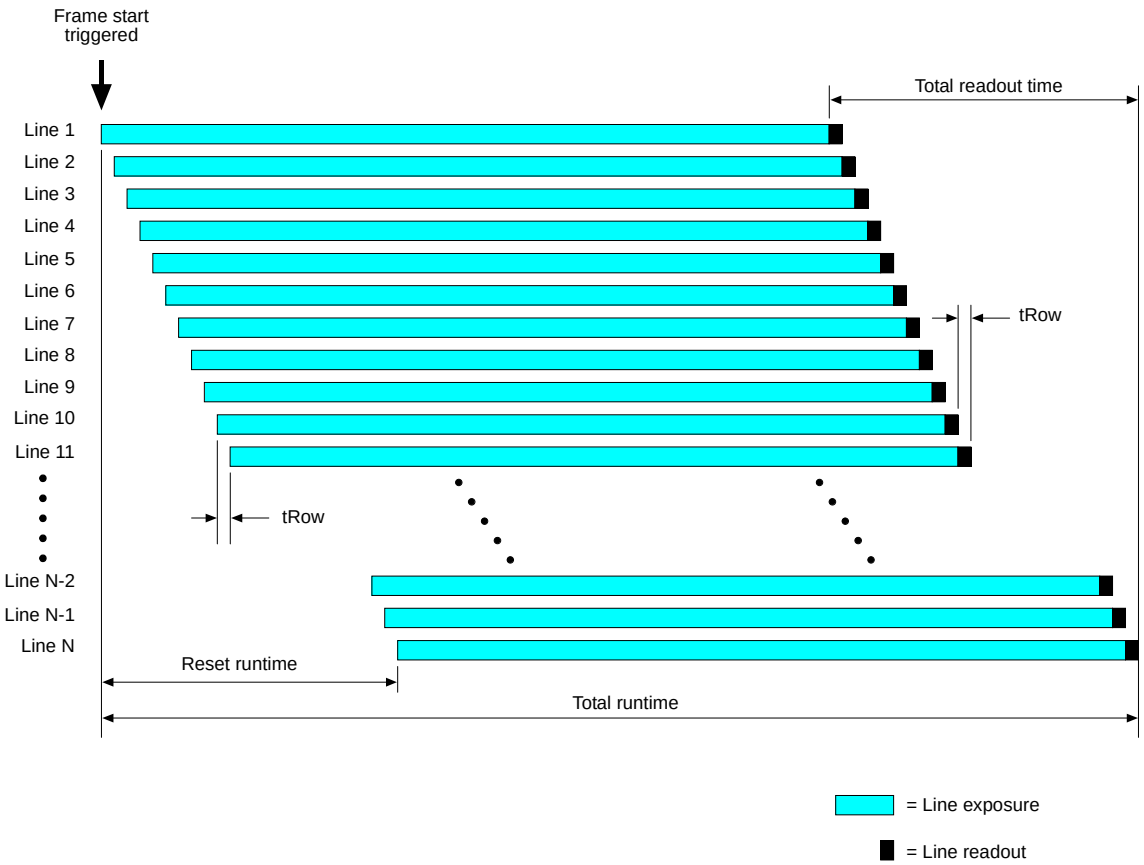


Fig. 76: Rolling Shutter in the ERS Mode

For calculating the ...	Formula	Notes
Reset Runtime	tRow x (AOI Height -1)	tRow: see Table 30.
Total Readout Time	[tRow x (AOI Height)] + C_{TRT} µs	C_{TRT} = Constant for evaluating total readout time. See Table 30.
Total Runtime	ExposureTimeAbs parameter + Total readout time	-

Table 29: Formulas for Calculating the Runtime and Readout Time (ERS Mode)

	In ERS mode, the flash window signal will not be available when the exposure time for the first row elapses before exposure for the last row of the current AOI has started, i.e. when Exposure Time ≤ Reset Runtime.

Camera Model	tRow	C_{TRT} [Constant for calculating the total readout time]	
		Mono/Mono NIR	Color
acA1280-60 acA1300-60	14 µs	13 x tRow	14 x tRow
acA1600-60	▪ 8-bit: 13 µs ▪ For pixel formats > 8 bit: 17 µs:		
acA1920-25gm/gc acA2500-40gm/gc	35 µs	490 µs (ERS mode) 810 µs (Global reset release mode)	
acA1920-50gm/gc	16 µs	40 x tRow	41 x tRow
acA3800-10gm/gc	▪ 8 bit: 31.6 µs ▪ 12 bit packed: 36.4 µs ▪ 12 bit: 39.6 us	ERS: 143 x tRow GRR: 901 x tRow	ERS: 144 x tRow GRR: 902 x tRow
acA4600-7gc	▪ 8 bit: 39.4 µs ▪ 12 bit packed: 43.4 µs ▪ 12 bit: 47.4 µs	-	ERS: 147 x tRow GRR: 756 x tRow

Table 30: Parameters for Evaluating the Readout Time (ERS Mode)

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 the last line 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 6.10.1 on page 177.
- the ExposureTimeAbs parameter, see Section 6.6 on page 156.
- the flash window, see Section 6.7.2.1 on page 167.

Global Reset Release Mode (GRR)

In the global reset release mode, all of the lines in the sensor reset and begin exposing when frame start is triggered. 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 ExposureTimeAbs parameter.
- for line two will end tRow after the exposure ends for line one.
- 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 77).

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 (see Table 30).

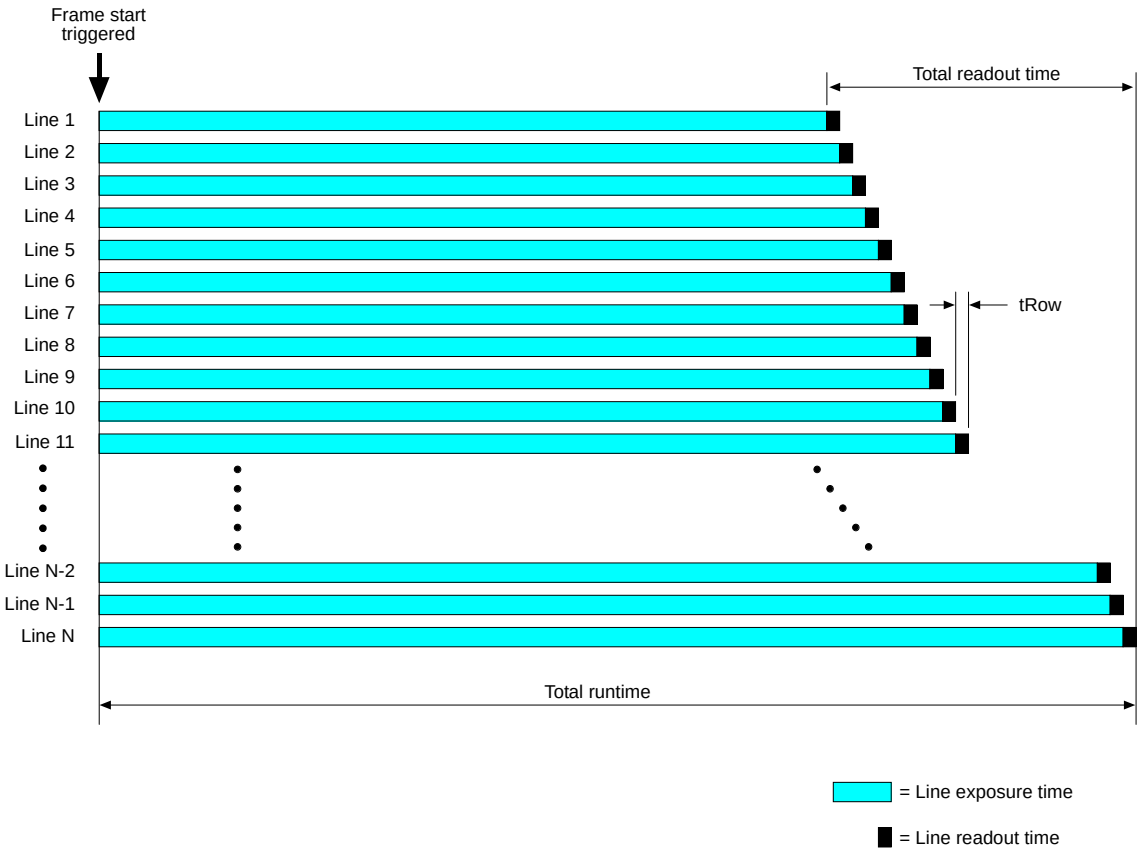


Fig. 77: Rolling Shutter in the Global Reset Release Mode

For calculating the ...	Formula	Notes
Total Readout Time	[tRow x (AOI Height)] + C_{TRT} µs	C_{TRT} = Constant for total readout time. See Table 30.
Total Runtime	ExposureTimeAbs parameter + Total readout time	-

Table 31: Formulas for Calculating the Runtime and Readout Time (Global Reset Release Mode)

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 the last line ends.

When the camera is operating with the rolling shutter in the global reset 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 6.10.1 on page 177.
- the ExposureTimeAbs parameter, see Section 6.6 on page 156.
- the flash window, see Section 6.7.2.1 on page 167.

Setting the Shutter Mode

The camera's shutter has two operating modes:

- electronic rolling shutter mode and
- global reset release mode.

If global reset release mode is

- **disabled**, the shutter will operate in the electronic rolling shutter mode.
- **enabled**, the shutter will operate in the 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.

6.7.2.1 The Flash Window

Flash Window in Electronic Rolling Shutter Mode (ERS)

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. **Flash window** = period of time during a frame acquisition when all of the lines in the sensor are open for exposure.

For calculating ...	Formula	Notes
Time to Flash Window Open	tRow x (AOI Height -1)	tRow: see Table 30 on page 164.
Flash window width	ExposureTimeAbs parameter - [(tRow x (AOI Height - 1)]	
Min. exposure time for flash window in ERS mode	Exposure time > tRow x (AOI Height - 1)	

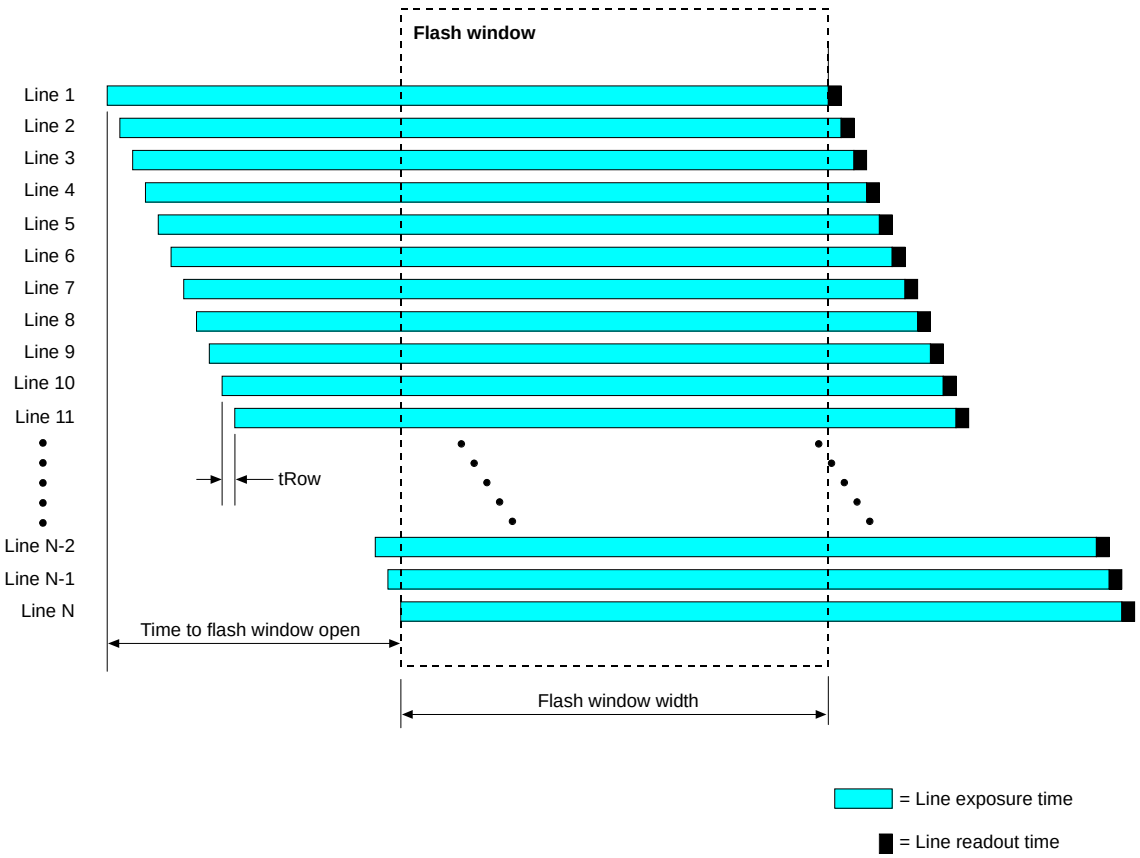


Fig. 78: Flash Window for Rolling Shutter in the ERS Mode

For more information about the ExposureTimeAbs parameter, see Section 6.6 on page 156.

Flash Window in Global Reset Release Operating mode (GRR)

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.
- **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.
Fash window = period of time during a frame acquisition when all of the lines in the sensor are open for exposure.

In global reset release mode, the flash window opens when the frame is triggered and closes after a time period equal to the ExposureTimeAbs parameter setting. Thus, the flash window width is equal to the ExposureTimeAbs parameter setting.
Flash window width = how long the flash window will remain open.

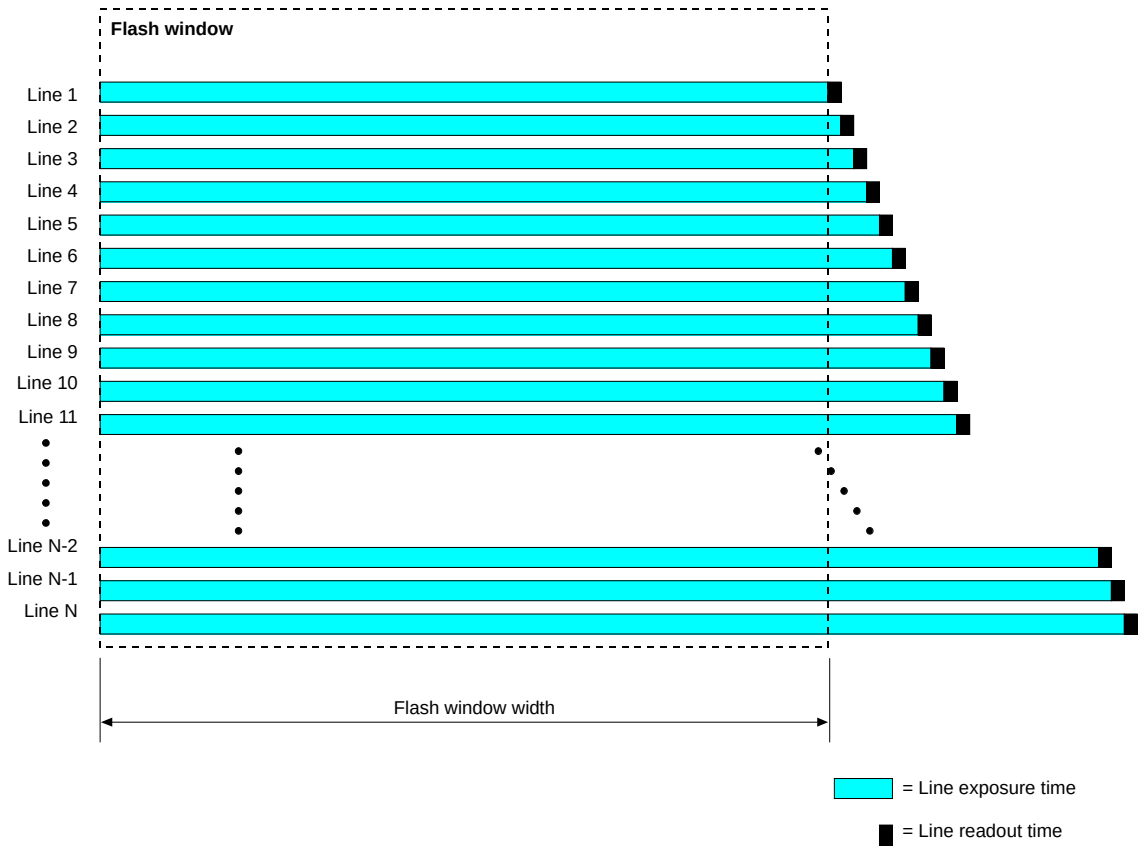


Fig. 79: Flash Window for Rolling Shutter in the Global Reset Release Mode

For more information about the ExposureTimeAbs parameter, see Section 6.6 on page 156.

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 89 illustrates the flash window signal on a camera with the shutter operating in the electronic rolling shutter mode.

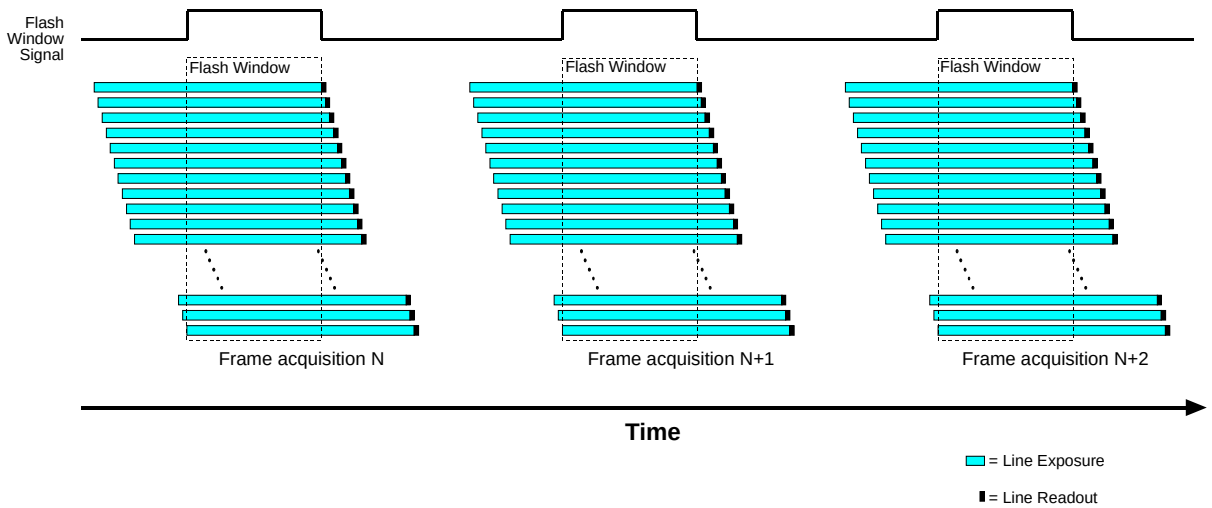


Fig. 80: Flash Window Signal on Cameras with a Rolling Shutter

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

For more information about the flash window signal, see Section 6.10.2 on page 180.

6.8 Overlapping Image Acquisitions - (Global Shutter Models)

Available for	Not Available for
All models with global shutter **Note** Only valid for acA1300-60 (*) and 1600-60 (*), if they are operated in the global shutter mode.	acA1280-60, acA1920-25, acA2500-14, acA3800-10, and acA4600-7 (*)
(*) The acA1300-60 and acA1600-60 cameras can only realize overlapped image acquisitions in the global shutter mode if they are operated in the free run mode; that means: ▪ the acquisition start trigger is set to off, ▪ the frame start trigger is set to off and ▪ the acquisition mode is set to continuous As soon as the sequencer is enabled, overlapping image acquisition is automatically disabled.	

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 81 with the camera set for the trigger width exposure mode.

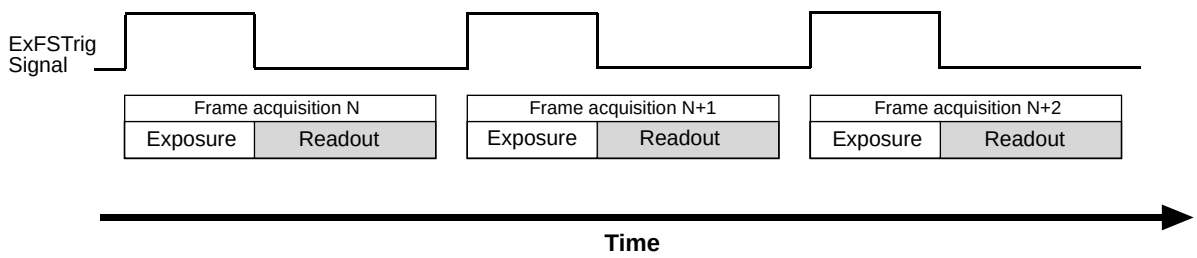


Fig. 81: Non-overlapped Exposure and Readout

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 82 with the camera set for the trigger width exposure mode.

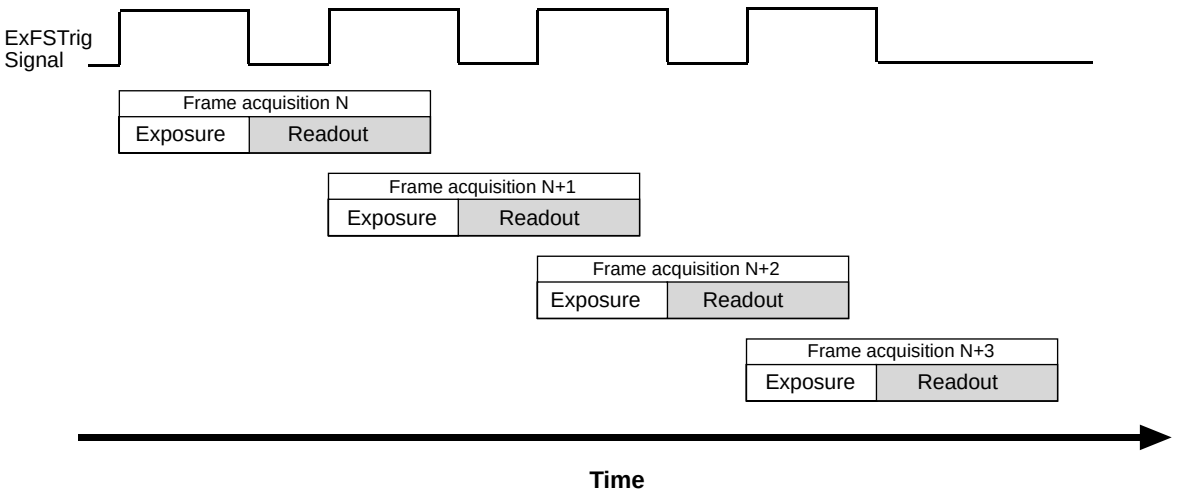


Fig. 82: Overlapped Exposure and Readout

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 ...	If
not overlap	Frame Period > ExposureTimeAbs parameter + Total readout time
overlap	Frame Period ≤ ExposureTimeAbs parameter + Total readout time

You can determine the readout time by reading the value of the ReadoutTimeAbs parameter. The parameter indicates what the readout time will be in microseconds given the camera’s current settings.

You can read the ReadoutTimeAbs 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 63.

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 FrameStartOvertrigger event.
This situation is illustrated in Figure 83 with the camera set for the trigger width exposure mode with rising edge triggering.

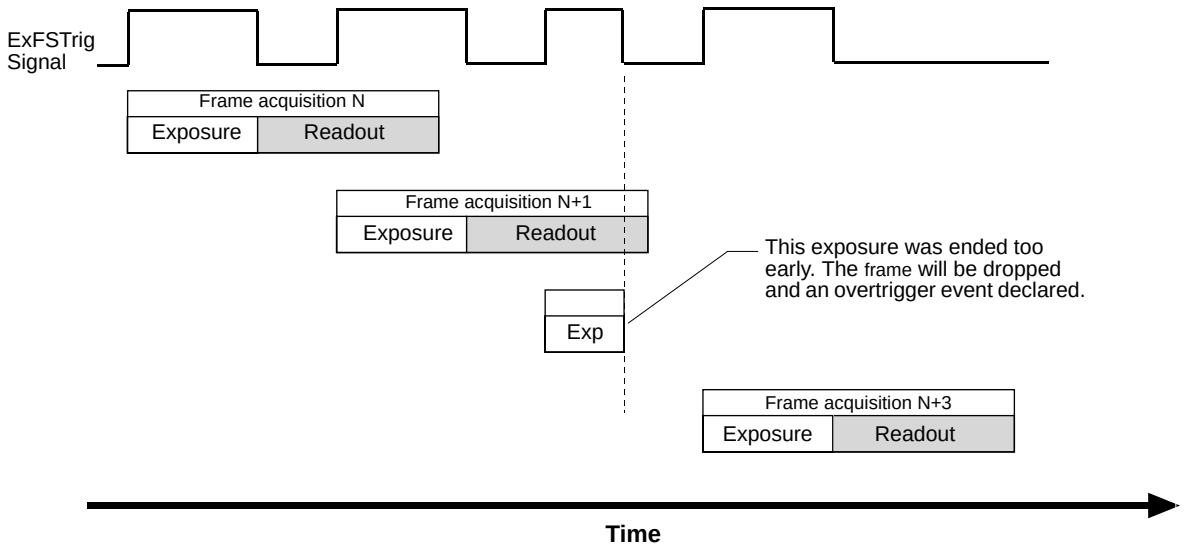


Fig. 83: Overtriggering Caused by an Early End of Exposure

You can avoid violating this guideline by using the camera's FrameTriggerWait signal to determine when exposure can safely begin and by properly setting the camera's ExposureOverlapTimeMaxAbs parameter.

For more information about

- the FrameTriggerWait signal and the ExposureOverlapTimeMaxAbs parameter, see Section 6.10.4 on page 183.
- trigger width exposure, see Section 6.4.3.2 on page 143.

6.9 Overlapping Image Acquisitions - (Rolling Shutter Models)

Available for	Not Available for
acA1280-60, acA1300-60, 1600-60-25, acA2500-14, acA3800-10, and acA4600-7 (*) **Note** Only valid for acA1300-60 and 1600-60, if they are operated in the rolling shutter mode.	All other models
* The acA1280-60, acA1300-60, acA1600-60, acA3800-10, and acA4600-7 cameras can only realize overlapped image acquisitions if they are operated in the free run mode; that means: ▪ the trigger mode is set to off (acquisition start trigger/frame start trigger), ▪ the acquisition mode is set to continuous. As soon as the sequencer is enabled, overlapping image acquisition is automatically disabled.	

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 84 with the camera using an external frame start trigger.

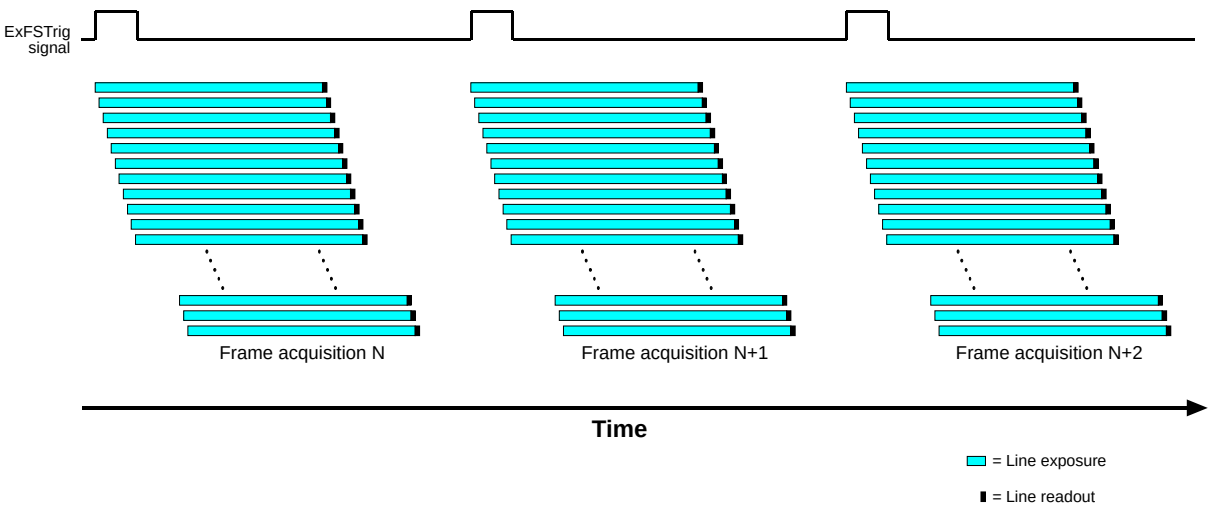


Fig. 84: Non-overlapped Acquisition

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 85.

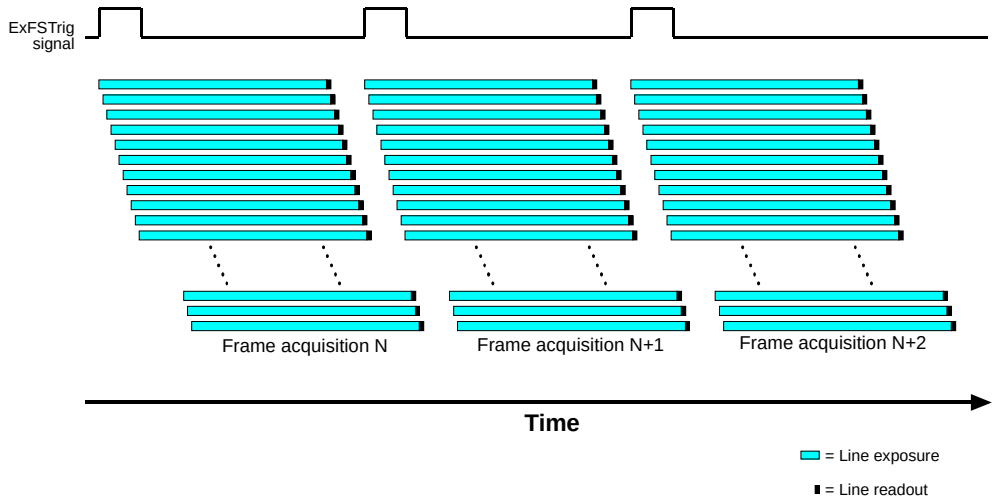


Fig. 85: Overlapped Exposure and Readout

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 ...	If
not overlap	Frame period > ExposureTimeAbs parameter + Total readout time
overlap	Frame period ≤ ExposureTimeAbs parameter + Total readout time

Overlapped frame acquisition

- **can only** be performed when the camera is in the electronic rolling shutter mode (ERS).
- **cannot** be performed when the camera is set for global reset release rolling shutter mode.

If you use the acA1920-25 and 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 ReadoutTimeAbs 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 ReadoutTimeAbs 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 63.

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 86.

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

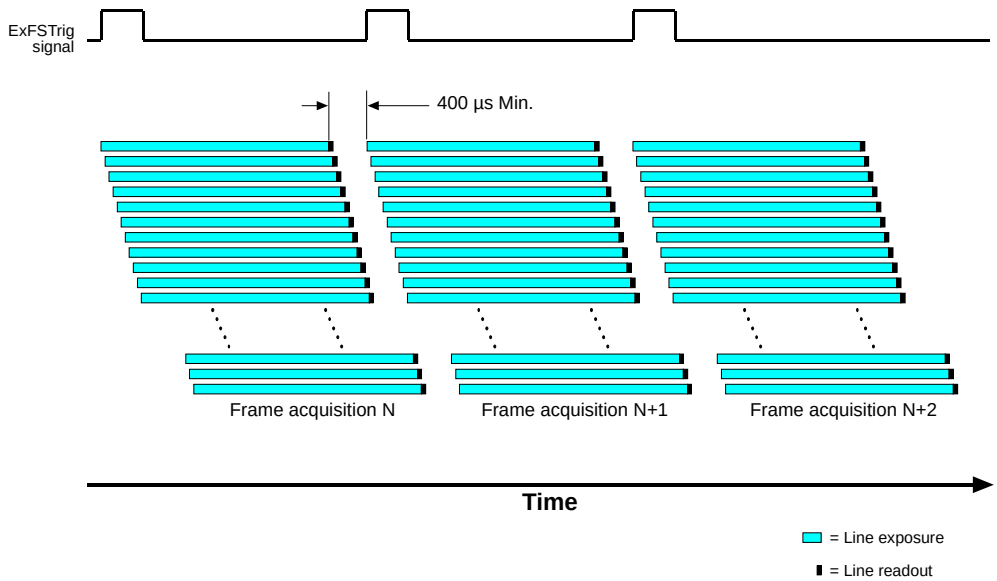


Fig. 86: Acquisition Overlap Guideline

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

Overlapped frame acquisition

- **can only** be performed when the camera is in the electronic rolling shutter mode (ERS).
- **can not** be performed when the camera is set for global reset release rolling shutter mode.

6.10 Acquisition Monitoring Tools

6.10.1 Exposure Active Signal

Exposure Active on Global Shutter Cameras

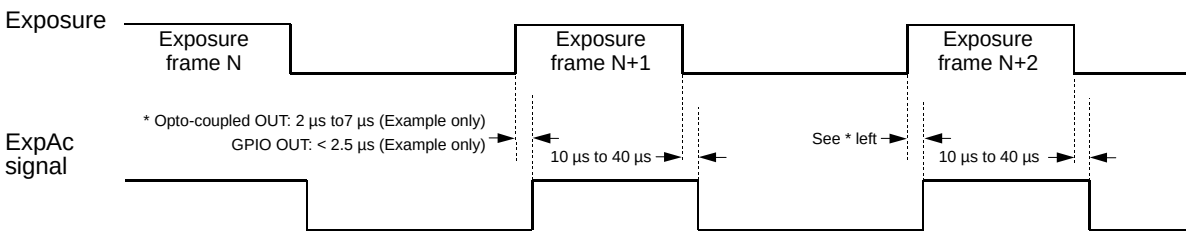
Available for	Not Available for
All cameras For acA1300-60 and 1600-60: see *	acA1920-25, acA2500-14, acA3800-10**, and acA4600-7**
* Only if they are operated in the global shutter mode. ** For these cameras the flash window signal is available and may in some cases serve as an alternative (see Section 6.7.2.1 on page 167).	

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 87.

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.



Timing charts are not drawn to scale.
Times stated are **only** given as an examples. They are **only valid** for the operating conditions given in Section 5.10.4

Fig. 87: Exposure Active Signal on Cameras with a Global shutter

See note next page.

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 87 for details.
- in the acA2000-50 and acA2040-25 cameras, make sure that the exposure active signal is set in a way that it observes the exposure time offset value automatically set by the sensor. For information on the exposure time offset, see "Trigger Width Exposure Mode with Special Exposure Time Offset" on page 145.
- Using the GPIO line, set for output, will bring about shorter delays, compared to using the opto-isolated output line. The exact delays depend on several factors See Section 5.9.2 on page 96 for details.

Exposure Active on Rolling Shutter Cameras

Available For	Not Available For
acA1300-60 (*), 1600-60 (*), acA1920-25, acA2500-14	All other models
* Only if they are operated in the rolling shutter mode.	

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 last line ends as shown in Figure 88.

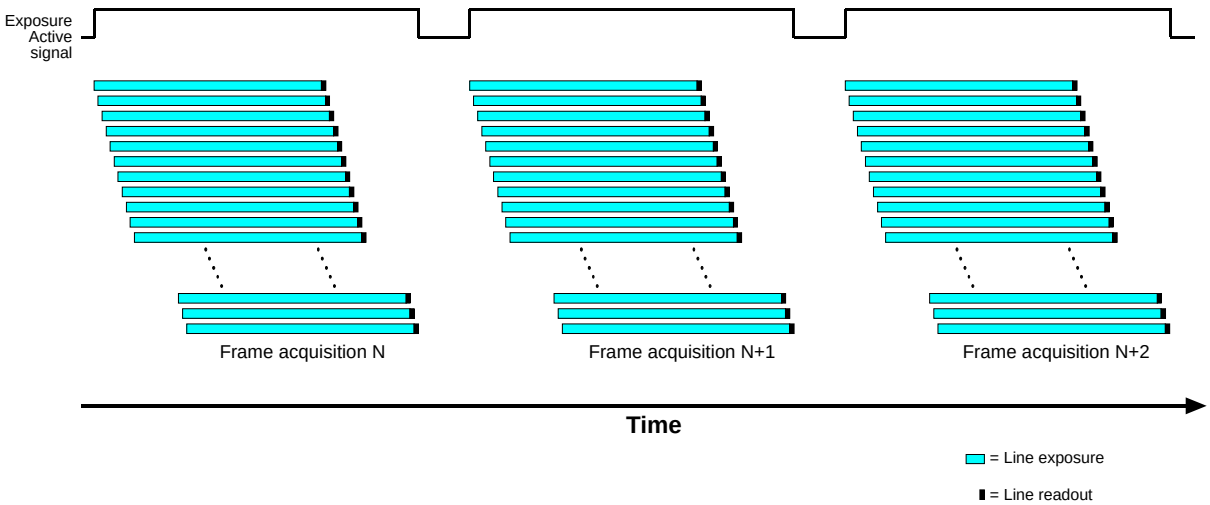


Fig. 88: Exposure Active Signal on Cameras with a Rolling Shutter

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 an output line.

To select a source signal for the output line:

Example For Models without GPIO	Example For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Set the LineSource parameter to the exposure active output signal.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Set the value of the LineSource parameter to the exposure active output signal.

The following code snippet illustrates using the API to set the parameters

For Models without GPIO

```
Camera.LineSelector.SetValue(LineSelector_Out1);
Camera.LineSource.SetValue(LineSource_ExposureActive);
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
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 63.
- changing which camera output signal is selected as the source signal for the output line, see Section 5.11.1 on page 102.
- the electrical characteristics of the camera's output line, see Section 5.7 on page 83.

6.10.2 Flash Window Signal

Available For	Not Available For
acA1280-60, acA1300-60 (*), 1600-60 (*), acA1920-25, acA2500-14, acA3800-10, and acA4600-7	All other models
* Only if they are operated in the rolling shutter mode.	

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 89 illustrates the flash window signal on a camera with the shutter operating in the electronic rolling shutter mode.

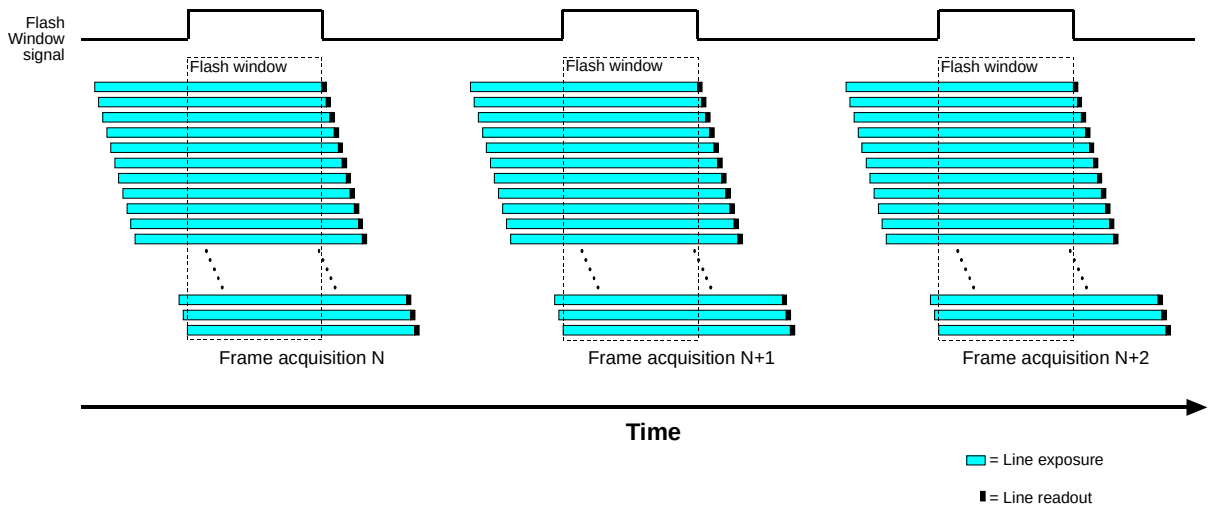


Fig. 89: Flash Window Signal on Cameras with a Rolling Shutter

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

For more information about the rolling shutter and the flash window, see Section 6.7.2 on page 162.

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 a camera output line.

To select a source signal for an output line:

Example For Models without GPIO	Example For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Set the LineSource parameter to the flash window signal.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Set the value of the LineSource parameter to the flash window signal.

The following code snippet illustrates using the API to set the parameters

For Models without GPIO

```
Camera.LineSelector.SetValue(LineSelector_Out1);
Camera.LineSource.SetValue(LineSource_FlashWindow);
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
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 63.
- changing which camera output signal is selected as the source signal for the output line, see Section 5.11.1 on page 102.
- the electrical characteristics of the camera's output line, see Section 5.7 on page 83.

6.10.3 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 ignore the trigger signal and will generate an acquisition start overtrigger event.
- a **software frame start trigger** signal when it is not in a "waiting for frame start trigger" acquisition status, it will 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:

1. Use the AcquisitionStatusSelector to select the AcquisitionTriggerWait status or the FrameTriggerWait status.
2. Read the value of the AcquisitionStatus 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();
```

6.10.4 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 ignore the trigger signal and will generate an acquisition start overtrigger event.
- a **hardware frame start trigger** signal when it is not in a "waiting for frame start trigger" acquisition status, it will ignore the trigger signal and will generate a frame start overtrigger event.

AcquisitionTriggerWait Signal	FrameTriggerWait Signal
Gives you the ability to check whether the camera is in a "waiting for acquisition start trigger" acquisition status.	Gives you the ability to check whether the camera is in a "waiting for frame start trigger" acquisition status.
These signals are designed to be used when you are triggering acquisition start or frame start **via a hardware trigger signal**.	
If you check the acquistion or frame trigger wait signal before you apply each corresponding hardware start trigger signal, you can avoid applying acquisition or frame start trigger signals to the camera that will be ignored.	

6.10.4.1 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.
- 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".
- 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 90 illustrates the Acquisition Trigger Wait signal with the AcquisitionFrameCount 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.

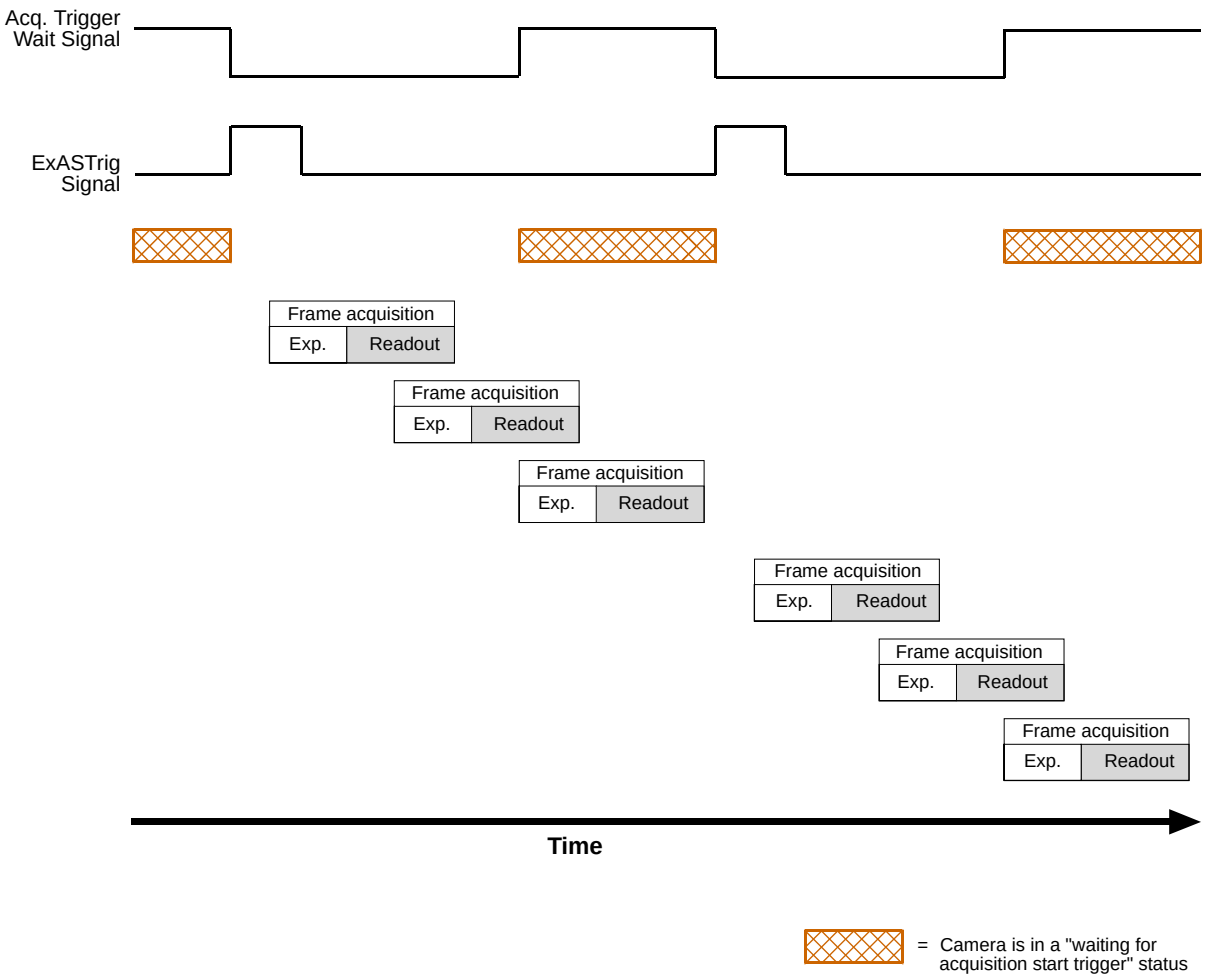


Fig. 90: Acquisition Trigger Wait Signal

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

For more information about event reporting, see Section 9.16 on page 350.

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

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

To select a source signal for an output line:

Example For Models without GPIO	Example For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Set the LineSource parameter to the acquisition trigger wait signal.	1. Use the LineSelector parameter to select Line3. 2. Set the LineMode parameter to Output. 3. Set the value of the LineSource parameter to the acquisition trigger wait signal.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
Camera.LineSelector.SetValue(LineSelector_Out1);
Camera.LineSource.SetValue(LineSource_AcquisitionTriggerWait);
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line3);
Camera.LineMode.SetValue(LineMode_Output);
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 63.
- changing which camera output signal is selected as the source signal for the output line, see Section 5.11.1 on page 102.
- the electrical characteristics of the camera's output line, see Section 5.7 on page 83.

6.10.4.2 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 FrameTriggerWait 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 91 illustrates the FrameTriggerWait 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.

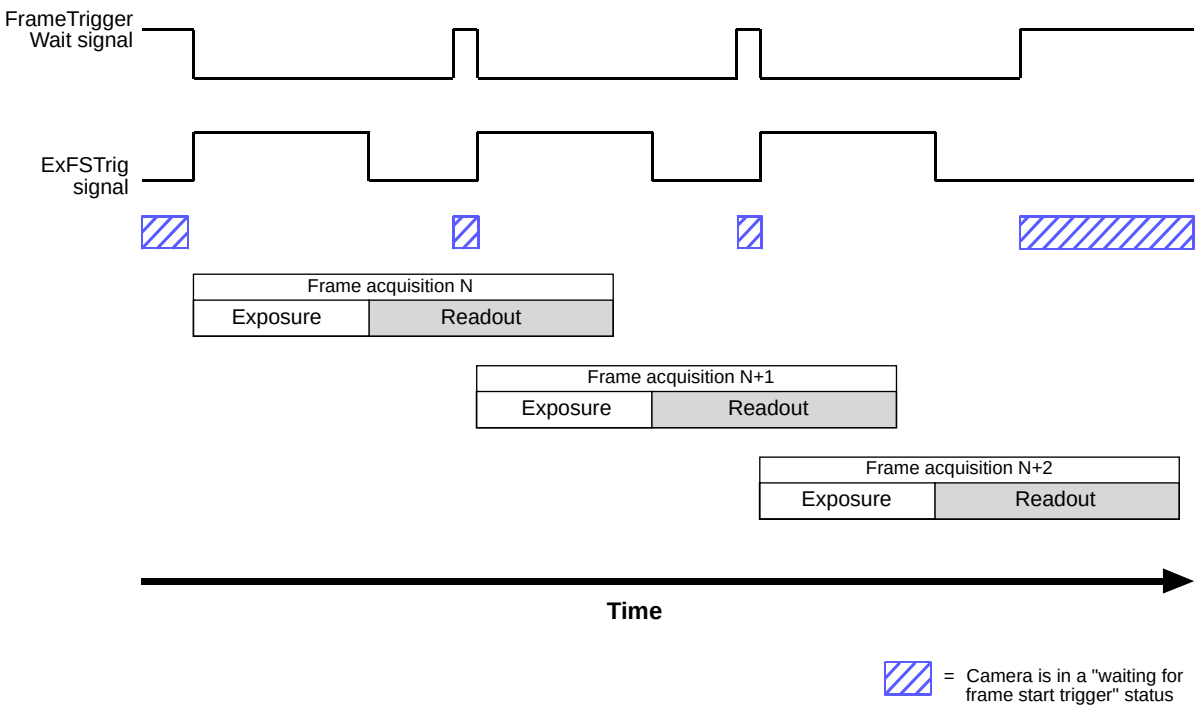


Fig. 91: Frame Trigger Wait Signal

	The FrameTriggerWait signal will only be available when hardware frame start triggering is enabled.

For more information about

- event reporting, see Section 9.16 on page 350.
- hardware triggering, see Section 6.4.3 on page 142.

FrameTriggerWait Signal Details (Global Shutter)

Available for	Not Available for
All camera models that are operated in gobal shutter mode. For acA1300-60, acA1600-60: Only valid if they are operated in the global shutter mode.	Camera models that are operated in rolling shutter mode: acA1280-60, acA1300-60 (*), acA1600-60 (*), acA1920-50, acA2500-14,acA3800-10, acA4600-7 (*) If operated in rolling shutter mode.

When the camera is set for the timed exposure mode, the rise of the FrameTriggerWait signal is based on the current ExposureTimeAbs parameter setting and on when readout of the current frame will end. This functionality is illustrated in Figure 92.

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

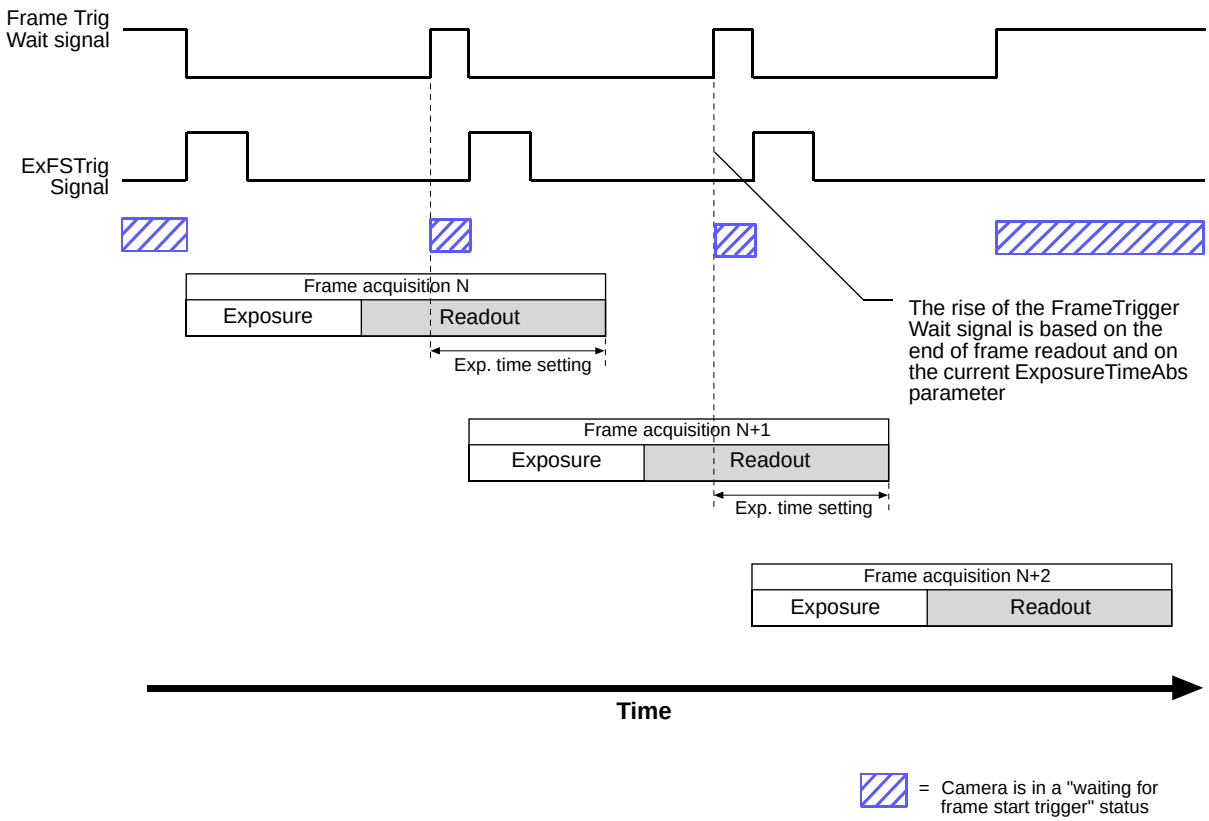


Fig. 92: Frame Trigger Wait Signal with the Timed Exposure Mode

When the camera is set for the trigger width exposure mode, the rise of the FrameTriggerWait signal is based on the ExposureOverlapTimeMaxAbs parameter setting and on when readout of the current frame will end. This functionality is illustrated in Figure 93.

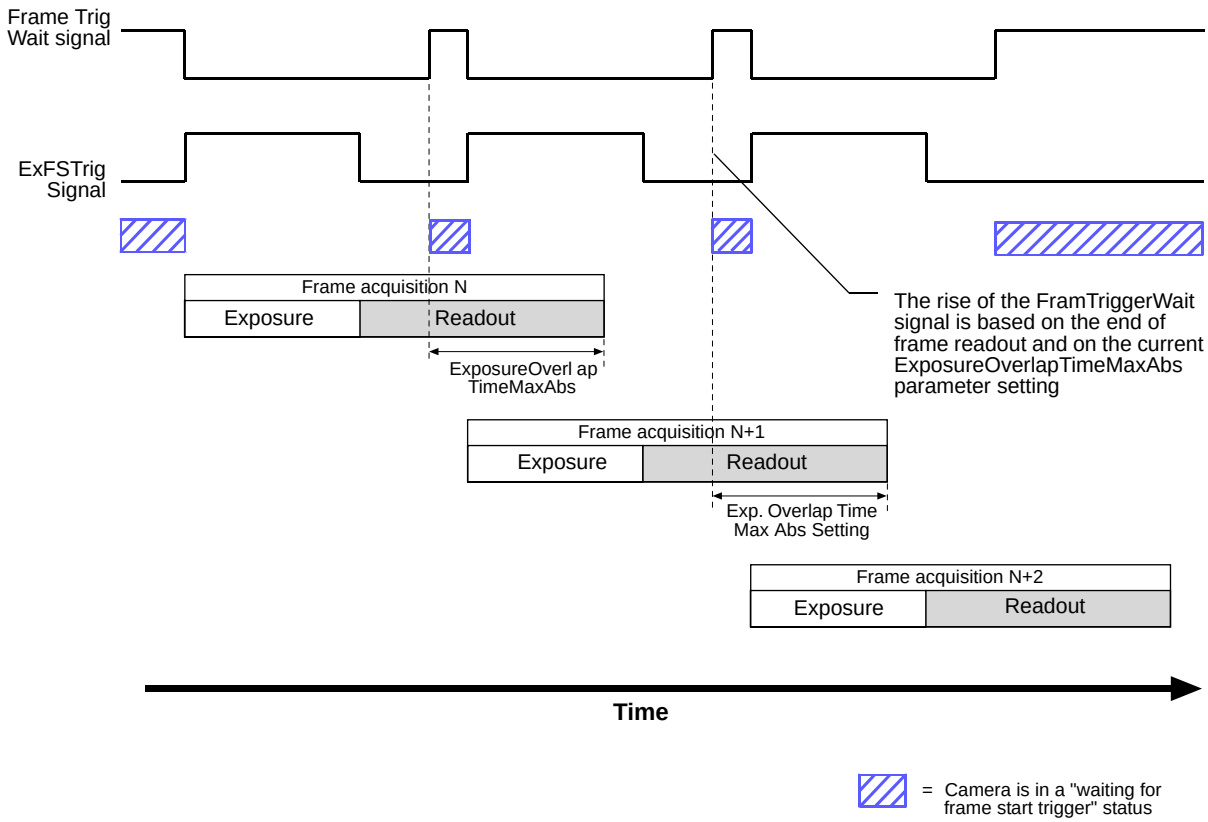


Fig. 93: Frame Trigger Wait Signal with the Trigger Width Exposure Mode

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 ExposureOverlapTimeMaxAbs 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 ExposureOverlapTimeMaxAbs parameter.
- Monitoring the camera's FrameTriggerWait signal and only using the ExFSTrig signal to start exposure when the FrameTriggerWait signal is high.

You should set the ExposureOverlapTimeMaxAbs 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 ExposureOverlapTimeMaxAbs parameter to 3000 µs.

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

```
// If the camera model is a camera with GPIO line, you first have to set the
ExposureOverlapTimeMode parameter:
//Set the exposure overlap time mode
Camera.ExposureOverlapTimeMode.SetValue(ExposureOverlapTimeMode_Manual);

// Valid for all cameras:
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 63.
- the electrical characteristics of the camera's output line, see Section 5.7 on page 83.
- which camera model has GPIO, see Section 5.2 on page 74.

FrameTriggerWait Signal Details (Rolling Shutter)

Available for	Not Available for
Camera models that are operated in rolling shutter mode: acA1280-60, acA1300-60 (*), acA1600-60 (*), acA1920-50, acA2500-14,acA3800-10, acA4600-7	All camera models operated in global shutter mode
* Only valid if the acA1300-60 (*) and acA1600-60 (*) are operated in the rolling shutter mode.	

For cameras with a rolling shutter, the rise of the FrameTriggerWait 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 94.

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

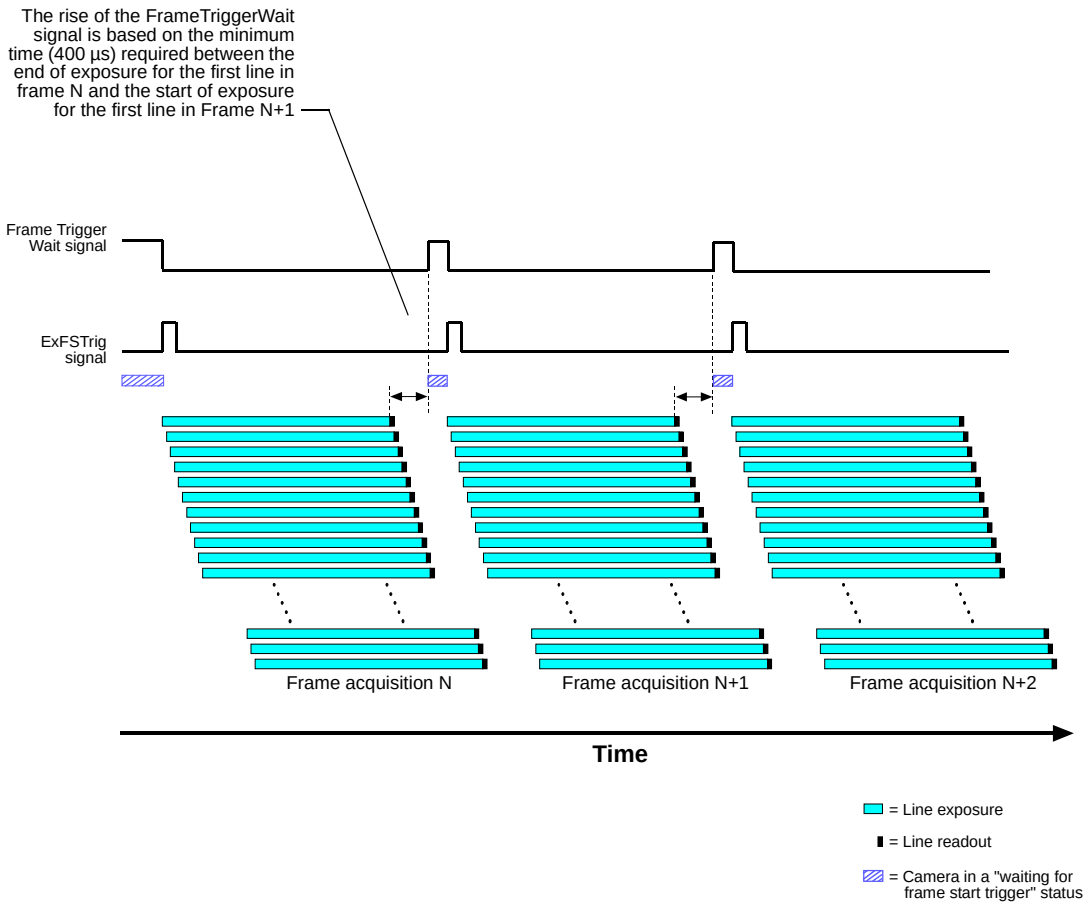


Fig. 94: FrameTriggerWait Signal on a Rolling Shutter Camera

Selecting the FrameTriggerWait Signal as the Source Signal for an Output Line

The FrameTriggerWait signal can be selected to act as the source signal for a camera output line.

To select a source signal for an output line:

Example For Models without GPIO	Example For Models with GPIO
1. Use the LineSelector parameter to select output line 1. 2. Set the LineSource parameter to the FrameTriggerWait signal.	1. Use the LineSelector parameter to select Line2. 2. Set the LineMode parameter to Output. 3. Set the value of the LineSource parameter to the FrameTriggerWait signal.

The following code snippet illustrates using the API to set the parameters:

For Models without GPIO

```
Camera.LineSelector.SetValue(LineSelector_Out1);
Camera.LineSource.SetValue(LineSource_FrameTriggerWait);
```

For Models with GPIO

```
Camera.LineSelector.SetValue(LineSelector_Line2);
Camera.LineMode.SetValue(LineMode_Output);
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 63.
- changing which camera output signal is selected as the source signal for the output line, see Section 5.11.1 on page 102.
- the electrical characteristics of the camera's output line, see Section 5.7 on page 83.

6.10.5 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 9.16 on page 350.

6.11 Acquisition Timing Chart

Figure 95 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 95 on page 194 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	Global Shutter		Rolling Shutter	
	Frame Acquisitions Not Overlapped Exposure Start Delay [µs]	**Frame Acquisitions Overlapped** Exposure Start Delay with maximum jitter included [µs]	**ERS Mode (Default)**	**GRR Mode**
acA640-90gm/gc	21.48	-	-	-
acA640-120gm/gc	17.62	-	-	-
acA640-300gm/gc	6.9	16.2	-	-
acA645-100gm/gc	21.84	-	-	-
acA750-30gm/gc	48.97	-	-	-
acA780-75gm/gc	24.50	-	-	-
acA800-200gm/gc	6.9	16.2	-	-

Table 32: Exposure Start Delays [µs]

Camera Model	Global Shutter		Rolling Shutter	
	Frame Acquisitions Not Overlapped Exposure Start Delay [µs]	**Frame Acquisitions Overlapped** Exposure Start Delay with maximum jitter included [µs]	**ERS Mode (Default)**	**GRR Mode**
acA1280-60gm/gc	-	-	190 to 200 µs (*)	-
acA1300-22gm/gc acA1300-30gm/gc	34.50	-	-	-
acA1300-60gm/gc acA1300-60gmNIR	43	-	190 to 200 µs (*)	35 to 48 µs (*)
acA1300-75gm/gc	6.9	16.2	-	-
acA1600-20gm/gc	45.54	-	-	-
acA1600-60gm/gc	41.5	41.5	180 to 240 µs (**)	34 to 51 µs (**)
acA1920-40gm/gc	▪ 8-bit pixel format: 57.6 ▪ 12-bit pixel format: 74.4	▪ 8-bit pixel format: 57.6 ▪ 12-bit pixel format: 74.4	-	-
acA1920-50gm/gc	47.4	47.4	-	-
acA2000-50gm/gc	0.2	0.2 to 17.2	-	-
acA2000-50gmNIR				
acA2040-25gm/gc				
acA2040-25gmNIR				
acA1920-25gm/gc acA2500-14gm/gc	-	-	848 to 883 µs (***)	848 µs
acA3800-10gm/gc	-	-	gm: 2900 µs gc: 2550 µs	gm: 2970 µs gc: 2620 µs
acA4600-7gc	-	-	7700 µs	7800 µs

* Depends on the exposure time

** Depends on the exposure time and the pixel format.

*** Depends on whether the frame acquisitions are overlapped or not overlapped.

Table 32: Exposure Start Delays [µs]

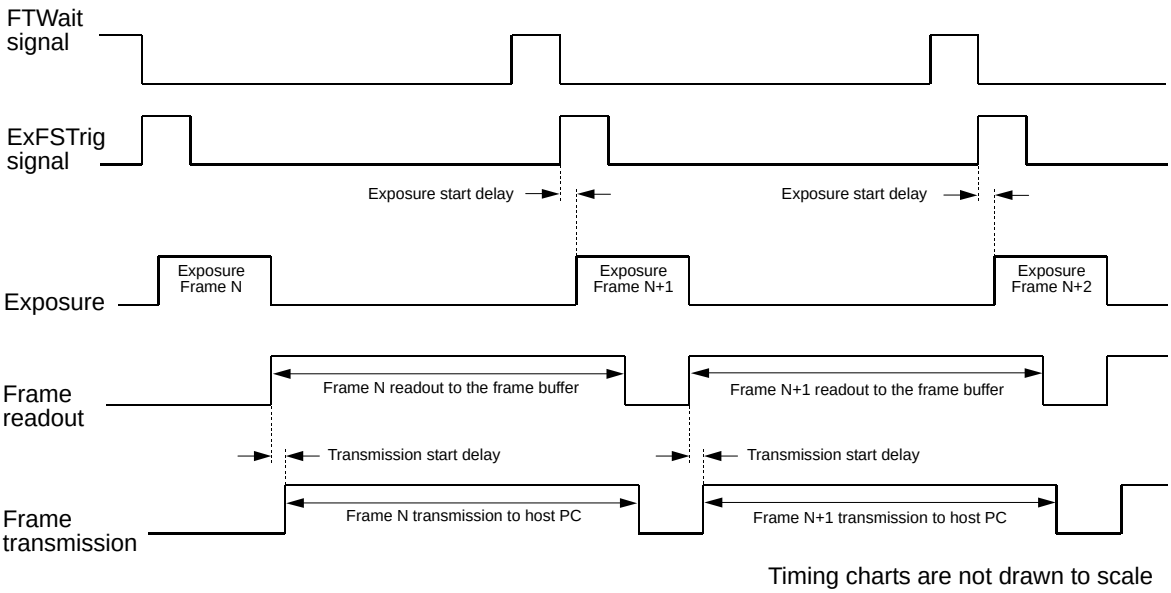


Fig. 95: Exposure Start Controlled with an ExFSTrig Signal

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 (for input line Line1 or the GPIO line Line3, if configured for input). Note that such delays are associated with the acquisition start trigger signal and the frame start trigger signal.
- 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 5.11.1 on page 102.
- 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 6.4.3.3 on page 148.

For example, assume that you are using an acA640-120 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 32) + 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 ReadoutTime 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 63.

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 395.

6.12 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 **acA1600-60, acA1920-50 cameras**:

  On the initial wake-up after delivery these cameras have default transport layer settings that do not allow to reach the specified maximum possible frame rate.

  If you want to obtain the maximum possible frame rate, change the values of the default transport layer parameters in pylon Viewer as indicated in Table 54 on page 401.

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 about the Stacked Zone Imaging feature, see Section 9.6 on page 266.

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 the Basler website: www.baslerweb.com
- You can use the Basler pylon API to read the value of the camera's ResultingFrameRateAbs parameter (see the next page).

For more information about

- AOI Height settings, see Section 9.5 on page 261.
- the field output modes on acA750-30 cameras, see Section 6.5 on page 150.

	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 6.11 on page 192.

6.12.1 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 ResultingFrameRateAbs 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 63.

6.12.2 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 **acA1920-25** or **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 9.6 on page 266.

If you are working with an **acA1600-60**, **acA1920-50** camera:

On the initial wake-up after delivery these cameras have default transport layer settings that do not allow to reach the specified maximum possible frame rate.

If you want to obtain the maximum possible frame rate, change the values of the default transport layer parameters in pylon Viewer as indicated in Table 54 on page 401.

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 9.5 on page 261.
- 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 395.

6.12.2.1 Sensor Readout Modes on Certain Cameras

Available for	Not Available for
acA640-300, acA800-200, acA1300-75	acA640-90, acA640-120, acA645-100, acA780-75, acA1280-60, acA1300-22, acA1300-30, acA1300-60 acA1600-20, acA1600-60, acA1920-25, acA1920-40, acA1920-50, acA2000-50, acA2040-25, acA2500-14, acA3800-10, acA4600-7

The camera models with sensor readout mode have two sensor readout modes:

- **normal readout mode**:

  In this mode the camera delivers images at a normal frame rate.
  You can use the normal readout mode if the image quality is important to you and if you want to use different AOI sizes.

- **fast readout mode**:

  In this mode the camera delivers images at higher frame rates.
  The fast readout mode can be used if your application requires higher frame rates.
  If you run the cameras in the fast readout mode and if you change the initial wake-up AOI to another size, there might be artifacts in the images.
  If you want to run the cameras at higher frame rates and if the image quality is important to you, keep the AOI sizes to the wake-up values.

The cameras wake up in the normal readout mode.

Selecting the Sensor Readout Mode

The following code snippet illustrates using the API to set the sensor readout mode and to check what readout mode is set in the camera:

```
camera.SensorReadoutMode.SetValue(SensorReadoutMode_Normal);
```

To check what readout mode is currently set:

```
SensorReadoutModeEnums e = camera.SensorReadoutMode.GetValue();
```

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

6.12.3 Removing the Frame Rate Limit (acA640-120 Only)

Normally, the maximum frame rate that an acA640-120 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 acA640-120 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 max. allowed frame rate:

1. Read the value of the ResultingFrameRateAbs parameter with the maximum frame rate limit enabled.
2. Use the Remove Parameter Limits feature to remove the limit.
3. Read the value of the ResultingFrameRateAbs parameter with the limit removed.

For more information about

- using the Remove Parameter Limits feature, see Section 9.3 on page 255.
- the ResultingFrameRateAbs parameter, see page 196.

6.13 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 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 - TriggerMode for Acquisition and Frame Start Triggers Both Off (Free Run)

Use case one is illustrated on page 202.

In this use case, the AcquisitionMode parameter is set to Continuous. The TriggerMode parameter for the acquisition start trigger and the TriggerMode 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 AcquisitionFrameRateAbs parameter unless the current camera settings result in a lower frame rate. If the AcquisitionFrameRateAbs 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 AcquisitionFrameRateAbs parameter, see Section 6.3.1.1 on page 128.

Use Case: "Free Run" (TriggerMode for acquisition start and for frame start set to Off)
The camera will generate acquisition start trigger signals internally with no action by the user.
The camera will generate frame start trigger signals internally with no action by the user.

Settings: AcquisitionMode = Continuous
TriggerMode for the acquisition start trigger = Off
TriggerMode for the frame start trigger = Off

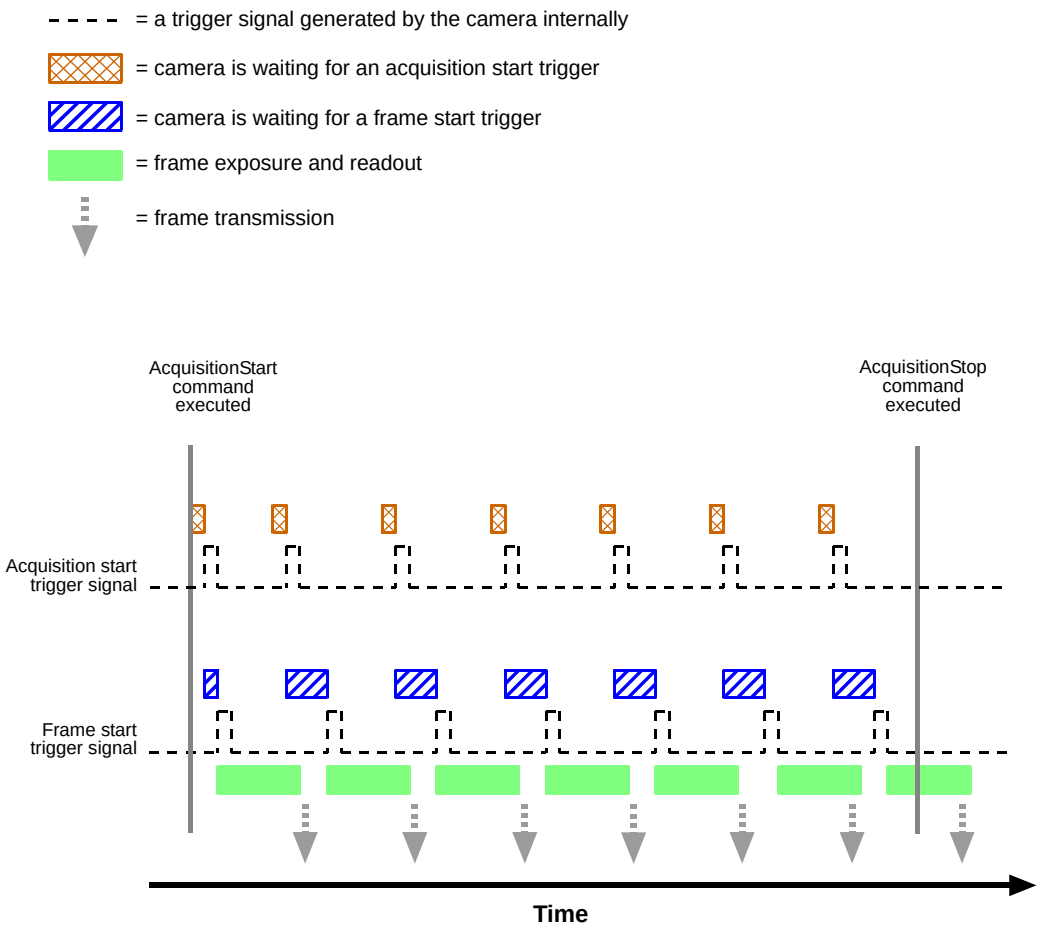


Fig. 96: Use Case 1 - TriggerMode for Acquisition Start Trigger and Frame Start Trigger Set to Off

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

Use case two is illustrated on page 204.

In this use case, the AcquisitionMode parameter is set to Continuous. The TriggerMode parameter for the acquisition start trigger is set to Off and the TriggerMode parameter for the frame start trigger is set to On.

Because the TriggerMode parameter for 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 TriggerMode parameter for 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: TriggerMode for acquisition start set to Off and for frame start set to On

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:
AcquisitionMode = Continuous
TriggerMode for the acquisition start trigger = Off
TriggerMode for the frame start trigger = On
TriggerSource for the frame start trigger = Line1
TriggerActivation for the frame start trigger = RisingEdge

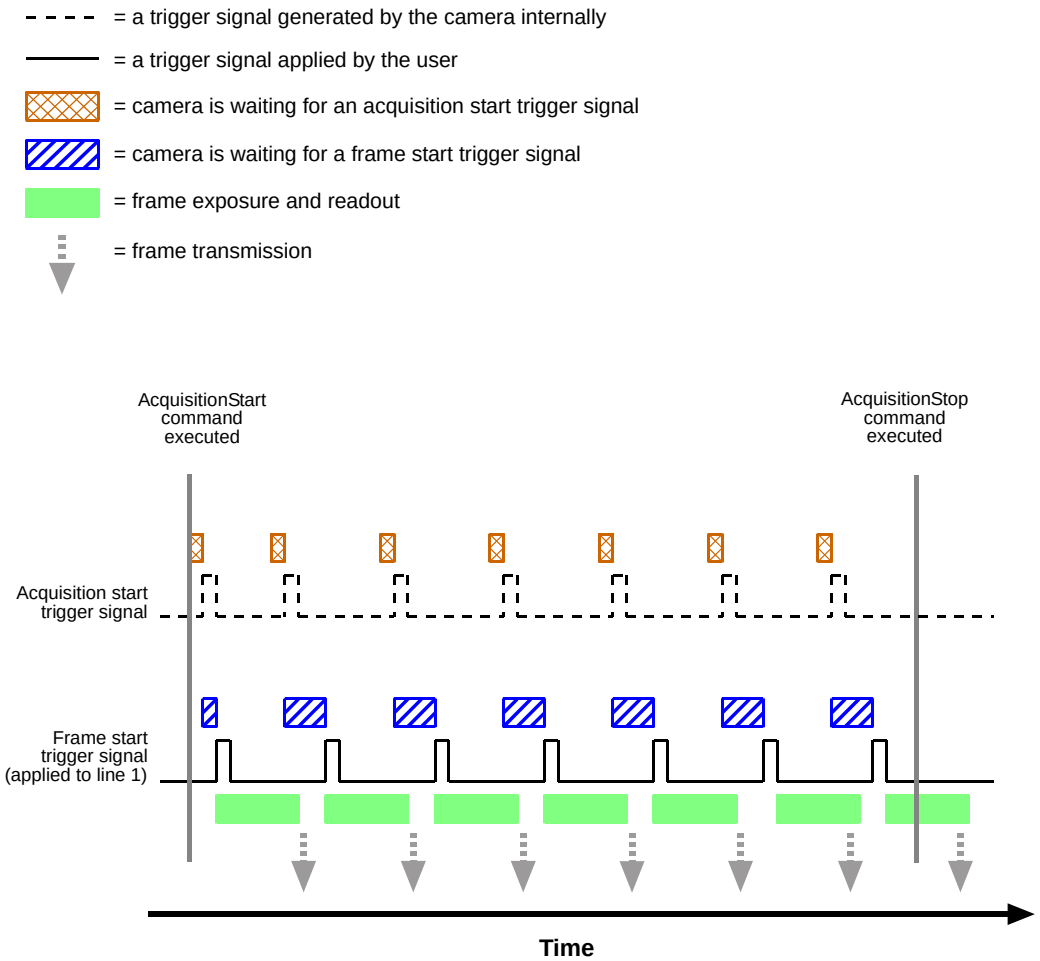


Fig. 97: Use Case 2 - TriggerMode for Acquisition Start Trigger Set to On and for Frame Start Trigger Set to Off

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

Use case three is illustrated on page 206.

In this use case, the AcquisitionMode parameter is set to Continuous. The TriggerMode parameter for the acquisition start trigger is set to On and the TriggerMode parameter for the frame start trigger is set to Off.

Because the TriggerMode parameter for the acquisition start trigger 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 AcquisitionFrameCount 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 TriggerMode parameter for 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 AcquisitionFrameRateAbs parameter. If the AcquisitionFrameRateAbs 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 AcquisitionFrameCount 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 AcquisitionFrameRateAbs parameter, see Section 6.3.1.1 on page 128.

Use Case: TriggerMode for acquisition start to On and for frame start trigger to 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:
AcquisitionMode = Continuous
TriggerMode for the acquisition start trigger = On
TriggerSource for the acquisition start trigger = Line 1
TriggerActivation for the acquisition start trigger = Rising Edge
AcquisitionFrameCount = 3
TriggerMode for the frame start trigger = Off

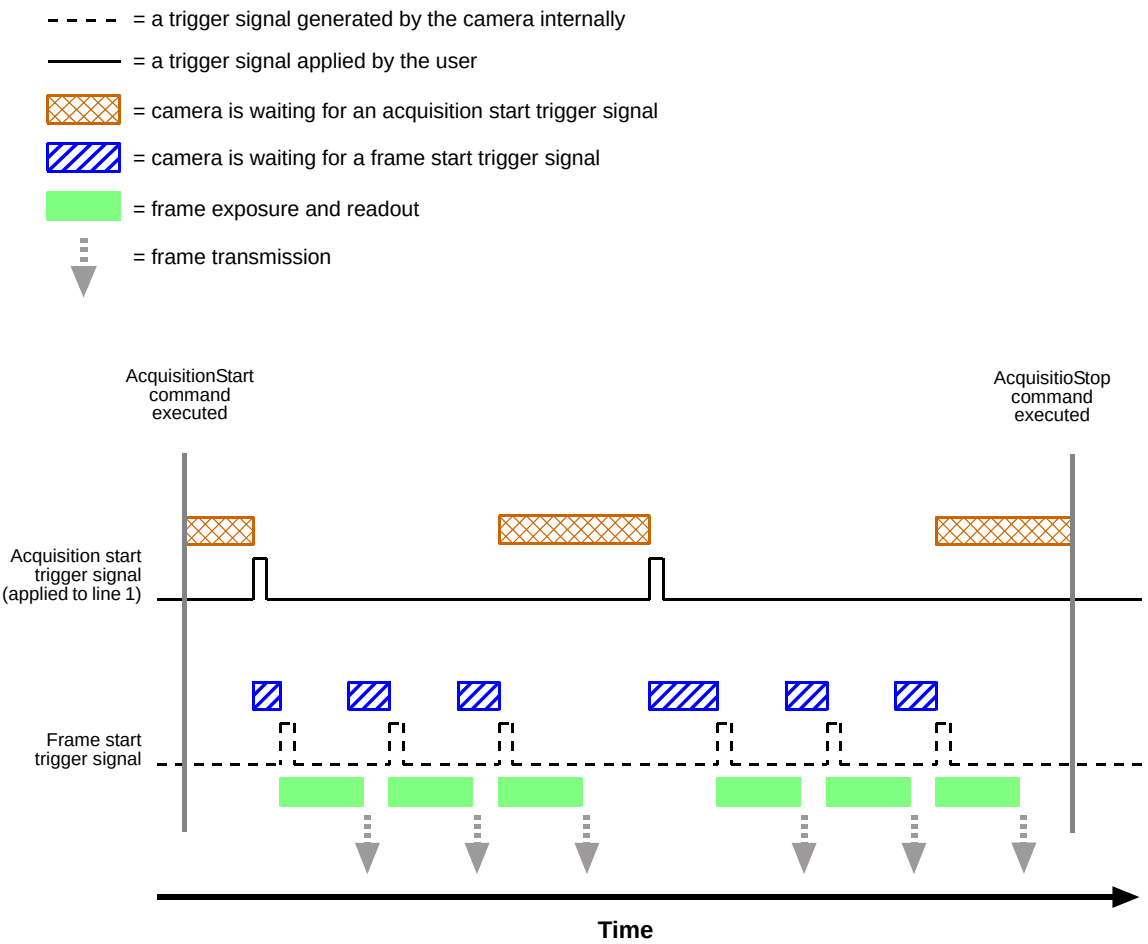


Fig. 98: Use Case 3 - Acquisition Start Trigger On and Frame Start Trigger Off

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

Use case four is illustrated on page 208.

In this use case, the AcquisitionMode parameter is set to Continuous. The TriggerMode parameter for the acquisition start trigger is set to On and the TriggerMode parameter for the frame start trigger is set to On.

Because the TriggerMode parameter for the acquisition start trigger 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 AcquisitionFrameCount 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 AcquisitionFrameCount 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: TriggerMode for acquisition start On and for frame start trigger On

The acquisition start trigger is on, and the TriggerSource for the acquisition start trigger 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:
AcquisitionMode = Continuous
TriggerMode for the acquisition start trigger = On
TriggerSource for the acquisition start trigger = Software
AcquisitionFrameCount = 3
TriggerMode for the frame start trigger = On
TriggerSource for the frame start trigger = Line1
TriggerActivation for the frame start trigger = RisingEdge

= a trigger signal applied by the user

= camera is waiting for an acquisition start trigger signal

= camera is waiting for a frame start trigger signal

= frame exposure and readout

= frame transmission

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

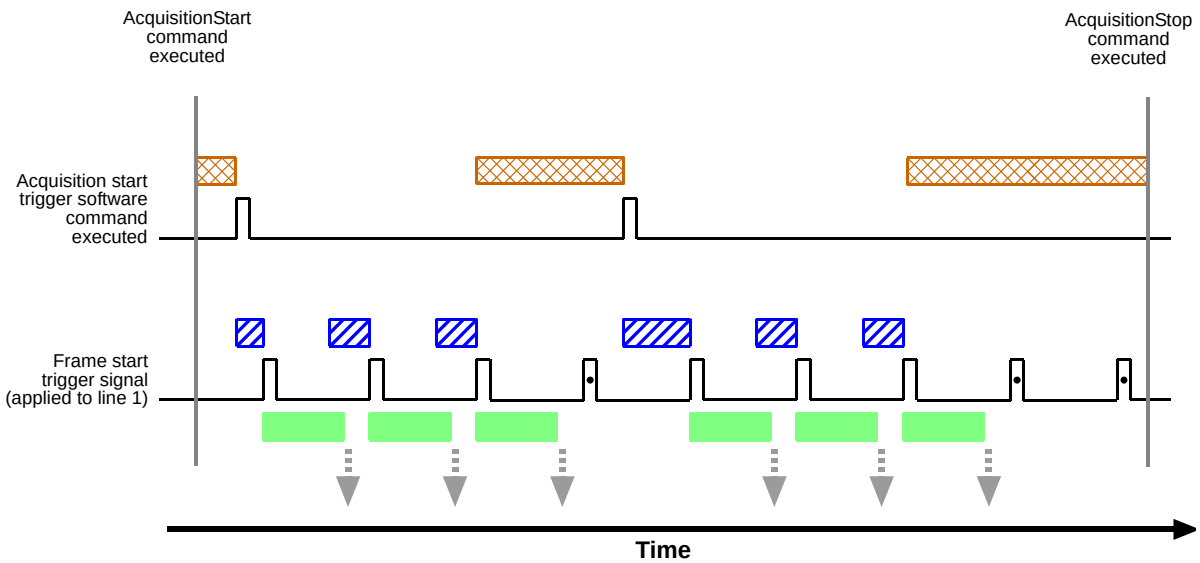


Fig. 99: Use Case 4 - Acquisition Start Trigger On and Frame Start Trigger On

7 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.

7.1 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 100 (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.)

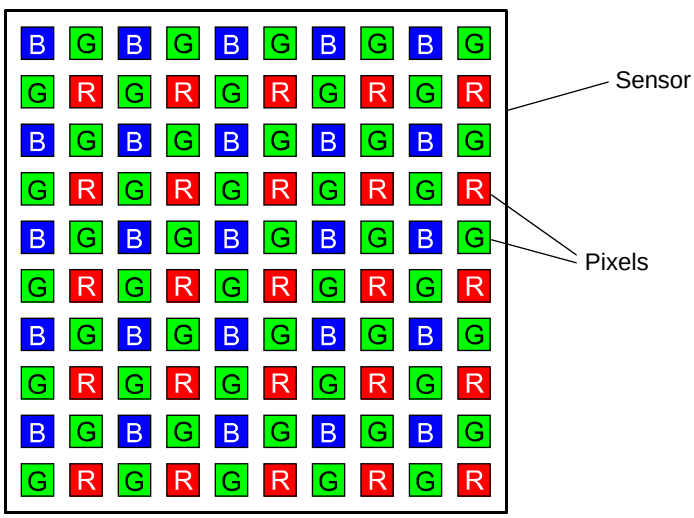


Fig. 100: Bayer Filter Pattern

7.1.1 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 33 shows the filter alignment for each available camera model.

Color Camera Model	Filter Alignment
acA640-90, acA640-120, acA640-300, acA645-100, acA780-75, acA800-200, acA1300-22, acA1300-30, acA1300-75, acA1600-20, acA3800-10, acA4600-7	BG
acA2500-14	GB
acA1280-60, acA1300-60, acA1600-60, acA1920-40, acA1920-50	RG
acA1920-25, acA2000-50, acA2040-25	GR

Table 33: 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, Bayer GB, Bayer RG or Bayer GR (see Table 33). 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 PixelColorFilter 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 most cameras: When either the Reverse X feature or the Reverse Y feature or both are used, the alignment of the color filter to the image remains Bayer BG, Bayer RG, Bayer GB or Bayer GR. The camera includes a mechanism that keeps the filter alignment constant when these features are used (exceptions: see message box below).

Use of mirror imaging features changes Bayer color filter alignment of certain cameras

For camera models (*): acA640-300gc, acA800-200gc, acA1300-75gc, acA1920-40gc, acA1920-50gc

When you configure the cameras mentioned above (see *), take into account that, if you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change.
For more information, see Section 9.12.1 on page 319.

For more information about

- the camera's AOI feature, see Section 9.5 on page 261.
- the Reverse X and Reverse Y features, see Section 9.12 on page 319.

7.1.2 Pixel Data Formats Available on Cameras with a Bayer Filter

Bayer Formats

Cameras equipped with a Bayer pattern color filter can output pixel data in the pixel data formats shown in the tables in Section 1.2 on page 2.

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.

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.

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".)

7.2 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 101.

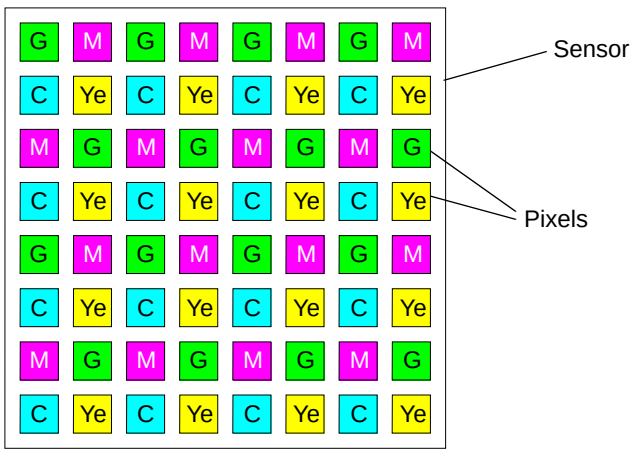


Fig. 101: Complementary Color Filter Pattern

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 102, 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

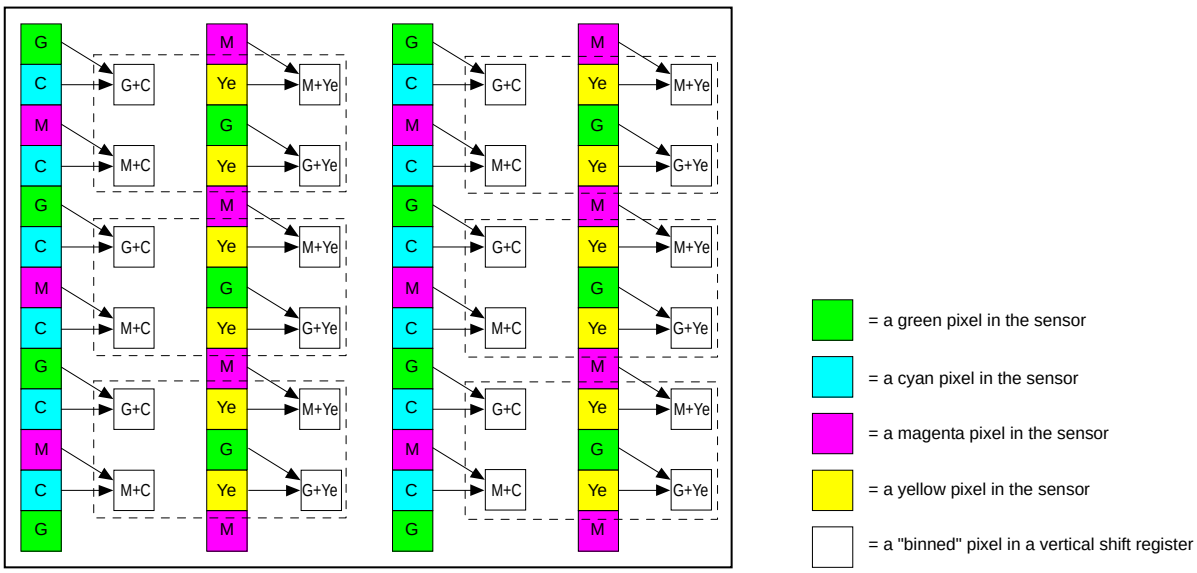


Fig. 102: Color Combinations for Field 0

As shown in Figure 103, 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

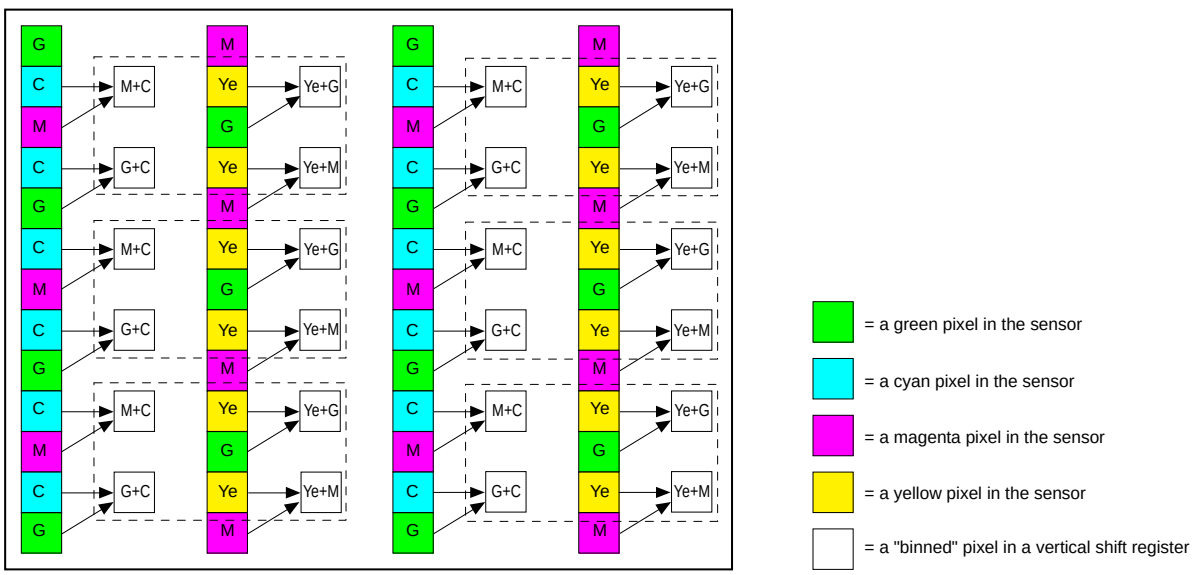


Fig. 103: Color Combinations for Field 1

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:

1. The CMYeG sensor colors are converted into a YUV color signal.
2. A matrix color transformation is performed on the YUV color information to obtain full RGB color information for each binned pixel.
3. If the camera's White Balance feature is used, it will act on the RGB information for each binned pixel.
4. If the camera's Color Adjustment feature is used, it will act on the RGB information for each binned pixel.
5. If the camera's Gamma Correction feature is used, it will act on the RGB information for each binned pixel.
6. A final transformation is performed on the RGB color information to convert it to YUV information for each binned pixel.
7. The binned pixel values are transmitted from the camera in a YUV format.

7.2.1 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.

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".)

7.3 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 49.

7.4 Color Enhancement Features

7.4.1 White Balance

Valid for ...	Not Available for
All color models	acA750-30gc

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.

Only valid for cameras acA1920-25gc and acA2500-14gc:

If color binning is enabled, white balancing will be applied **after** color binning has been performed.

For more information about color binning, see Section 9.9.1 on page 308.

Setting the White Balance

This section describes how a color camera's white balance can be adjusted "manually", i.e., by setting the value of the BalanceRatioAbs 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 BalanceRatioAbs 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 9.14 on page 328.
- the Balance White Auto function, see Section 9.14.8 on page 342.

When you are using matrix color transformation and you set the LightSourceSelector 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 7.4.3 on page 222 and Section 7.4.4 on page 227.

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

The BalanceRatioAbs 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 BalanceRatioAbs parameter for a color:

1. Set the BalanceRatioSelector to red, green, or blue.
2. Set the BalanceRatioAbs parameter to the desired value for the selected color.

You can set the BalanceRatioSelector and the BalanceRatioAbs 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 63.

White Balance Reset

The camera includes a WhiteBalanceReset 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 LightSourceSelector parameter setting.

You can execute the WhiteBalanceReset 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 63.
- the LightSourceSelector parameter, see Section 7.4.3.1 on page 225 or Section 7.4.4.1 on page 229.

7.4.2 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.

	Only valid for cameras acA1920-25gc and acA2500-14gc: If color binning is enabled, gamma correction will be applied **after** color binning has been performed. For more information about color binning, see Section 9.9.1 on page 308.

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 GammaEnable parameter.

You can use the GammaSelector 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 GammaEnable parameter, use the GammaSelector, 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 63.

7.4.3 Matrix Color Transformation on Color Models

Available for	Not Available for
All color models	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.

Only valid for cameras acA1920-25gc and acA2500-14gc:

If color binning is enabled, matrix color transformation will be applied **after** color binning has been performed.

For more information about color binning, see Section 9.9.1 on page 308.

Matrix Color Transformation Parameters

The initial parameter that you must consider when working with the Matrix Color Transformation feature is the ProcessedRawEnable parameter (see * below). If the camera is set to output pixel data in the Bayer xx format, then the ProcessedRawEnable 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 ProcessedRawEnable 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 **ColorTransformationSelector** 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.

(*) For acA640-300, acA800-200, acA1300-75, acA1920-40, and acA1920-50 camera models you don't need the ProcessedRawEnable parameter. As a consequence, this parameter isn't available for these cameras.

The second parameter associated with matrix color transformation is the **LightSourceSelector** 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 **ColorTransformationMatrixFactor** 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 ColorTransformationMatrixFactor 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 ColortransformationMatrixFactorRaw parameter.

When the LightSourceSelector parameter is set to Off or Custom, the ColorTransformationMatrixFactor parameter will not be available.

Setting Matrix Color Transformation

You can set the ProcessedRawEnable, ColorTransformationSelector and LightSourceSelector 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 ProcessedRawEnable parameter value to enabled.

	(*) For acA640-300, acA800-200, acA1300-75, acA1920-40, and acA1920-50 camera models you don't need the ProcessedRawEnable parameter. As a consequence, this parameter isn't available for these cameras.

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 (exception: not for cameras mentioned
// above, see(* above).
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 63.

7.4.3.1 The Custom Light Source Setting

	The "Custom" setting for the LightSourceSelector 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:

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

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 ColorTransformationValueRaw 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 ColorTransformationValueSelector, ColorTransformation Value, and ColorTransformationValueRaw 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 63.

7.4.4 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 ColorTransformationSelector 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 LightSourceSelector 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 ColorTransformationMatrixFactor 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 TransformationMatrixFactor 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 ColortransformationMatrixFactorRaw 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 ColorTransformationSelector and LightSourceSelector 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 63.

7.4.4.1 The Custom Light Source Setting (acA750-30gc)

	The "Custom" setting for the LightSourceSelector 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 ColorTransformationValueSelector to select one of the GainXY positions in the matrix and using the ColortransformationValue 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 ColorTransformationValueRaw 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 ColorTransformationValueSelector, ColorTransformationValue, and ColorTransformationValueRaw 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 63.

7.4.5 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.

For more information about

- the white balance, see Section 7.4.1 on page 217 and
- an overall procedure for setting the color enhancement features, see Section 7.4.6 on page 236

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 7.4.3 on page 222 and Section 7.4.4 on page 227 for more information about color matrix transformation.

Only valid for cameras acA1920-25gc and acA2500-14gc:

If color binning is enabled, color adjustment will be applied **after** color binning has been performed.

For more information about color binning, see Section 9.9.1 on page 308.

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 104) 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 104) 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.

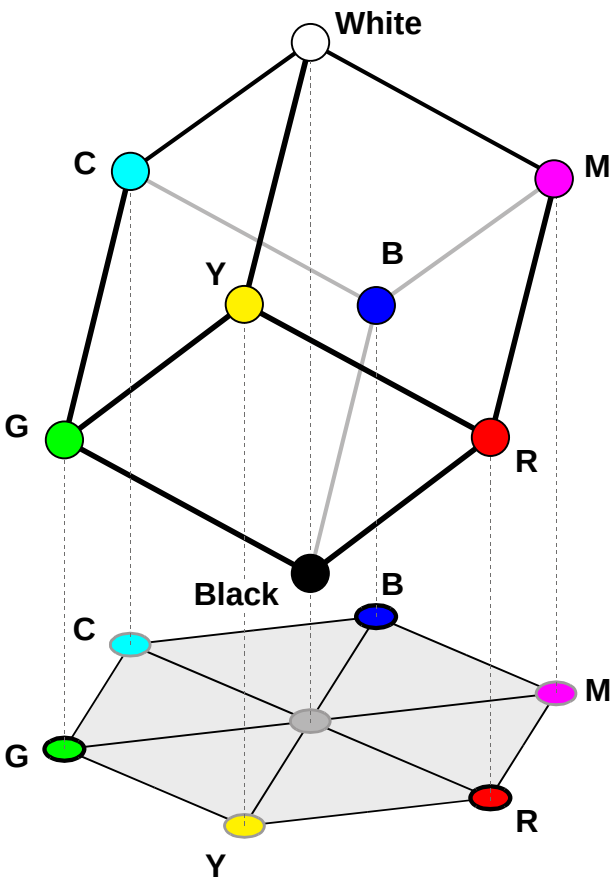


Fig. 104: RGB Color Cube With YCM Secondary Colors, Black, and White, Projected On a Plane

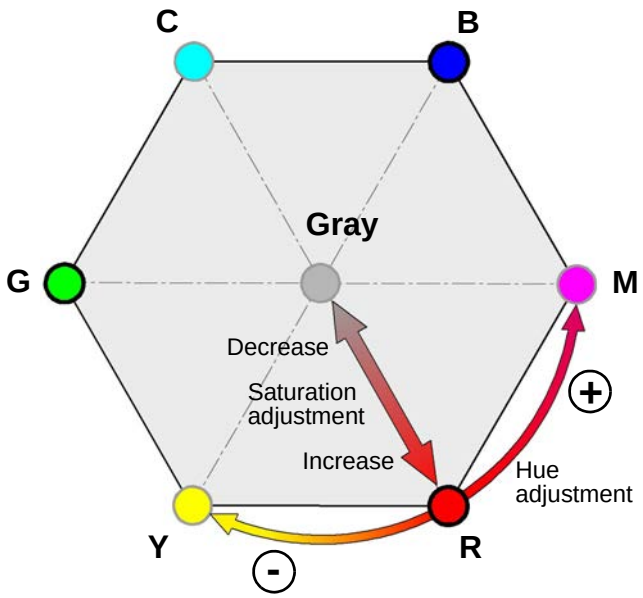


Fig. 105: Hue and Saturation Adjustment In the Color Hexagon. Adjustments Are Indicated for Red as an Exam-

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 ProcessedRawEnable parameter (exceptions: see * below). If you are working with a camera that is set to output pixel data in a Bayer xx format, then the ProcessedRawEnabled 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 ProcessedRawEnable parameter is not set to enabled, the Matrix Color Transformation feature and the Color Adjustment feature will have no effect on the camera operation.

(*) For acA640-300, acA800-200, acA1300-75, acA1920-40, and acA1920-50 camera models you don’t need the ProcessedRawEnable parameter. As a consequence, this parameter isn’t available for these cameras.

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

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

You can use the ColorAdjustmentHue 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 ColorAdjustmentHueRaw 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 ColorAdjustmentSaturation 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 ColorAdjustmentSaturationRaw 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 ProcessedRawEnable (see * above), ColorAdjustmentEnable, ColorAdjustmentSelector, ColorAdjustmentHue, ColorAdjustmentHueRaw, ColorAdjustmentSaturation, and ColorAdjustmentSaturationRaw 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 ProcessedRawEnable 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 (exception: not for cameras mentioned
// above, see(* above).
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 63.

Color Adjustment Reset

The camera includes a ColorAdjustmentReset 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 LightSourceSelector parameter setting.

You can execute the ColorAdjustmentReset 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 63.

7.4.6 A Procedure for Setting the Color Enhancements

When setting the color enhancements on the camera, the procedure for setting the parameters slightly varies, depending on the camera model:

- For camera models **without** GPIO:
  These cameras wake up with a standard parameter configuration with color enhancement parameters **disabled/not set**. For these cameras, color enhancement parameters have to be set or a color factory set has to be loaded and fine-tuned.
- For camera models **with** GPIO:
  These cameras wake up with a standard parameter configuration with color enhancement parameters enabled/set. As for these cameras the basic color enhancement parameters are set by default, you just have to check the image and fine-tune the parameters, if required, to your needs.
  If you want to check what the images look like for these cameras, with disabled color enhancement parameters, you can load the Raw Color factory set.

Since it makes changing camera parameters quick and easy, we recommend using the Basler pylon Viewer software when you are making adjustments.

To check/set the color enhancements:

1. Depending on your camera model the first step varies:
   - For camera models **without** GPIO:
     Load the color factory set into the active set.
   - For camera models **with** GPIO:
     In the standard factory set the color enhancement parameters are **enabled/set**.
     Make sure that the standard factory set is loaded.

   Theses sets contain the basic color enhancement settings with wake-up values.
2. 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.
3. 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.

We recommend including a standard color chart within your camera's field of view when you are adjusting the color enhancements; see next steps. 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).

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

   The color fidelity should now be quite good.

5. 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.

6. If you make changes to the loaded factory set, save the changes to a user set, so that your changes don't get lost. The user set with the changed parameters can be loaded into the active set and designated as the startup set.
   For more information, see Section 9.20 on page 360.

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 recover from this situation

- by using the camera's color adjustment reset command (see page 235).
- Depending on the camera model:
  - camera models **without** GPIO:
    by loading the cameras "color factory set" into the camera's active set.
    See the next section for more information about the camera's color factory set.
  - cameras **with** GPIO:
    by loading the cameras "standard factory set" into the camera's active set.
    In the standard factory set the basic color enhancement parameters are properly adjusted.

For information about which camera model has GPIO, see Section 5.2 on page 74.

7.4.7 Factory Sets

When a camera leaves the factory, it contains several "factory sets" stored in its permanent memory. A factory set is a collection of settings for the parameters needed to operate the camera. Each one of the factory sets is optimized to make the camera perform well in a particular situation.

For more information about the different factory sets and about selecting and loading configuration sets, see Section 9.20 on page 360.

In the following two of the factory sets are explained in detail.

7.4.7.1 The "Color" Factory Set

Available for	Not Available for
All color models (exceptions, see right)	acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

One of the factory sets is known as the "color factory set", and the parameter settings contained in the color factory set are optimized to produce good color images under the most common lighting conditions.

To make the parameters contained in the color factory set become the ones that are actively controlling camera operation, you must load the color factory set into the active set.

When you do this, it will set

- the GammaSelector parameter to sRGB
- the ProcessedRawEnable parameter to enabled.
- the LightSourceSelector parameter to Daylight 5000.
- the white balance parameters to values that are suitable for daylight lighting.

If you have badly mis-adjusted 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 set as the startup set and loading it, is a good way to recover from gross mis-adjustment of the color features.

7.4.7.2 The "Raw Color" Factory Set

Available	Not Available
acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50	All other color models
For information about which camera model has GPIO or not, see Section 5.2 on page 74.	

Another set is known as the "raw color factory set", which has the color enhancement parameters disabled. The raw color factory set makes it possible that you can see what the images will look like without color enhancement settings.

To make the parameters contained in the raw color factory set become the ones that are actively controlling camera operation, you must load the raw color factory set into the active set.

When you do this, the settings will be as follows:

- The GammaEnable parameter is disabled.
- The GammaSelector parameter is set to User.
- The LightSourceSelector parameter is set to Off.
- The ColorAdjustmentEnable parameter is disabled.
- The following parameters are set:
  BalanceRatioRaw to 64, BalanceRatioAbs to 1.0 for red, green and blue.

8 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 information about the available pixel data formats.

For information about the pixel formats available on mono and color cameras, see the "Pixel Data Formats" entries in the corresponding specifications tables in Section 1.2, from page 2 on.

Note

You can find detailed information about the mono and color pixel formats in the Pixel Format Naming Convention, Version 1.1 and above. You can obtain the document from the Automated Imaging Association (AIA).

Some details of the color formats are described in Section 8.2 on page 242.

8.1 Setting Pixel Format Parameter Values

You can set the PixelFormat 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_Mono10);
Camera.PixelFormat.SetValue(PixelFormat_Mono10p);
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_BayerBG10);
Camera.PixelFormat.SetValue(PixelFormat_BayerBG10p);
Camera.PixelFormat.SetValue(PixelFormat_BayerBG12);
Camera.PixelFormat.SetValue(PixelFormat_BayerBG12Packed);
Camera.PixelFormat.SetValue(PixelFormat_BayerGB8);
Camera.PixelFormat.SetValue(PixelFormat_BayerGB10);
Camera.PixelFormat.SetValue(PixelFormat_BayerGB10p);
Camera.PixelFormat.SetValue(PixelFormat_BayerGB12);
Camera.PixelFormat.SetValue(PixelFormat_BayerGB12Packed);
Camera.PixelFormat.SetValue(PixelFormat_BayerGR8);
Camera.PixelFormat.SetValue(PixelFormat_BayerGR10);
Camera.PixelFormat.SetValue(PixelFormat_BayerGR10p);
Camera.PixelFormat.SetValue(PixelFormat_BayerGR12);
Camera.PixelFormat.SetValue(PixelFormat_BayerGR12Packed);
Camera.PixelFormat.SetValue(PixelFormat_BayerRG8);
Camera.PixelFormat.SetValue(PixelFormat_BayerRG10);
```

```
Camera.PixelFormat.SetValue(PixelFormat_BayerRG10p);
Camera.PixelFormat.SetValue(PixelFormat_BayerRG12);
Camera.PixelFormat.SetValue(PixelFormat_BayerRG12Packed);
```

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 63.

8.2 Pixel Data Output Formats: Some Details for Color Cameras

Bayer Formats

Depending on the camera model, the cameras equipped with a Bayer pattern color filter can output color images based on the Bayer pixel formats given in the specifications tables in Section 1.2, from page 2 on.

Depending on the camera model: When a color camera is set for one of these Bayer pixel formats, it outputs 8, 10, or 12 bits of data per pixel and the pixel data is not processed or interpolated in any way. For each pixel covered with a red filter, you get 8, 10, or 12 bits of red data. For each pixel covered with a green filter, you get 8, 10, or 12 bits of green data. And for each pixel covered with a blue filter, you get 8, 10, or 12 bits of blue data. This type of pixel data is sometimes referred to as "raw" output.

Use of mirror imaging features changes Bayer color filter alignment of certain cameras

For some color cameras, provisions are made ensuring that the effective color filter alignment will be constant for both, normal and mirror images.

Exceptions (*): acA640-300gc, acA800-200gc, acA1300-75gc, acA1920-40gc, acA1920-50

When you configure the cameras mentioned above (see *), take into account that, if you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change.
For more information, see Section 9.12.1 on page 319.

For more information about the Bayer filter, see Section 7.1 on page 209.

YUV Formats

Most color cameras with a Bayer filter can output color images based on pixel data in YUV format.

When a color camera is set for this 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 7 on page 209.

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.

Mono 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".

9 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.

9.1 Gain

The camera's gain setting is adjustable. As shown in Figure 106, 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.

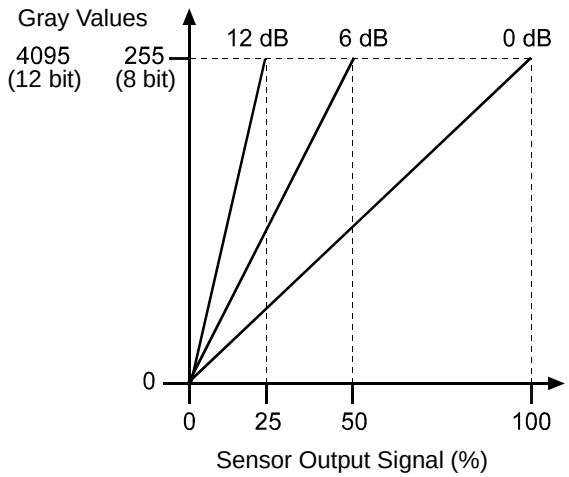


Fig. 106: Gain in dB

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.

9.1.1 Setting the Gain

This section (Section 9.1) describes how gain can be adjusted "manually", i.e., by setting the value of the GainRaw parameter.

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

For more information about

- auto functions in general, see Section 9.14 on page 328.
- the Gain Auto function, see Section 9.14.4 on page 335.

The camera's gain is determined by the value of the GainRaw parameter. GainRaw is adjusted on an integer scale. The minimum setting varies depending on the camera model and on whether vertical binning is enabled (see Table 34). The maximum setting depends on whether the camera is set for a pixel data format that yields

- 8 bit effective pixel depth (e.g. Mono 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed) or
- an effective pixel depth of 12 bits per pixel (e.g. 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 (10-bit depth)	Max Setting (12-bit depth)
acA640-90gm/gc (*)	190	100	1023	-	400
acA640-120gm/gc (*)	100	100	1023	-	600
acA640-300gm/gc	155	155	618	618	-
acA645-100gm/gc (*)	200	140	1023	-	400
acA750-30gm/gc	0	NA	1023	-	600
acA780-75gm/gc (*)	250	140	1023	-	400
acA800-200gm/gc	155	155	618	618	-
acA1280-60gm/gc (*)	See page 250.				
acA1300-22gm/gc acA1300-30gm/gc (*)	300	200	850	-	400
acA1300-60gm/gc/NIR (*)	See page 250.				
acA1300-75gm/gc	155	155	618	618	-
acA1600-20gm/gc (*)	230	220	850	-	400
acA1600-60gm/gc (*)	See page 250.				
acA1920-25gm/gc	0	0	63	-	63

Table 34: Minimum and Maximum Allowed Gain Raw Settings

Camera Model	Min Setting	Min Setting with Vertical Binning (mono cameras)	Max Setting (8-bit depth)	Max Setting (10-bit depth)	Max Setting (12-bit depth)
acA1920-40gm/gc	0	0	360	-	240
acA1920-50gm/gc	0	0	360	-	240
acA2000-50gm/gc/gmNIR (*) acA2040-25gm/gc/gmNIR (*)	36	33	512	-	512
acA2500-14gm/gc	0	0	63	-	63
acA3800-10gm/gc	33	33	255	-	255
acA4600-7gc	33	33	255	-	255
* On these cameras, the minimum setting for the GainRaw parameter can be reduced to 0 by using the Remove Parameter Limits feature. For more information about the Remove Parameter Limits feature, see Section 9.3 on page 255. NA = Not available					

Table 34: Minimum and Maximum Allowed Gain Raw Settings

To set the GainRaw parameter value:

1. Set the GainSelector to GainAll.
2. Set the GainRaw parameter to your desired value.

You can set the GainSelector and the GainRaw 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 63.

If you know the current decimal setting for the GainRaw, you can use the corresponding formula from Table 35 to calculate the dB of gain that will result from that setting:

Camera Model	Formula for Calculating db of Gain	Notes
acA640-90gm/gc acA640-120gm/gc acA645-100gm/gc acA750-30gm/gc acA780-75gm/gc acA1300-22gm/gc acA1300-30gm/gc acA1600-20gm/gc	$Gain_{dB}$ = 0.0359 x GainRaw Setting	Example: GainRaw setting of 200. $Gain_{dB}$ = 0.0359 x 200 = 7.2
acA1280-60gm/gc acA1300-60gm/gc acA1600-60gm/gc	Settable: analog or digital gain **Analog gain** See information on page 250. **Digital gain**: ▪ Digital GainRaw Setting: 0 - 31 Gain in dB = 20 log10 (1 + ((Digital GainRaw Setting * 2) /64)) ▪ Digital GainRaw Setting: 31 - 95 Gain in dB = 20 $\log_{10}$ (2 * (1 + ((Digital GainRaw Setting - 32) /64)))	--
acA640-300, acA800-200, acA1300-75	Gain dB = 20 $\log_{10}$ (GainRaw Setting / 128)	--
acA1920-50gm/gc acA1920-40gm/gc,	Gain dB = 0.1 x GainRaw	Example: Gain raw setting of 200. $Gain_{dB}$ = 0.01 x 200 = 2
acA2000-50gm/gc/gmNIR acA2040-25gm//gc/gmNIR acA3800-10gm/gc acA4600-7gc	Gain in dB = 20 $\log_{10}$ (GainRaw Setting / 32)	Example: GainRaw setting of 128. Gain in dB = 20 $\log_{10}$ (128 / 32) Gain in dB = 12.0

Table 35: Calculating dB of Gain

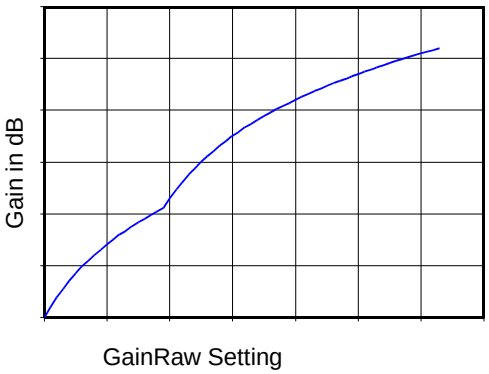
Camera Model	Formula for Calculating db of Gain	Notes
acA1920-25gm/gc acA2500-14gm/gc	The camera's gain is determined by the value of the GainRaw parameter. GainRaw is adjusted on an integer scale. The minimum setting is 0 and the maximum setting is 63. At a setting of 0, the camera's gain will be 0 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 107 shows the gain in dB that will be yielded for each GainRaw parameter setting.  Fig. 107: Gain in dB Yielded by GainRaw Settings	

Table 35: Calculating dB of Gain

The following table shows the minimum and maximum possible dB of gain for each camera mode.

Camera Model	dB Gain at Min Setting	dB Gain at Max Setting (8-bit depth)	dB Gain at Max Setting (10-bit depth)	dB Gain at Max Setting (12-bit depth)
acA640-90gm/gc	6.8	36.7	-	14.4
acA640-120gm/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
acA1280-60gm/gc acA1300-60gm/gc/NIR acA1600-60gm/gc	See page 250.			
acA1300-22gm/gc acA1300-30gm/gc	10.8	30.5	-	14.4
acA1600-60gm/gc	8.3	30.5	-	14.4
acA1920-25gm/gc	0	26.0	-	26.0

Table 36: Minimum and Maximum dB of Gain

Camera Model	dB Gain at Min Setting	dB Gain at Max Setting (8-bit depth)	dB Gain at Max Setting (10-bit depth)	dB Gain at Max Setting (12-bit depth)
acA1920-40gm/gc	0	36.0	-	24.0
acA1920-50gm/gc	0	36.0	-	24.0
acA640-300gm/gc, acA800-200gm/gc, acA1300-75gm/gc	0	12.0	12.0	-
acA2000-50gm/gc	1.02	24	-	24
acA2000-50gmNIR				
acA2040-25gm/gc				
acA2040-25gmNIR				
acA2500-14gm/gc	0	26.0	-	26.0
acA3800-10gm/gc	0.27	18.03	-	18.03
acA4600-7gc				

Table 36: Minimum and Maximum dB of Gain

acA1280-60, acA1300-60, and acA1600-60 Only

Via the GainSelector you can determine whether you want to use the analog or digital gain settings of the camera.

The camera's gain is determined by the value of the GainRaw parameter. GainRaw is adjusted on an integer scale. The minimum setting is 0 and the maximum setting is

- 3 (for the analog gain) and
- 95 (for the digital gain).

Analog Gain

Camera Model	Min Setting	Min Setting with Vertical Binning (mono cameras)	Max Setting (12-bit depth)
acA1300-60gm/gc	0	0	3
acA1300-60gmNIR			
acA1600-60gm/gc	1	1	3

Table 37: Minimum and Maximum Allowed GainRaw Settings (Analog Gain)

Camera Model	Analog Gain / Raw Setting	Analog Gain / dB
acA1280-60gm/gc acA1300-60gm/gc acA1300-60gmNIR acA1600-60gm/gc	0	0.0
	1	3.5
	2	6.0
	3	9.5

Table 38: Examples of Analog Gain Settings and their Gain

Digital Gain

If you know the current decimal setting for the GainRaw, you can use the formulas in Table 35 on page 247 to calculate the dB of gain that will result from that setting:

At a **digital gain** setting of 0, the camera's digital gain will be 0 dB. At a setting of 95, the gain is approximately 12 dB.

Camera Model	Min Setting	Min Setting with Vertical Binning (mono cameras)	Max Setting (12-bit depth)
acA1280-60gm/gc	0	0	95
acA1300-60gm/gc			
acA1300-60gmNIR			
acA1600-60gm/gc			

Table 39: Minimum and Maximum Allowed GainRaw Settings (Digital Gain)

To ensure a good image quality the factory limit for the analog gain is normally from 0 to 3.

For special camera uses, however, it may be helpful to set parameter values outside of the factory limits.

If required, you can use the Remove Parameter Limits feature for the gain to enlarge the gain range. For information on the Remove Parameter Limits feature, see Section 9.3 on page 255.

9.2 Black Level

Adjusting the camera's black level will result in an offset in the digital values output for the pixels:

- **Increasing** the black level setting will result in a **positive** offset.
- **Decreasing** the black level setting will result in a **negative** offset.

Interaction between black level and color enhancement feature

The calibration of the color enhancement is done with a fixed black level parameter value. This fixed black level parameter is the set black level wake-up value. If you want to use the color enhancement feature, the black level parameter must be set to its wake-up value.

The offset depends on the camera model; see the following examples.

Valid for ...
acA640-90, acA640-120, acA645-100, acA750-30, acA780-75, acA1300-22, acA1300-30, acA1600-20

Effective Bit Depth (*)	BlackLevel Parameter	Resulting Offset (Brightness Value for Pixels in Camera)
8 bit	+64 - 64	+ 1 - 1
12 bit	+4 -4	+1 -1
Range: 0 - 1023 (*) For 8 bit: Mono 8, Bayer BG 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed For 12 bit: Mono 12, Bayer 12 BG		

Table 40: Effect of Increasing or Decreasing the BlackLevel Parameter

Valid for ...
acA1920-25, acA1920-40, acA1920-50, acA2000-50, acA2040-25, and acA2500-14, acA3800-10, acA4600-7

Effective Bit Depth (*)	Black Level Parameter (Range: 0 - 511)	Resulting Offset (Brightness Value for Pixels in Camera)
8 bit	+ 16 - 16	+ 1 - 1

Table 41: Effect of Increasing or Decreasing the BlackLevel Parameter II

Effective Bit Depth (*)	Black Level Parameter (Range: 0 - 511)	Resulting Offset (Brightness Value for Pixels in Camera)
12 bit	+1 -1	+1 -1
Range: 0 - 511 (*) xx: either GB or GR For 8 bit: Mono 8, Bayer xx 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed For 12 bit: Mono 12, Mono 12 Packed, Bayer xx 12, Bayer xx 12 Packed)		

Table 41: Effect of Increasing or Decreasing the BlackLevel Parameter II

Valid for ...
acA640-300, acA800-200, acA1300-75, acA1280-60, acA1300-60, acA1600-60

Effective Bit Depth (*)	Black Level Parameter (Range: 0 - 255)	Resulting Offset (Brightness Value for Pixels in Camera)
8 bit	+4 - 4	+ 1 - 1
10 bit	+1 - 1	+ 1 - 1
12 bit	+1 - 1	+ 1 - 1
Range: 0 - 255 (*) xx: either RG, GB or GR for 8 bit: Mono 8, Bayer RG 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed for 10 bit: Mono 10, Mono 10p, Bayer xx 10, Bayer xx 10p for 12 bit: Mono 12, Mono 12 Packed, Bayer xx 12, Bayer xx 12 Packed [12-bit image data based on 10-bit sensor data.]		

Table 42: Effect of Increasing or Decreasing the BlackLevel Parameter III

9.2.1 Setting the Black Level

The black level can be adjusted by changing the value of the BlackLevelRaw parameter.
The range of the allowed settings for the BlackLevelRaw parameter value varies by camera model as shown in Table 43.

Camera Model	Min Allowed Black Level Raw Setting	Max Allowed Black Level Raw Setting
acA640-90gm/gc, acA640-120gm/gc acA645-100gm/gc acA750-30gm/gc, acA780-75gm/gc acA1300-22gm/gc; acA1300-30gm/gc acA1600-20gm/gc acA3800-10gm/gc acA4600-7gc	0	1023
acA1280-60gm/gc, acA1300-60gm/gc/gmNIR, acA1600-60gm/gc	0	80
acA640-300gm/gc acA800-200gm/gc acA1300-75gm/gc acA2000-50gm/gc/gmNIR acA2040-25gm/gc/gmNIR	0	255
acA1920-25gm/gc; acA2500-14gm/gc	0	63
acA1920-40gm/gc acA1920-50gm/gc	0	511

Table 43: BlackLevelRaw Parameter Range

To set the BlackLevelRaw parameter value:

1. Set the BlackLevelSelector to BlackLevelAll.
2. Set the BlackLevelRaw parameter to your desired value.

You can set the BlackLevelSelector and the BlackLevelRaw 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);
```

9.3 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 (exceptions, see *)

  Removing the parameter limits on the gain feature
  - will remove the lower and the upper limit.
  - will increase the gain range.

  For those cameras where the lower limit is already 0, removing the limits has no effect.
- The maximum allowed frame rate on acA640-120 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.

(*) Exceptions: For these camera models the parameter limits can't be removed for the gain feature: acA1920-25, acA2500-14

For more information about

- the Gain feature, see Section 9.1 on page 244.
- the frame rate limit on acA640-120 cameras, see Section 6.12.3 on page 200.

Removing Parameter Limits

To remove the limits for a parameter:

1. Use the ParameterSelector to select the parameter whose limits you want to remove.
2. Set the value of the RemoveLimits parameter.

You can set the ParameterSelector and the value of the RemoveLimits 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 63.

9.4 Digital Shift

Available for	Not available for
acA640-90, acA640-120, acA645-100, acA750-30, acA780-75, acA1280-60, acA1300-22, acA1300-30, acA1300-60 acA1600-20, acA1600-60, acA2000-50, acA2040-25, acA3800-10, acA4600-7	acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

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 pixel values of the camera as indicated in the table below, thus increasing the brightness in the image.

If you cannot see details in dark image areas, for example, you can use the Digital Shift feature to increase the brightness in these dark image areas (and in the rest of the image).

Digital Shift by	Means that the ...	Multiplication of the pixel values by
1	least significant bit is set to 0	2
2	least 2 significant bits are set to 0	4
3	least 3 significant bits are set to 0	8
4	least 4 significant bits are set to 0	16

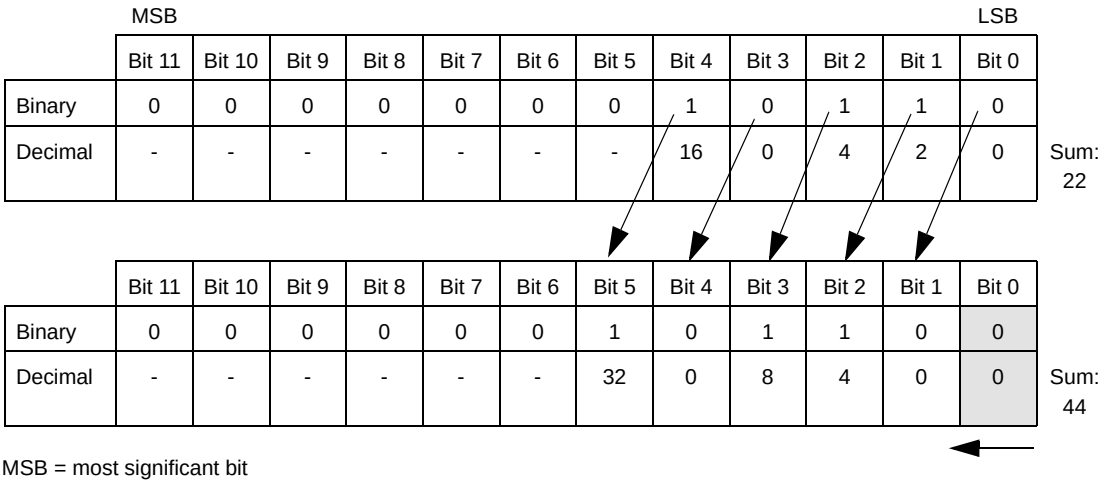
When the least significant

- bits are set to 0, the values from these bits will be shifted to the next most significant bits.
- bit is set to 0, no odd gray values can be output and the gray value scale will only include values of 2, 4, 6, 8, 10, and so on.

Depending on the camera model, the cameras have either a 12-bit or a 10-bit ADC to digitize the output.
In the following, some examples are shown to explain the functional principle.

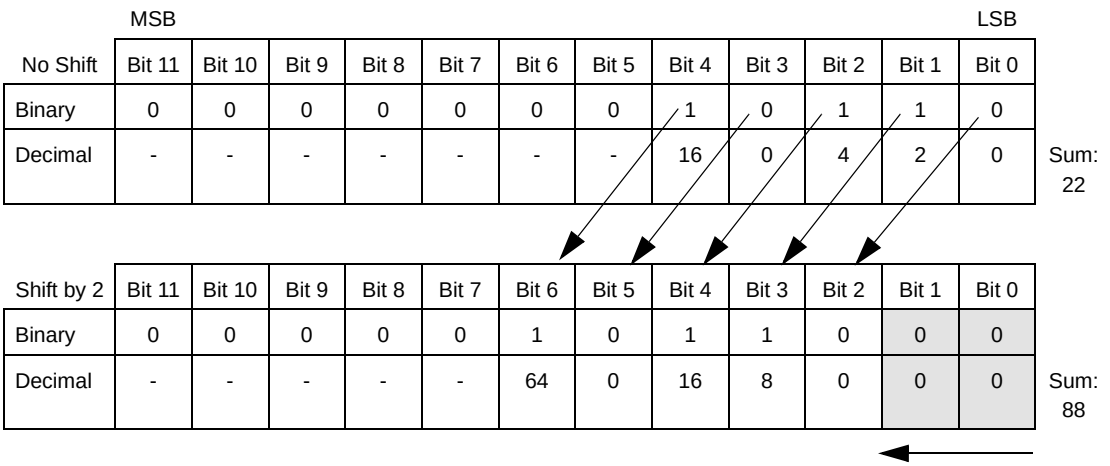
Shift Examples: 12-bit ADC Digitizing a 12-bit Pixel Format

Shift by 1 - multiplication by 2



MSB = most significant bit
LSB = least significant bit

Shift by 2- multiplication by 4



MSB = most significant bit
LSB = least significant bit

Shift Example: 12-bit ADC Digitizing an 8-bit Pixel Format

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, new: bit 7 through bit 0).

Shift by 1 - multiplication by 2

	MSB											LSB	
No Shift	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	~~Bit 3~~	~~Bit 2~~	~~Bit 1~~	~~Bit 0~~	
Binary	0	0	0	1	0	1	1	1	(*)	(*)	(*)	(*)	(*) Dropped
Decimal	-	-	-	16	0	4	2	1	-	-	-	-	Sum: 23
Shift by 1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	~~Bit 3~~	~~Bit 2~~	~~Bit 1~~	~~Bit 0~~	
Binary	0	0	1	0	1	1	1	0	(*)	(*)	(*)	(*)	
Decimal	-	-	32	0	8	4	2	0	-	-	-	-	Sum: 46

MSB = most significant bit
LSB = least significant bit

High Values

If the resulting sum of the digital shift is bigger than the maximum value of the n-bit word, **all** bits will automatically be set to 1, i.e. to the maximum brightness in the image.

Example for 12-bit output:

	MSB											LSB	
No Shift	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Binary	1	0	1	1	0	0	0	1	0	1	1	1	
Decimal	2048	0	512	256	0	0	0	16	0	4	2	1	Sum: 2839

If digital shift was applied, the resulting sum would be 1582. Therefore, all bits set to 1.

Shift by 1	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Binary	1	1	1	1	1	1	1	1	1	1	1	1	
Decimal	2048	1024	512	256	128	64	32	16	8	4	2	1	Sum: 4095

9.4.1 Enabling and Setting Digital Shift

You can enable or disable the Digital Shift feature by setting the value of the DigitalShift parameter.

When the parameter is

- set to zero, digital shift will be disabled.
- 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 DigitalShift 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 63.

9.5 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 108.

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 108.

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.

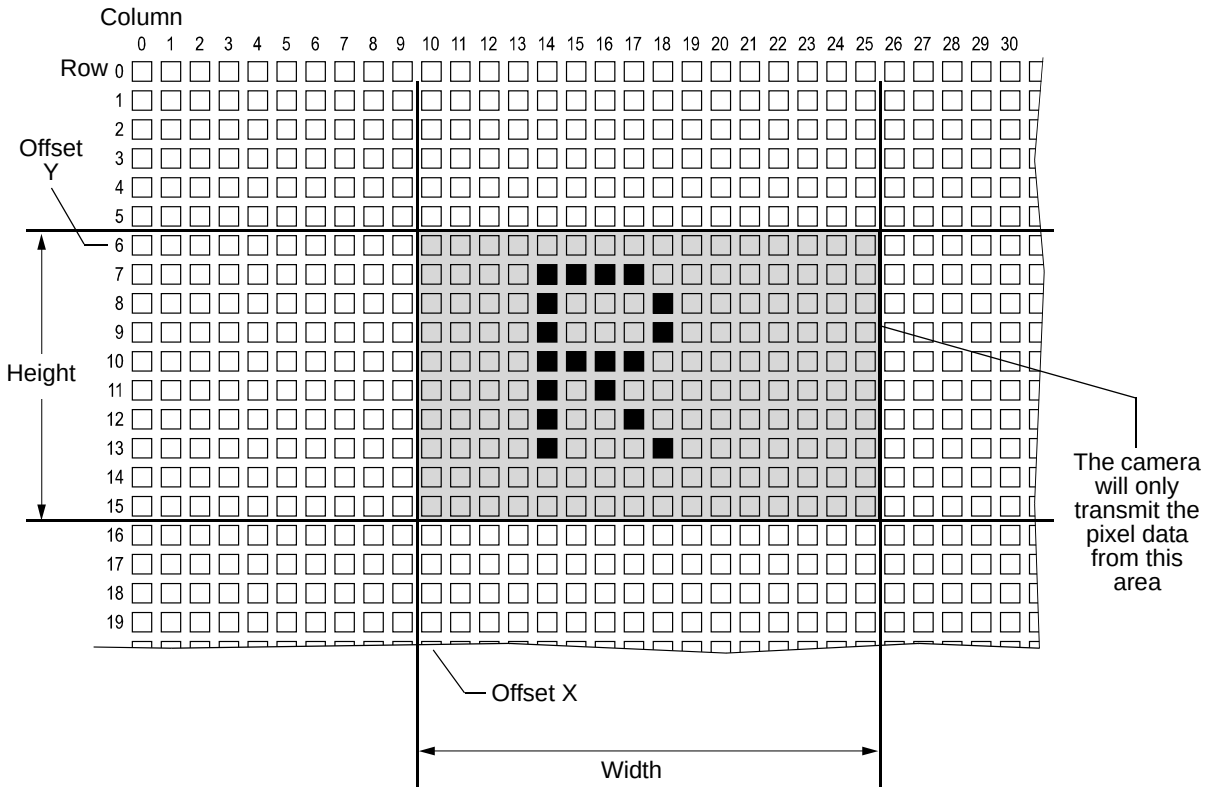


Fig. 108: Area of Interest

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 6.12 on page 196.

If you want to set

- offset X,
  make sure the Center X feature for automatic AOI centering is disabled.
- offset Y,
  make sure the Center Y feature for automatic AOI centering is disabled.

For more information about automatic AOI centering and the effects on Center X and Center Y settings, see Section 9.5.1 on page 265.

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 OffsetX, OffsetY, Width, and Height parameters.

- Offset X: determines the starting column for the area of interest.
- Offset Y: determines the starting row for the area of interest.
- Width: determines the width of the area of interest.
- Height: determines the height of the area of interest.

When you are setting the camera's area of interest (AOI), you must follow these guidelines:

Valid for All Camera Models	
Offset X + AOI width < Width of camera sensor	Example: acA640-120gm: Sum of Offset X + Width < 659.
Offset Y + AOI height < Height of camera sensor	Example: acA640-120gm: Sum of Offset Y+ Height < 494.

Valid for	AOI Parameters		Example
All other camera models (exceptions, see below)	OffsetX OffsetY	**Mono** cameras: Can be set in increments of 1.	0, 1, 2, 3, 4, 5, etc.
		Color cameras: ▪ Can be set in increments of 2. ▪ Must be set to an even number.	0, 2, 4, 6, 8, etc.
	Width Height	**Mono** cameras: Can be set in increments of 1.	1, 2, 3, 4, 5, etc.
		Color cameras: ▪ Can be set in increments of 2. ▪ Must be set to an even number.	2, 4, 6, 8, etc.

Valid for	AOI Parameters		Example
acA750	OffsetX OffsetY	**Mono** and **color** cameras: ▪ Can be set in increments of 2. ▪ Must be set to an even number.	0, 2, 4, 6, 8, etc.
	Width Height	**Mono** cameras: ▪ Can be set in increments of 4. ▪ Must be set to an even number. ▪ Minimum value is 4.	4, 8, 12, 16, etc.
		Color cameras: ▪ Can be set in increments of 4. ▪ Minimum value is 4.	4, 8, 12, 16, etc.
acA640-300, acA800-200, acA1300-75	OffsetX	**Mono** and **color** cameras: Can be set in increments of 16.	0, 16, 32, 48, etc.
	OffsetY	**Mono** cameras: Can be set in increments of 1.	0, 1, 2, 3, 4, etc.
		Color cameras: Can be set in increments of 2.	0, 2, 4, 6, etc.
	Width	**Mono** and **color** cameras: ▪ Can be set in increments of 16. ▪ Minimum value is 256.	256, 272, 288, etc.
	Height	**Mono** cameras: ▪ Can be set in increments of 1. ▪ Minimum value is 1.	1, 2, 3, 4, etc.
		Color cameras: ▪ Can be set in increments of 2. ▪ Minimum value is 2.	2, 4, 6, 8, etc.
acA1920-40, acA1920-50	OffsetX OffsetY	**Mono** and **color** cameras: ▪ Can be set in increments of 2. ▪ Must be set to an even number.	0, 2, 4, 6, 8, etc.
	Width Height	**Mono** cameras: ▪ Can be set in increments of 1. ▪ Minimum value is 1.	1, 2, 3, 4 etc.
		Color cameras: ▪ Can be set in increments of 2. ▪ Minimum value is 2.	2, 4, 6, 8, etc.
acA1920-25, acA2500-14	Minimum AOI size for width and height: 64		

	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 or decimation is enabled, these parameters are set in terms of "virtual" columns and rows. For more information, see Section 9.9.3 on page 310.

You can set the OffsetX, OffsetY, 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 OffsetX, OffsetY, 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 63.

9.5.1 Center X and Center Y

The AOI feature also includes Center X and Center Y capabilities for horizontal and vertical centering. When Center X is enabled, the camera will automatically center the AOI along the sensor's X axis. When Center Y is enabled, the camera will automatically center the AOI along the sensor's Y axis.

When CenterX is enabled, the OffsetX setting is adjusted accordingly and becomes read only.

Note: When CenterX is disabled, the original OffsetX setting that applied when CenterX was enabled, will not be automatically restored. If you want to return to the original OffsetX setting, you will have to do so "manually".

The OffsetY setting behaves analogously when CenterY is enabled and disabled.

Enabling AOI Centering

You can enable AOI centering from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to enable automatic AOI centering:

```
camera.CenterX.SetValue(true);
camera.CenterY.SetValue(true);
```

9.5.2 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 OffsetX and OffsetY parameters. Changes to the AOI size are not allowed on-the-fly.

9.6 Stacked Zone Imaging

Available for	Not Available for
acA2000-50, acA2040-25	All other models

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 StackedZoneImagingEnable parameter is used to enable or disable stacked zone imaging. When the parameter is set to true, stacked zone imaging is enabled.

The OffsetX 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 109 on page 267. In the figure, OffsetX 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 StackedZoneImagingIndex parameter to select a zone number and then using the StackedZoneImagingZoneEnable parameter to enable the selected zone.
At least one zone must be enabled.

Once a zone has been enabled, you must use the StackedZoneImagingZoneOffsetY parameter to set the offset (in pixels) between the top of the sensor and the top of the zone. And you can use the StackedZoneImagingZoneHeight parameter to set the height of the zone.

In Figure 109, 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 StackedZoneImagingZoneOffsetY parameter is set to 6
- The StackedZoneImagingZoneHeight parameter is set to 6.

For zone 2:

- The StackedZoneImagingZoneOffsetY parameter is set to 20
- The StackedZoneImagingZoneHeight parameter is set to 10.

For zone 3:

- The StackedZoneImagingZoneOffsetY parameter is set to 38.
- The StackedZoneImagingZoneHeight 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.

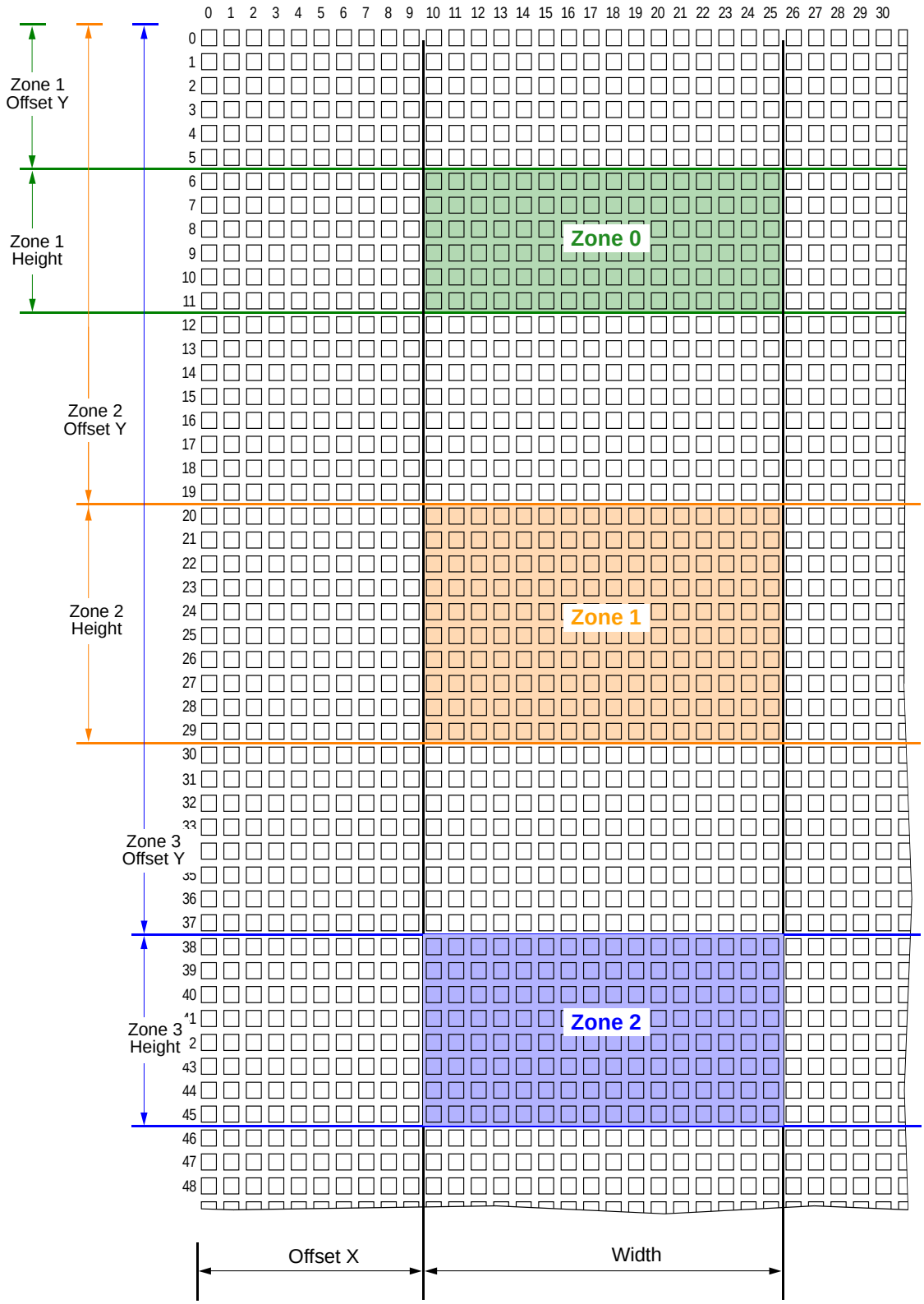


Fig. 109: Stacked Zone Imaging

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
OffsetX (valid for all zones)	10	5	3
Width (valid for all zones)	16	8	5
OffsetY	6	3	2
Height	6	3	2

Table 44: 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 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 following parameters become read only:

- OffsetY: parameter indicates the Y offset for the zone nearest to the top of the sensor.
- Height: 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).

9.6.1 Setting Stacked Zone Imaging

Guidelines

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

<table>
<tr><th colspan="3">Available for acA2000-50, acA2040-25</th></tr>
<tr><td>Offset X + Stacked zone imaging zone width
< Width of camera sensor</td><td colspan="2">Example: acA2000-50gm:
Sum of Offset X + Width < 2048.</td></tr>
<tr><td>Offset Y + Stacked zone imaging zone height
< Height of camera sensor</td><td colspan="2">Example: acA2000-50gm:
Sum of Offset Y+ Height < 1088.</td></tr>
<tr><td rowspan="2">Offset X
Offset Y
Width
Height</td><td>Mono cameras:
Can be set in increments of 1</td><td>Example:
1, 2, 3, 4, 5, etc.</td></tr>
<tr><td>Color cameras:
▪ Can be set in increments of 2
▪ Must be set to an even number</td><td>Example:
0, 2, 4, 6, 8, etc.</td></tr>
</table>

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 63.

9.7 Error Codes

The camera can detect several user correctable errors. If one of these errors is present, the camera will set an error code.

The following table indicates the available error codes:

Code	Condition	Meaning
0	No Error	The camera has not detected any errors since the last time when the error memory was cleared.
1	Overtrigger	An overtrigger has occurred. ▪ The user has applied an acquisition start trigger to the camera when the camera was not in a waiting for acquisition start condition. Or: ▪ The user has applied a frame start trigger to the camera when the camera was not in a waiting for frame start condition.
2	User set load	An error occurred when attempting to load a user set. Typically, this means that the user set contains an invalid value. Try loading a different user set.
3	Invalid Parameter *	A parameter is set out of range or in an otherwise invalid manner. Typically, this error only occurs when the user is setting parameters via direct register access.
4	Over temperature	Only available on the camera models indicated below (*). The camera goes into the over-temperature idle mode when the internal temperature of 80 °C (+176 °F) is reached. The temperature of these camera models is measured on the core board. Indicates that an over temperature condition exists and that damage to components of the camera may occur
5	Power failure	Indicates that the power supply is not sufficient. Check the power supply.
6	Insufficient trigger width	When a received trigger in the trigger width exposure mode is shorter than the minimum allowed exposure time, an insufficient trigger width error is reported.
* Only available for acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50. For information about which cameras have a GPIO line, see Section 5.2 on page 74.		

Table 45: Error Codes

When the camera detects a user-correctable error, it sets the appropriate error code in an error memory. If two or three different detectable errors have occurred, the camera will store the code for each type of error that it has detected (it will store one occurrence of each code no matter how many times it has detected the corresponding error).

To check error codes:

1. Read the value of the LastError parameter.
   The LastError parameter will indicate the last error code stored in the memory.
2. Execute the ClearLastError Command to clear the last error code from the memory.
3. Continue reading and clearing the last error until the parameter indicates a No Error code.

Reading and Clearing the Error Codes Using Basler pylon

You can use the pylon API to read the value of the LastError parameter and to execute a ClearLastError command from within your application software. The following code snippets illustrate using the pylon API to read the parameter value and execute the command:

```
// Read the value of the last error code in the memory
LastErrorEnums lasterror = Camera.LastError.GetValue();

// Clear the value of the last error code in the memory
Camera.ClearLastError.Execute( );
```

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

For more information about the pylon API and the pylon Viewer, see Section 3 on page 63.

9.8 Sequencer

Available for	Not Available for
acA640-90, acA640-120, acA645-100, acA750-30, acA780-75, acA1280-60, acA1300-22, acA1300-30, acA1300-60, acA1600-20, acA1600-60, acA1920-25 acA2000-50, acA2040-25, acA3800-10, acA4600-7	acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

The Sequencer feature will not work, if the Auto Functions feature is enabled. For more information about the Auto Functions feature, see Section 9.14 on page 328.

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.

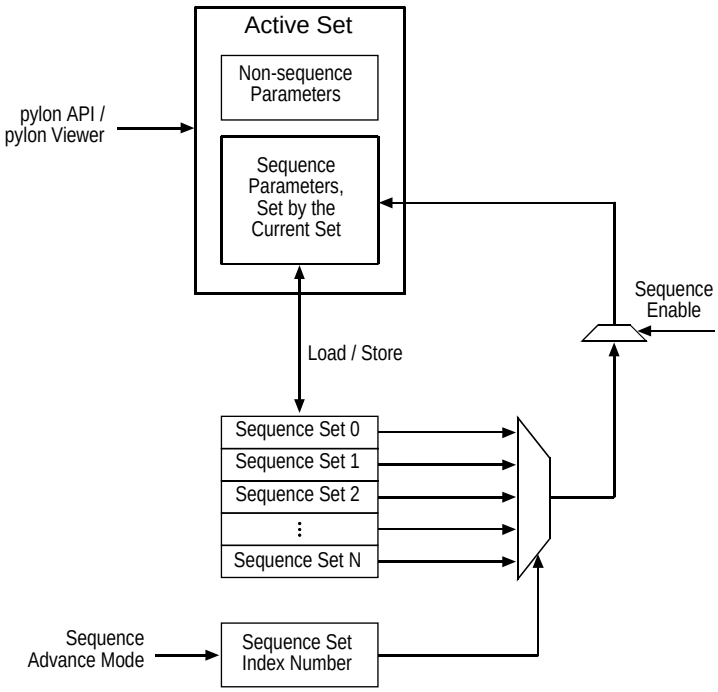


Fig. 110: Sequence Feature Block Diagram

The parameters in the active set can be divided into two types (Figure 110):

- "**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 360) with default values:
SequenceEnable, SequenceSetExecutions, SequenceControlSource, SequenceAddressBitSource, SequenceSetTotalNumber, SequenceSetIndex.

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 (acA1920-25, acA2500-14)

If you use the acA1920-25 and 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".

These parameters are included in each sequence set:

AcquisitionFrameRate	Height
BalanceRatio	LUTEnable
BinningHorizontal	PixelFormat
BinningVertical	ProcessedRawEnable
BlackLevel	ReverseX
CenterX	ReverseY
CenterY	ScalingHorizontal
ColorAdjustmentEnable	SequenceSetExecutions**
ColorAdjustmentHue	StackedZoneImagingZoneEnable
ColorAdjustmentSaturation	StackedZoneImagingZoneOffsetY
ColorTransformationValue	StackedZoneImagingZoneHeight
ColorTransformationMatrixFactor	SubsamplingHorizontal
ChunkModeActive	SyncUserOutput
ChunkEnable	TestImage
DecimationVertical	TimerDelay*
DigitalShift	TimerDuration*
EnableAcquisition rameRate	TimerDelayTimebase*
EnabledStackedZoneImaging	TimerDurationTimebase*
ExposureTime	Width
Gain	XOffset
GammaEnable	YOffset

* 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 may 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 10.8 on page 379.

Using the Load Command

Make sure the Sequencer feature is disabled before issuing the SequenceSetLoad command.

The **SequenceSetLoad** 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 SequenceSetLoad command is **not suitable** for real-time applications.

If you use the SequenceSetSelector parameter to select a sequence set and then you execute the SequenceSetLoad 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 SequenceSetLoad 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.

9.8.1 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 SequenceSetExecutions 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 SequenceSetExecutions parameter value, the sequence set cycle starts again with sequence set 0.

The SequenceSetTotalNumber parameter specifies the total number of different sequence sets that are available and included within a sequence set cycle. The maximum number is 64.

9.8.1.1 Operation

Operating the Sequencer

The following use case (see also Figure 111) 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 SequenceSetExecutions parameter settings. After one sequence set cycle is complete, another one starts.

In this use case, the SequenceSetTotalNumber parameter was set to six. Accordingly, the available sequence set index numbers range from 0 through 5. The SequenceSetExecutions 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 is received, the camera checks the current Sequence Set Executions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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 is received, the camera checks the current SequenceSetExecutions parameter value. Because the SequenceSetExecutions parameter is 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: SequenceSetTotal Number = 6
SequenceSetExecutions = 1 for sequence sets 0, 2, 3, and 4
SequenceSetExecutions = 2 for sequence set 5
SequenceSetExecutions = 3 for sequence set 1

▼ = camera selects a sequence set as the current sequence set

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

= frame exposure and readout

= frame transmission

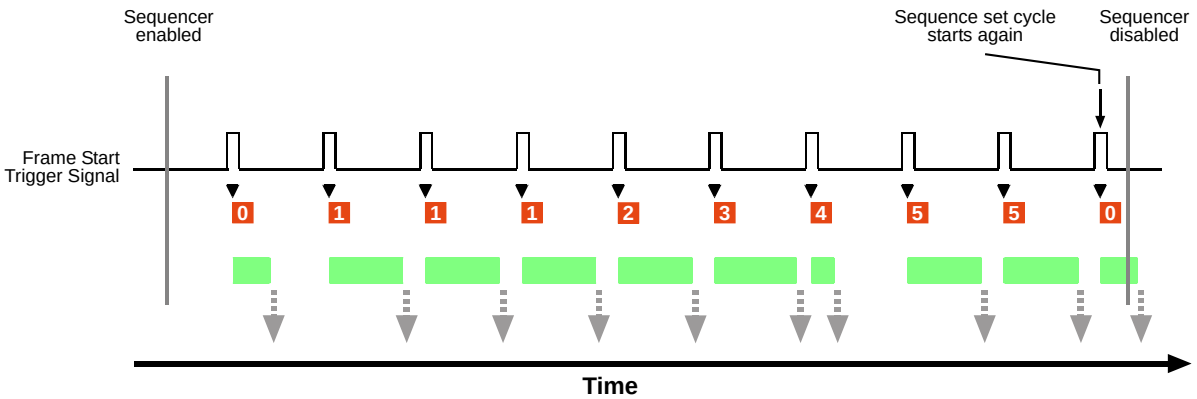


Fig. 111: Sequencer in Auto Sequence Advance Mode

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.

9.8.1.2 Configuration

Configuring Sequence Sets and Advance Control

To populate sequence sets and to make the related settings:

1. Make sure that the Sequencer feature is disabled.
2. Set the SequenceAdvanceMode parameter to Auto.
3. Set the SequenceSetTotalNumber parameter. The maximum number is 64.
4. Select a sequence set index number by setting the SequenceSetIndex 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 SequenceSetTotalNumber 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 SequenceSetExecutions parameter. The available numbers range from 1 to 256.
8. Execute the SequenceSetStore 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.

9.8.2 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:

- AlwaysActive: The advance from one sequence set to the next proceeds automatically as frame triggers are received.
- Line1: The states of the input line 1 control sequence set advance.
- Disabled: Sequence set advance is only controlled by AsyncAdvance software commands.

The SequenceSetTotalNumber parameter specifies the total number of different sequence sets that are available and included within a sequence set cycle. The maximum number is 64.

9.8.2.1 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 112) 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 SequenceSetTotal 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:	SequenceSetTotalNumber = 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

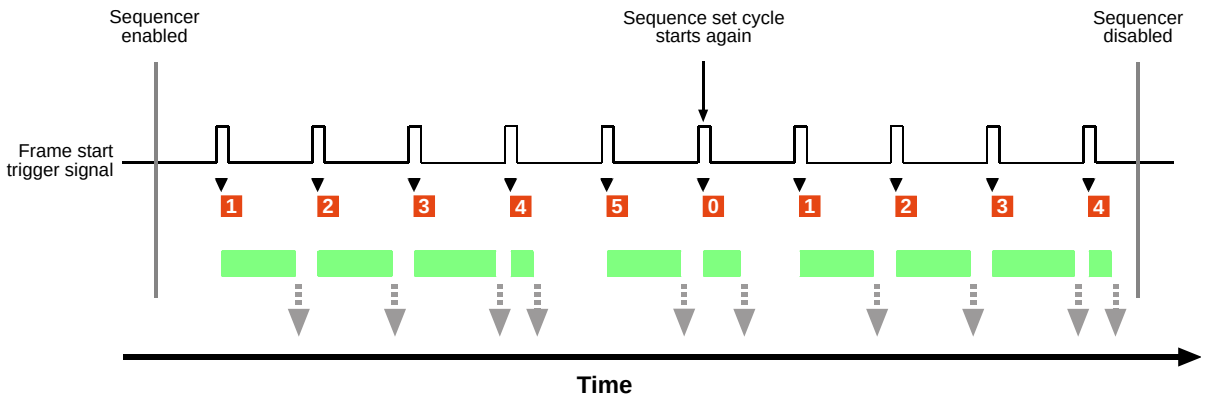


Fig. 112: Sequencer in Controlled Sequence Advance Mode with AlwaysActive as the Sequence Control Source

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 113), the same settings were made as in the previous use case: The SequenceSetTotalNumber 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 AlwaysActive 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 113).

	Make sure not to send a frame start trigger while the input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored. Make sure to only send a frame start trigger when the camera is in "waiting for frame start trigger" status. For information about possibilities of getting informed about the "waiting for frame start trigger" status, see the Acquisition Monitoring Tools section.

- When the next frame start trigger 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 AlwaysActive as the sequence control source: Automatic cycling through the sequence set cycles with two synchronous restartscontrolled by input line 1.
Setting:	SequenceSetTotalNumber = 6 Line1 (not set for invert) is selected as the source for controlling restart

= camera is waiting for a frame start trigger

= camera selects a sequence set as the current sequence set

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

= frame exposure and readout

= frame transmission

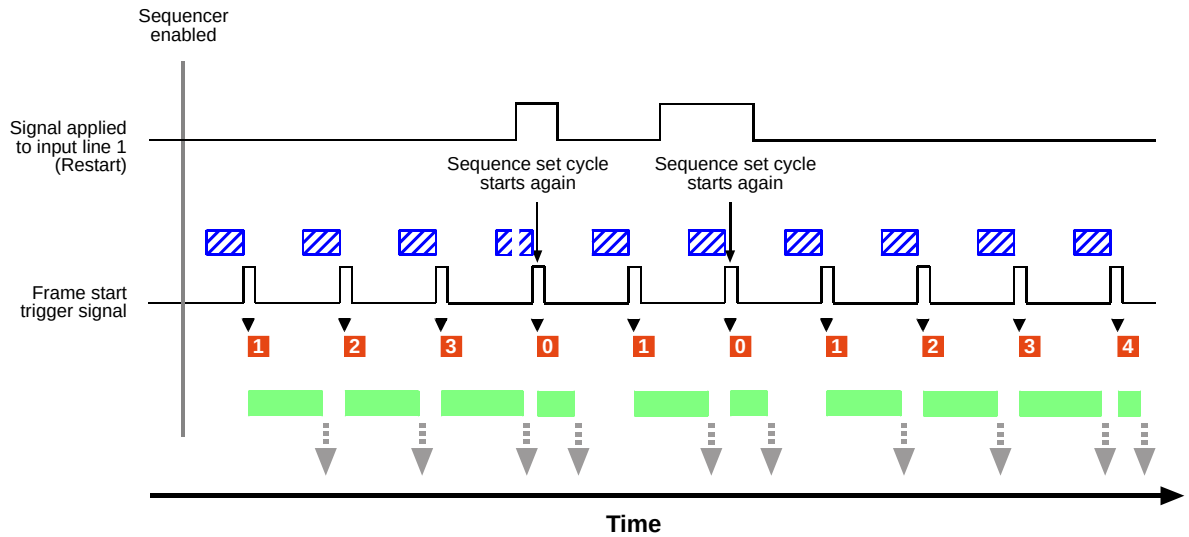


Fig. 113: Sequencer in Controlled Sequence Advance Mode with SequenceControlSource set to AlwaysActive and Synchronous Restart Controlled by Line 1

9.8.2.2 Operation with the Input Line as Sequence Control Source

Operating the Sequencer

When the SequenceControlSource parameter is set to Line1, 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 114) illustrates the operation of the sequencer in controlled sequence advance mode with the SequenceControlSource parameter set to Line1. 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 SequenceSetTotalNumber parameter was set to 6. 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 114).

	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:	SequenceSetTotalNumber = 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

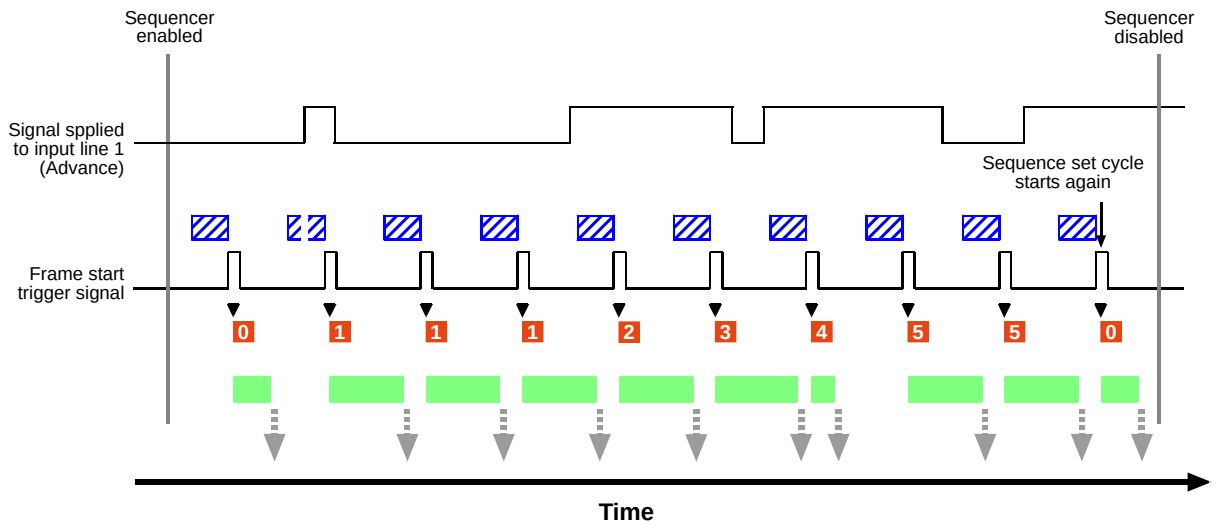


Fig. 114: Sequencer in Controlled Sequence Advance Mode with SequenceControlSource Set to Line1

9.8.2.3 Operation with the SequenceControlSource Set to Disabled

Operating the Sequencer

When the SequenceControlSource parameter is set to Disabled, the advance from one sequence set to the next proceeds in ascending sequence set index numbers and is only possible by asynchronous advance.
Similarly, sequence set restart is only possible by asynchronous restart.

	The delay between sending an AsyncAdvance or an AsyncRestart software command and it becoming effective will depend on the specific installation and the current load on the network. Accordingly, the number of image acquisitions that may occur between sending the software command and it becoming effective can not be predicted. Using the Sequencer feature with Disabled sequence control source is therefore not suitable for real-time applications, it may, however, be useful for testing purposes. We strongly recommend **not** to use the Sequencer feature with disabled sequence control source for real-time applications.

The following use case (see also Figure 115) illustrates the operation of the sequencer in controlled sequence advance mode with the SequenceControlSource set to Disabled. 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 SequenceSet TotalNumber parameter was set to 6. Accordingly, the available sequence set index numbers range from 0 through 5. The TriggerActivation parameter for the framestart trigger is set to RisingEdge.

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 115).

	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 115).

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:	SequenceSetTotalNumber = 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

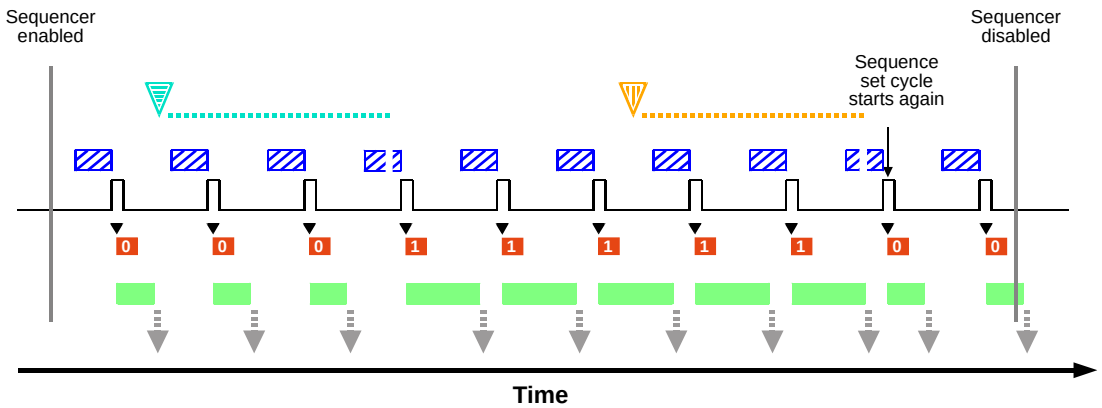


Fig. 115: Sequencer in Controlled Sequence Advance Mode with the SequenceControlSource Set to Disabled and Asynchronous Advance and Restart

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.

9.8.2.4 Configuration

Configuring Sequence Sets and Advance Control

To populate sequence sets and to set the sources:

1. Make sure that the Sequencer feature is disabled.
2. Set the SequenceAdvance mode to Controlled.
3. Set the SequenceSetTotalNumber parameter. The maximum number is 64.
4. Set the SequenceControlSelector parameter to Advance to configure synchronous sequence set advance.
5. Set the SequenceControlSource 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 Line1 is not allowed.

   The following sources are available:
   - AlwaysActive
   - Line1
   - Disabled
6. Set the SequenceControlSelector parameter to Restart to configure sequence set cycle restart.
7. Set the SequenceControlSource 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, set the SequenceControlSource parameter to Disabled.

The following sources are available:

- Line 1
- Disabled

8. Select a sequence set index number by setting the SequenceSetIndex 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.
9. Set up your first acquisition scenario (i.e., lighting, object positioning, etc.)
10. Adjust the camera parameters to get the best image quality with this scenario (you are adjusting the parameters in the active set).
11. Execute the SequenceSetStore 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.)
12. Repeat the above steps for the other sequence sets.

For information about setting the input line for invert, see Section 5.10.3 on page 101.

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.

9.8.3 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 SequenceSetTotal Number parameter specifies the total number of sequence sets that are available. The maximum number is 2.

9.8.3.1 Operation

Operating the Sequencer

The following use case (see also Figure 116) illustrates the operation of the sequencer in free selection sequence advance mode.

In this use case, the SequenceSetTotalNumber parameter was set to 2. 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 TriggerActivation parameter for the framestart trigger is set to RisingEdge.

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 is 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 is 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 is 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 is 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 happens when input line 1 goes high before the frame start trigger is received (see also Figure 116).

	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. 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 are 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 are 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 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:	SequenceSetTotalNumber = 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

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

= frame exposure and readout

= frame transmission

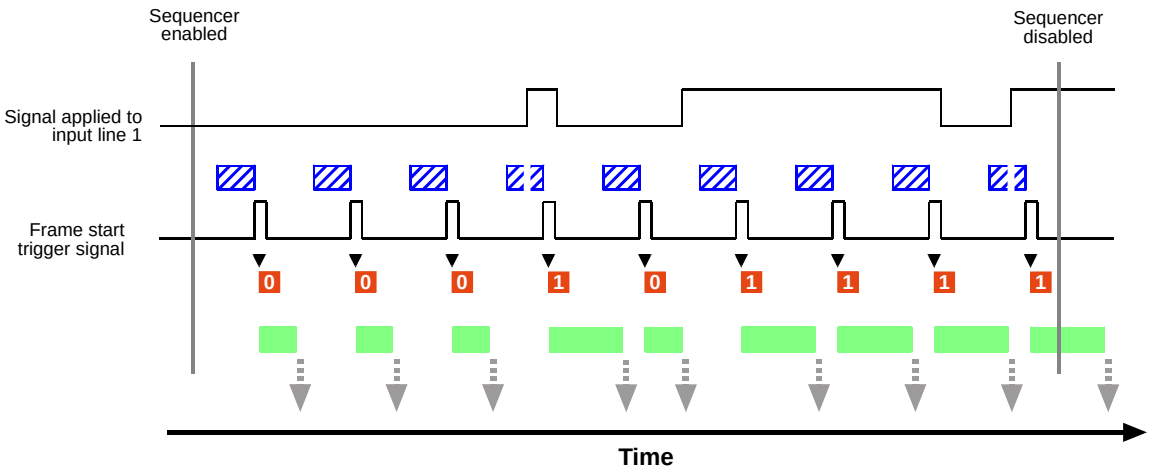


Fig. 116: Sequencer in Free Selection Mode

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 the binary expression of the sequence set index number (see Table 46).

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.
- 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 46: Sequence Set Addresses and Related Sequence Sets (Input Line Not Set for Invert)

9.8.3.2 Configuration

Configuring Sequence Sets and Advance Control

To populate sequence sets and to set the source:

1. Make sure that the Sequencer feature is disabled.
2. Set the SequenceAdvanceMode parameter to FreeSelection.
3. Set the SequenceSetTotalNumber 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:

   Bit 0 will be selected by default as the sequence set address bit.

   a. Set input line 1 as the control source for setting bit 0.
5. Use the SequenceSetIndex 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 SequenceSetStore 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.

9.9 Binning

With binning, multiple sensor pixels are combined and reported out of the camera as a single pixel.

Binning Directions

You can set binning in two directions: horizontal or vertical.

- With **vertical binning**, adjacent pixels from a specific number of **rows** (2, 3, 4) in the imaging sensor array are combined and are reported out of the camera as a single pixel.
- With **horizontal binning**, adjacent pixels from a specific number of **columns** (2, 3, 4) are combined and are reported out of the camera as a single pixel.

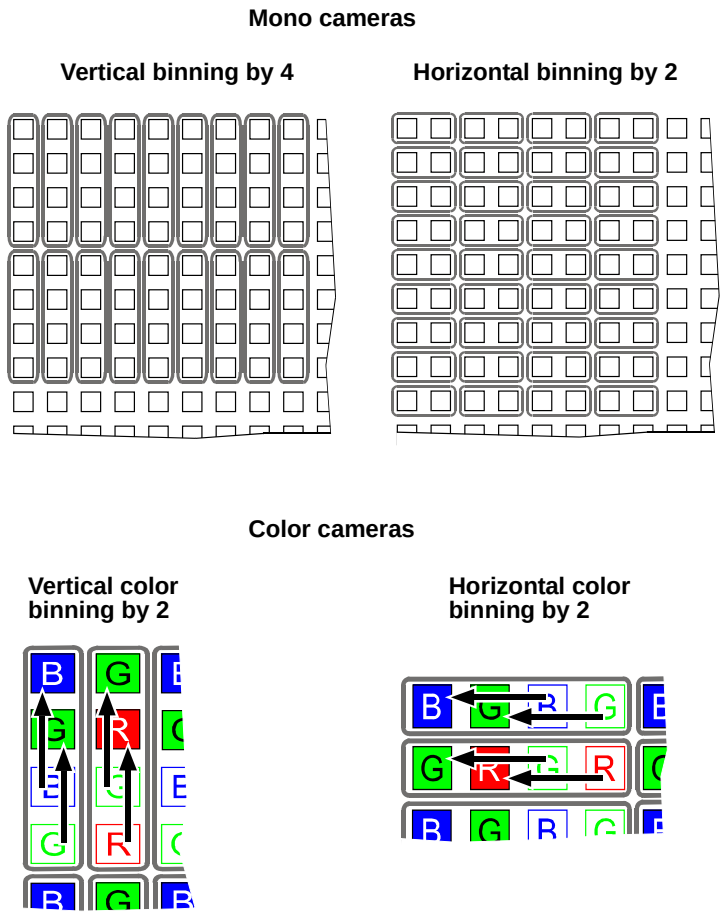


Fig. 117: Binning Direction Examples

You can use both vertical and horizontal binning at the same time. This, however, may cause objects to appear distorted in the image. For more information about possible image distortion, see Section 9.9.3 on page 310.

The number of binned pixels depends on the vertical binning and the horizontal binning settings. For more information about the binning settings, see Section 9.9.1 on page 308.

Binning Modes

Two modes can be used to perform binning:

- **Summing**: The values of the affected pixels are summed. This increases the camera's response to light.
- **Averaging**: The values of the affected pixels are averaged. This increases the signal-to-noise-ratio, effectively reducing image noise. The camera's response to light will not be increased.

Both modes reduce the amount of image data to be transferred, thus enabling higher camera frame rates.

The vertical **binning** mode and the horizontal binning mode can be set independently.

Usually, the binning modes used by the camera (vertical and horizontal) are preset and cannot be changed. However, on specific camera models and for specific binning directions, the binning mode can be set (see Table 47).

Camera Model	Vertical Binning Mode	Horizontal Binning Mode
acA640-90gm, acA640-120gm, acA645-100gm, acA780-75gm, acA1300-22gm, acA1300-30gm, acA1600-20gm, acA2000-50gm/gmNIR, acA2040-25gm/gmNIR	Summing	Summing
acA640-300gm, acA800-200gm, acA1300-75gm,	Averaging	Averaging
acA750-30gm	-	Summing
acA1280-60gm acA1300-60gm acA1600-60gm acA1920-40gm/gc, acA1920-50gm	Averaging or Summing (settable)	Averaging or Summing (settable)
acA3800-10gm acA4600-7gc	Averaging or Summing (settable)	Summing
acA1920-25gm/gc acA2500-14gm/gc	Averaging, Summing or a combination	Summing (settable)

Table 47: Camera Models and Supported Binning Modes

9.9.1 Setting Binning Parameters

You can enable

- **vertical binning** by setting the **Binning Vertical** parameter.
- **horizontal binning** by setting the **Binning Horizontal** parameter.

This applies to both color and mono cameras.

Setting the parameter's value to

- 2, 3, or 4: enables vertical or horizontal binning by 2, by 3, or by 4, respectively.
- 1: disables vertical or horizontal binning.

The range of allowed settings for the BinningVertical and the BinningHorizontal parameter values varies by camera model as shown in Table 48.

Camera Model	Allowed Settings for the Binning Vertical Parameter	Allowed Settings for the Binning Horizontal Parameter	Notes [1 disables horizontal or vertical binning]
acA640-90gm, acA640-120gm, acA640-300gm, acA645-100gm, acA780-75gm, acA800-200gm, acA1300-22gm, acA1300-30gm, acA1300-75gm, acA1600-20gm, acA1920-40gm,acA1920-50gm, acA2000-50gm/gmNIR, acA2040-25gm/gmNIR, acA1280-60gm, acA1300-60gm, acA1600-60gm/	1, 2, 3, 4	1, 2, 3, 4	-
acA750-30gm	1	1, 2	-
acA1920-25gm/gc acA2500-14gm/gc	1, 2, 4 (*) 3 (**)	1, 2, 3, 4	* The gray values of adjacent pixels from 2 or 4 rows are averaged. ** The gray values of adjacent pixels from 3 rows are combined (mixture of summing and averaging). Recommended: 2 or 4

Table 48: Binning Vertical and Binning Horizontal Settings

Camera Model	Allowed Settings for the Binning Vertical Parameter	Allowed Settings for the Binning Horizontal Parameter	Notes [1 disables horizontal or vertical binning]
acA3800-10gm acA4600-7gc	1, 2 (*), 4 (**)	1, 2, 3, 4	* The gray values of adjacent pixels from 2 rows are averaged. ** The gray values of ■ adjacent pixels from 2 rows (e.g. gray values of line 1 and 2) are summed. ■ the next 2 rows (e.g. gray values of line 3 and 4) are skipped. ■ the next 2 rows (e.g. gray values of line 5 and 6) are summed again and so on.

Table 48: Binning Vertical and Binning Horizontal Settings

You can set the BinningVertical or the BinningHorizontal 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.

9.9.2 Setting the Binning Mode

Usually, the binning modes used by the camera (vertical and horizontal) are preset and cannot be changed. However, on specific camera models and for specific binning directions, the binning mode can be set.

If supported, you can set the

- **horizontal binning mode** by setting the **Binning Horizontal Mode** parameter
- **vertical binning mode** by setting the **Binning Vertical Mode** parameter.

If supported, you can set the

- **horizontal binning mode** by setting the **Binning Horizontal Mode** parameter
- **vertical binning mode** by setting the **Binning Vertical Mode** parameter.

For more information about the supported binning modes, see Section 9.9 on page 306.

You can set the BinningVerticalMode and the BinningHorizontalMode 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:

```
// Set the horizontal binning mode to "Average"
camera.BinningHorizontalMode.SetValue(BinningHorizontalMode_Average);

// Determine the vertical binning mode
e = camera.BinningVerticalMode.GetValue();
```

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

9.9.3 Considerations When Using Binning

Increased Response to Light

Using binning can greatly increase the camera's response to light (except vertical binning by 3 for the acA1920-25 and acA2500-14 where the binning decreases the camera's reponse 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 9.1 on page 244.
Note: The vertical binning of the acA1920-25gm and acA2500-14gm works differently. For more information, see Section 9.9 on page 306.

Reduced Resolution

Using binning effectively reduces the resolution of the camera's imaging sensor. For example, the sensor in the acA640-120gm camera normally has a resolution of 659 (H) x 494 (V) pixels. 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 the dimensions of the sensor are not multiples of 3 and therefore can't be divided evenly by 3. To compensate for this, the values were rounded down to the nearest whole number.

To ensure that the desired scene appears completely in a binned image:

1. Set the desired binning factor.

   Settings made for offset, AOI width, and AOI height refer to the virtual sensor rows and columns.
2. Acquire an image.
3. Check whether the desired scene appears completely in the image.
4. If necessary, adjust the settings for the virtual rows or columns to fully capture the desired scene.

When you disable binning, the resolution will revert back to its original values.

Binning's Effect on AOI Settings

When you have set the camera to use binning, the maximum area of interest (AOI) will be made up of the binned lines and columns, i.e. it is going to be smaller than the actual sensor's maximum AOI. You can think of this as a "virtual sensor". Also, any offsets refer to the virtual sensor's position.

For example, assume that you are using an acA640-120gm camera set for 3 by 3 binning as described above. In this case, the maximum AOI would be 219 columns by 164 lines. The AOI width and height parameters are adjusted automatically to reflect this. Likewise, any offsets you have defined before enabling binning will be adjusted automatically.

When you disable binning, the AOI will increase again but may be smaller than the AOI you had set originally. This happens when the original AOI values can't be evenly divided by the binning factor, leaving a remainder of lines and columns, which is then ignored when the AOI is increased again. Therefore, Basler recommends to always check the AOI and offset settings after disabling binning and, if necessary, to manually set the AOI to the desired values.

For more information about the area of interest (AOI) feature, see Section 9.5 on page 261.

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 Decimation

If vertical binning is used, vertical decimation (see below) is automatically disabled, and vice versa, i.e. if vertical decimation is used, vertical binning is disabled.

Horizontal binning works independently of the Decimation Vertical feature.

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 49: Examples: Stacked Zone Imaging Settings for Zone 1

For more information about the Stacked Zone Imaging feature, see Section 9.6 on page 266.

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.

9.10 Decimation

Available for	Not Available for
All models, exceptions see on the right side	acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

Available for Camera Models	Vertical Decimation	Horizontal Decimation
acA1280-60, aca1300-60, acA1600-60	Decimation factor: 1 (disabled) to 32	Decimation factor: 1 (disabled) to 32
acA3800-10, acA4600-7	Decimation factor: 1 (disabled), 2, 4	Decimation factor: 1 (disabled), 2, 4
acA2000-50, acA2040-25	Decimation factor: 1 (disabled), 2, 4	-

Table 50: Camera Models and Decimation

9.10.1 Vertical Decimation

Valid for: see Table 50 above

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.

Examples
Blue rows will be transmitted:

If vertical decimation is set to

- 1: the complete frame will be transmitted out of the camera (no vertical decimation is realized); see Figure 118.
  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 119).
- 2 for color cameras: only every second pair of rows of the acquired frame will be transmitted out of the camera (Figure 120).

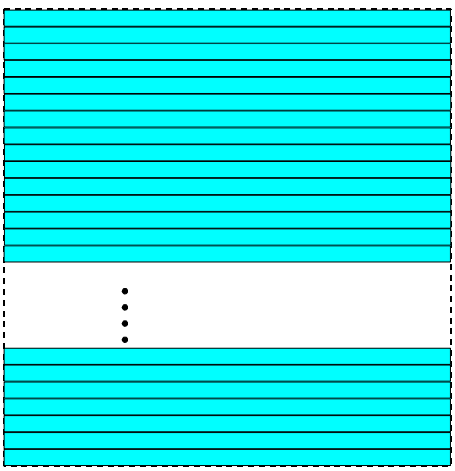


Fig. 118: Decimation Disabled

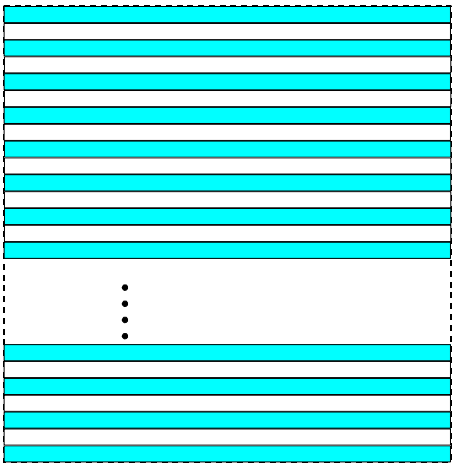


Fig. 119: Decimation of 2 (Mono Cameras)

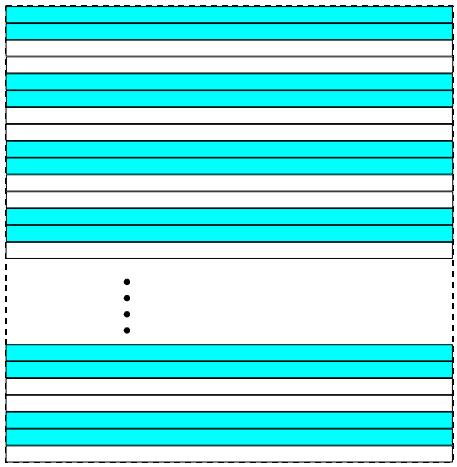


Fig. 120: Decimation of 2 (Color Cameras)

By using the Vertical Decimation feature, you can increase the frame rate of the camera.

Setting Vertical Decimation

You can enable vertical decimation by setting the DecimationVertical parameter. Setting the parameter's value to 1 disables vertical decimation.

You can set the DecimationVertical 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 63.

9.10.2 Horizontal Decimation

Available for Camera Models	Vertical Decimation	Horizontal Decimation
acA1280-60, aca1300-60, acA1600-60	Decimation factor: 1 (disabled) to 32	Decimation factor: 1 (disabled) to 32
acA3800-10, acA4600-7	Decimation factor: 1 (disabled), 2, 4	Decimation factor: 1 (disabled), 2, 4
acA2000-50, acA2040-25	Decimation factor: 1 (disabled), 2, 4	-

Table 51: Decimation and Camera Models

The Horizontal Decimation feature (sub-sampling in horizontal direction) lets you specify the extent of horizontal sub-sampling of the acquired frame, i.e. you can define pixel columns that you want to be left out from transmission.

The Horizontal Decimation feature does not increase the frame rate.

AOI width

If you use the Horizontal Decimation feature and you reset the decimation parameter back to 1, i.e. the Horizontal Decimation feature is deactivated, the AOI width can be smaller than the maximum possible width (determined by the pixel resolution in horizontal direction).

In this case you can manually set the AOI width back to the maximum possible width.

Setting Horizontal Decimation

You can enable Horizontal decimation by setting the DecimationHorizontal parameter. Setting the parameter's value to 1 disables horizontal decimation.

You can set the DecimationHorizontal 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 Horizontal Decimation by 8
Camera.DecimationHorizontal.SetValue(8);

// Disable Vertical Decimation
Camera.DecimationHorizontal.SetValue(1);
```

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

9.10.3 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 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

When you have the camera set to use decimation, keep in mind that the settings for your area of interest (AOI) will refer to the lines in the sensor and not to the physical lines in the sensor as they normally would.

For detailed information on the effect of the Decimation feature on the AOI settings, see section "Possible Image Distortion" on page 311.

AOI height

If you use the Vertical Decimation feature and you reset the decimation parameter back to 1, i.e. the Vertical Decimation feature is deactivated, the AOI height can be smaller than the maximum possible height (determined by the pixel resolution in vertical direction).

In this case you can manually set the AOI height back to the maximum possible height.

9.11 Scaling

Available for	Not Available for
acA3800-10gm/gc, acA4600-7gm/gc	All other models

The Scaling feature proportionally reduces the image size in horizontal and vertical direction by interpolation, i.e. pixel values are combined and averaged in order to obtain a reduced image size.

Range of settings for the ScalingHorizontal and ScalingVertical parameters: 0.125 - 1

You can set values that you obtain via this formula: 16/x (x = a whole number between 16 to 128).

Examples: 1 and 0.941, 0.888, 0.8.42 etc.

0.125: The image size is reduced by factor 8.

0.5: The image size is reduced by factor 2 (by half).

1: Disables scaling. The original image size is kept.
Binning and decimation are available.

In order to maintain the original height-to-width ratio, only the ScalingHorizontal parameter can be set. When the ScalingHorizontal parameter is changed, the ScalingVertical parameter is automatically adapted.

The Scaling feature can be used if the Sequencer feature is enabled, i.e. the sequencer sets used within a sequence can contain special scaling parameters.

Setting the Scaling Parameters

You can enable the Scaling feature by setting the ScalingHorizontal parameter.
If the Scaling feature is enabled, binning and decimation are automatically disabled.

Setting the ScalingHorizontal parameter value to 1 disables scaling.

You can set the ScalingHorizontal parameter value from within your application software by using the Basler pylon API. Setting the ScalingHorizontal parameter automatically sets the ScalingVertical accordingly. The following code snippet illustrates using the API to set the parameter value:

```
// Enable horizontal scaling by half
Camera.ScalingHorizontal.SetValue(0.5);

// Disable scaling
Camera.ScalingHorizontal.SetValue(1);
```

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

9.11.1 Considerations when Using Scaling

Binning and Decimation

If scaling is used, binning and decimation are automatically disabled, and vice versa, i.e. if binning or decimation is used, scaling is disabled.

Scaling’s Effect on AOI Settings

When you have set the camera to use scaling, the maximum area of interest (AOI) will be made up of the reduced lines and columns, i.e. it is going to be smaller than the actual sensor’s maximum AOI.

You can think of this as a "virtual sensor". Also, any offsets refer to the virtual sensor’s position.

For example, assume that you are using an acA4600-7gc camera set for scaling 0.2.
In this case, the maximum AOI would be 921 lines by 657 columns. The AOI height and width parameters are adjusted automatically to reflect this. Likewise, any offsets you have defined before enabling scaling will be adjusted automatically.

When you disable scaling, the AOI will increase again but may be smaller than the AOI you had set originally. This happens when the original AOI values can’t be evenly divided by the scaling factor, leaving a remainder of lines and columns, which is then ignored when the AOI is increased again. Therefore, Basler recommends to always check the AOI and offset settings after disabling scaling and, if necessary, to manually set the AOI to the desired values.

The information about the AOI is also valid for the Auto Function AOI, i.e. always check the Auto Function AOI after disabling scaling and, if necessary, manually set the Auto Function AOI to the desired values

9.12 Mirror Imaging

The camera's reverse X and reverse Y functions let you flip the captured images horizontally and/or vertically before they are transmitted from the camera.

Note that the reverse X and reverse Y functions may both be enabled at the same time if so desired.

Use of mirror imaging features changes Bayer color filter alignment of certain cameras

For color cameras, provisions are made ensuring that the effective color filter alignment will be constant for both, normal and mirror images.

Exceptions (*): acA640-300gc, acA800-200gc, acA1300-75gc, acA1920-40gc acA1920-50gc

When you configure the cameras mentioned above (see *), take into account that, if you enable the Reverse X and/or the Reverse Y feature, the effective Bayer color filter alignment will change as follows: If you use

- the Reverse X feature to mirror the image horizontally, the effective Bayer color filter alignment will be GB.
- the Reverse Y feature to mirror the image vertically, the effective Bayer color filter alignment will be GR.
- the Reverse X and the Reverse Y feature, the effective Bayer color filter alignment will be RG.

9.12.1 Reverse X

Available for
All camera models

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 121 shows a normal image on the left and an image captured with reverse X enabled on the right.

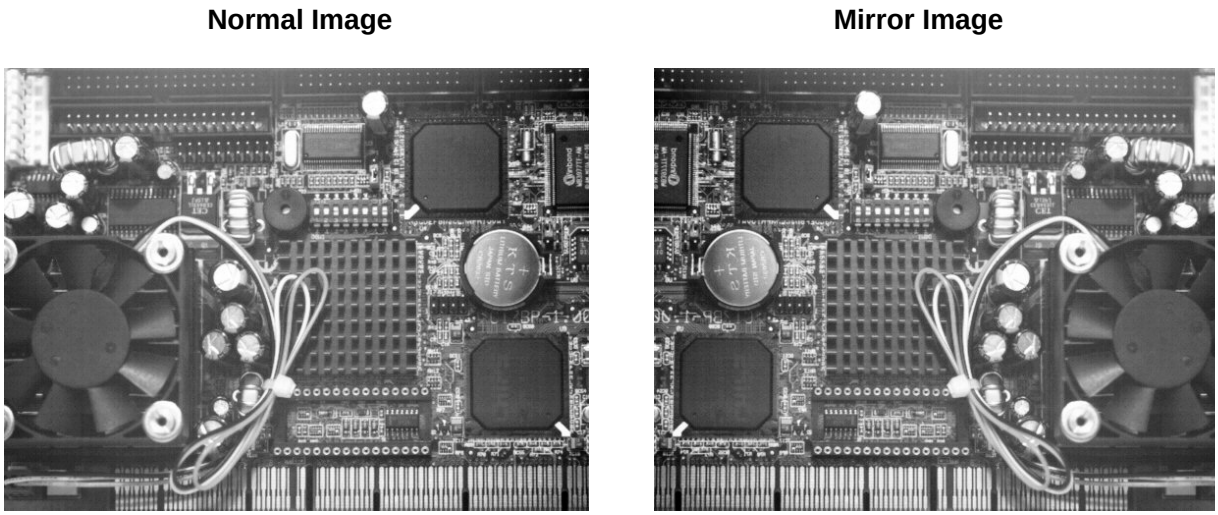


Fig. 121: Reverse X Mirror Imaging

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.

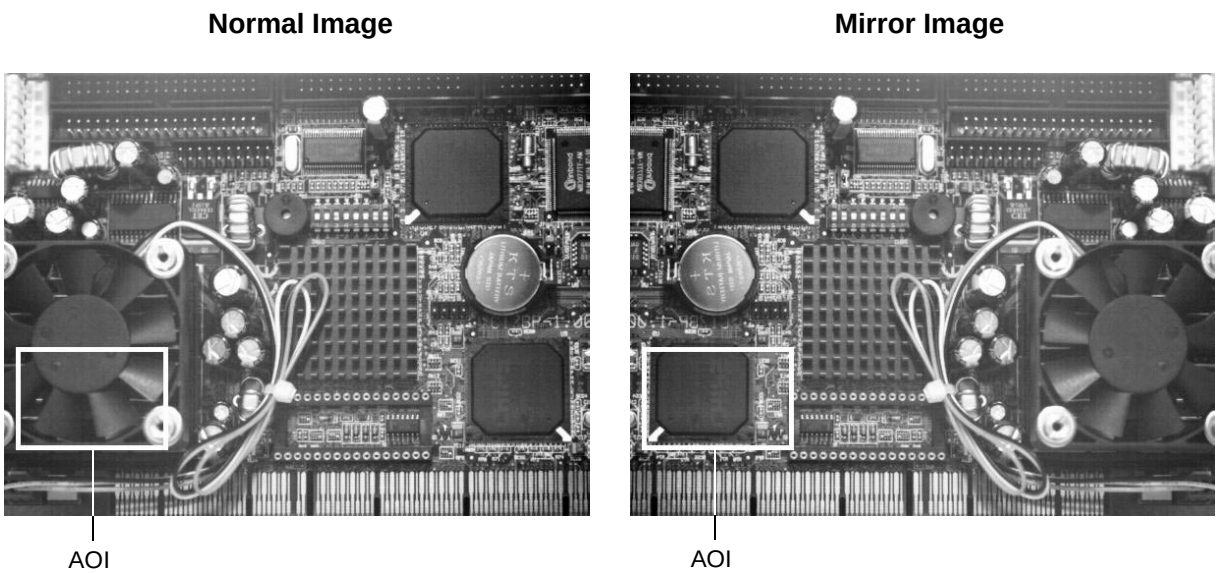


Fig. 122: Using an AOI with Reverse X Mirror Imaging

	AOIs used for the Auto Function feature will behave analogously to "standard" AOIs: When reverse X is used, the position of the auto function AOIs relative to the sensor remains the same. As a consequence, each auto function AOI will include a different portion of the captured image depending on whether or not the Reverse X feature is enabled. As a consequence, each auto function AOI will include a different portion of the captured image depending on whether or not the Reverse X feature is enabled.

For more information about auto functions, see Section 9.14 on page 328.

9.12.2 Reverse Y

Available for	Not Available for
acA640-300gm/gc, acA800-200gm/gc, acA1300-75gm/gc, acA1920-40gm/gc, acA1920-50gm/gc, acA2000-50gm/gc, acA2040-25gm/gc	All other models

The Reverse Y feature is a vertical mirror image feature. When the Reverse Y feature is enabled, the lines in a captured image will be swapped top-to-bottom. This means that the top line in the image will be swapped with the bottom line, the next-to-top line will be swapped with the next-to-bottom line, and so on.

Figure 123 shows a normal image on the left and an image captured with reverse Y enabled on the right.

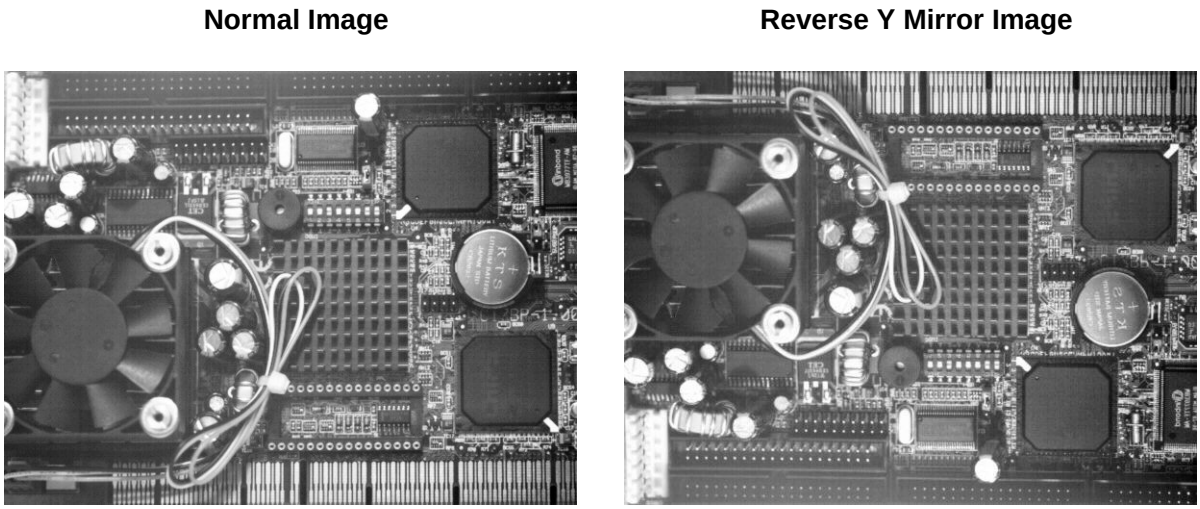


Fig. 123: Reverse Y Mirror Imaging

The Effect of Reverse Y on the Auto Function AOIs

If you are using the camera's auto functions, you should be aware of the effect that using the Reverse Y feature will have on the auto function AOIs. When reverse Y is used, the position of the auto function AOIs relative to the sensor remains the same. As a consequence, each auto function AOI will include a different portion of the captured image depending on whether or not the Reverse Y feature is enabled.

Figure 124 shows the effect the reverse Y mirroring will have on the auto function AOIs.

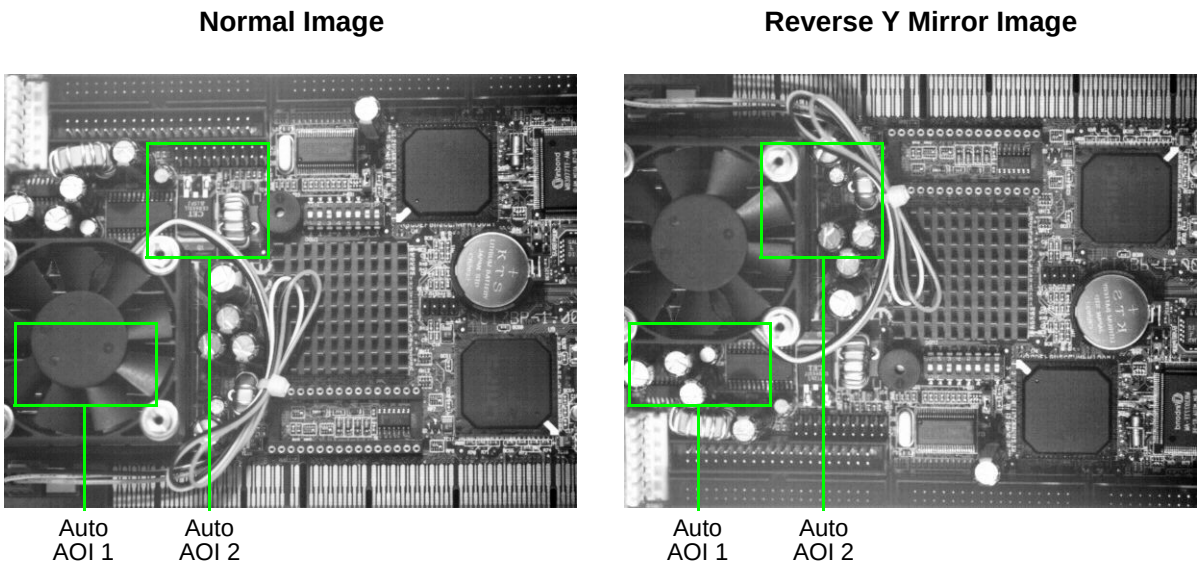


Fig. 124: Using Reverse Y Mirror Imaging with Auto Functions Enabled

For more information about auto functions and auto function AOIs, see Section 9.14 on page 328.

9.12.3 Enabling Reverse X and Reverse Y

You can enable the Reverse X and Reverse Y features by setting the ReverseX and the ReverseY parameter values. You can use the pylon API to set the parameter values from within your application software. The following code snippet illustrates using the API to set the values:

```
// Enable reverse X
Camera.ReverseX.SetValue(true);

// Enable reverse Y
Camera.ReverseY.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 63.

9.13 Luminance Lookup Table

Depending on the camera, pixel data from the imaging sensor is digitized by the ADC

- at 12-bit depth
- at 10-bit-depth

Whenever the camera is set for a 12-bit or 10-bit pixel format (e.g., Mono 12 or Mono 10), the 12 bits or 10 bits transmitted out of the camera for each pixel normally represent the 12 bits or 10 bits reported by the camera's ADC.

The Luminance Lookup Table feature lets you use

- a custom 10-bit to 10-bit lookup table to map the 10 bits reported out of the ADB to 10 bits that will be transmitted by the cameras.
- a custom 12-bit to 12-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 (for 12 bits) or 1024 values (for 10 bits); however, not every value in the table is actually used. If we number the values in the table from 0 through 4095 or from 0 to 1024, the table works like this (12-bit example):

Number(s) at location 12-bit depth	Represents
0	... the 12 bits that will be transmitted out of the camera when the ADC reports that a pixel has a value of 0.
1 - 7	Not used
8	... the 12 bits that will be transmitted out of the camera when the ADC reports that a pixel has a value of 8
9 - 15	Not used
16	... the 12 bits that will be transmitted out of the camera when the ADC reports that a pixel has a value of 16
17 - 23	Not used
24	... 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	...
4089	Not used
4095	Not used

Table 52: Luminance Lookup Table Numbers and What they Represent (12-bit Depth)

As you can see, the table does not include a user defined 12-bit value for every pixel value that the sensor can report. 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.

Location 4088 is the last location that will have a defined 12-bit value associated with it. 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 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.

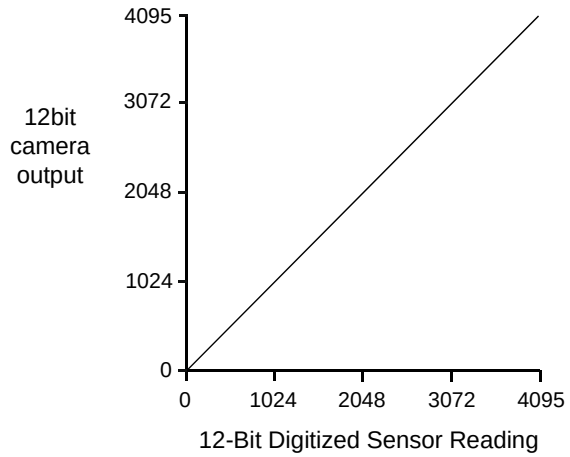


Fig. 125: Lookup Table with Values Mapped in a Linear Fashion

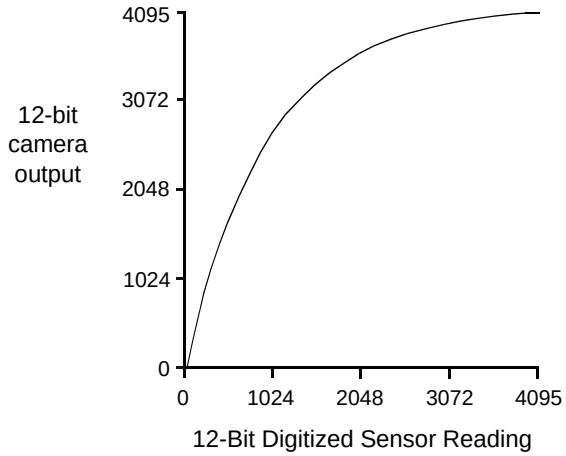


Fig. 126: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

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

To change the values in the lookup table and to enable the table:

1. 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.
2. 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.
3. Use the LUT Value parameter to set the selected value in the lookup table.
4. Use the LUT Index parameter and LUT value parameters to set other table values as desired.
5. 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 63.

9.14 Auto Functions

	The Auto Functions feature will not work, if the Sequencer feature is enabled. For more information about the Sequencer feature, see Section 9.8 on page 273.

9.14.1 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 always share an 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 acA1920-25gc, and 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 9.9 on page 306.

9.14.2 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.

9.14.3 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 127.

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 127.

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.

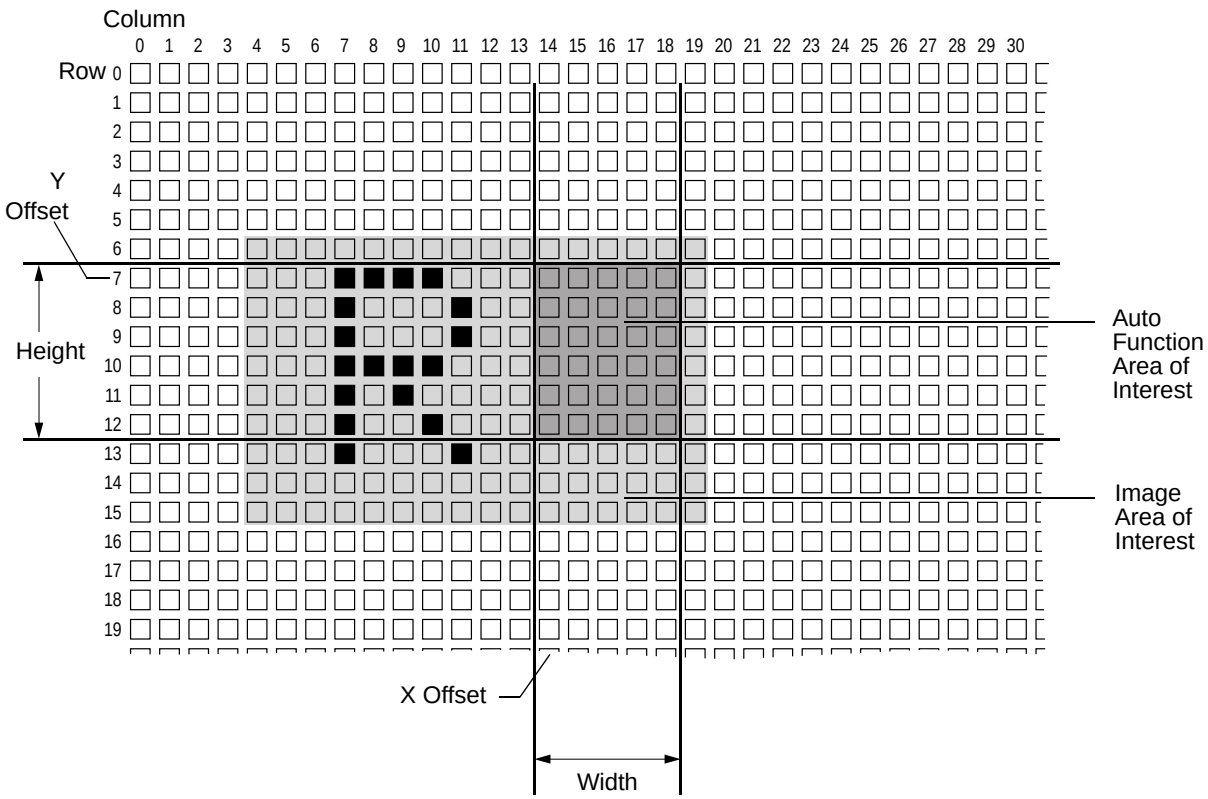


Fig. 127: Auto Function Area of Interest and Image Area of Interest

9.14.3.1 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.

9.14.3.2 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 128. 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 128), 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 128), 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 128), 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 128), 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 and Reverse Y mirroring features. For information about the behavior and roles of Auto Function AOI and Image AOI when also using the Reverse X or Reverse Y mirroring feature, see the "Mirror Image" (Section 9.12 on page 319).

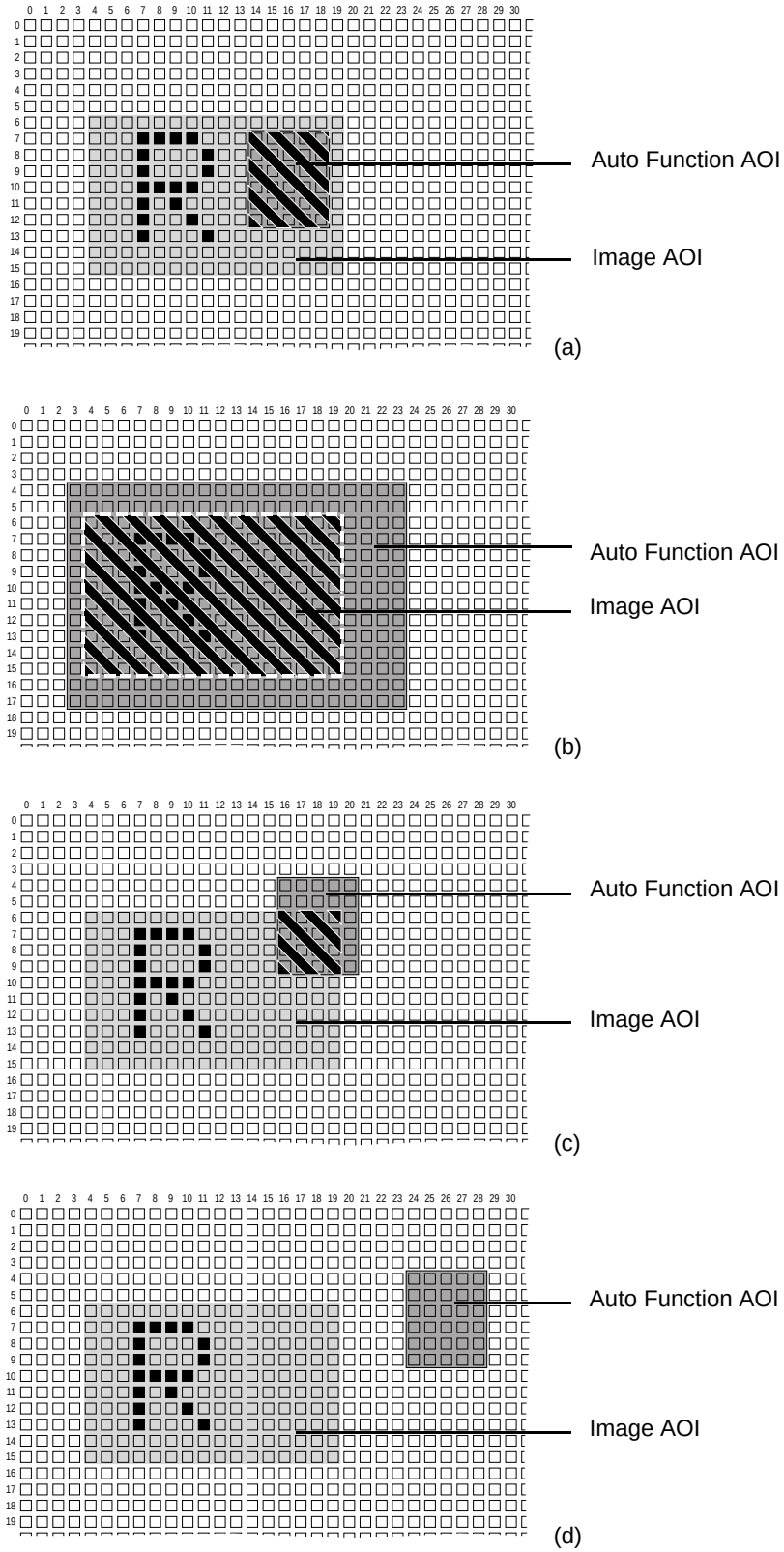


Fig. 128: Various Degrees of Overlap Between the Auto Function AOI and the Image AOI

9.14.3.3 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.

- Offset X: determines the starting column for the Auto Function AOI.
- Offset Y: determines the starting row for the Auto Function AOI.
- Width: determines the width of the Auto Function AOI.
- Height: determines the height of the Auto Function AOI.

When you are setting an Auto Function AOI, you must follow these guidelines:

Valid for All Camera Models	
Offset X + Autofunction AOI width < Width of camera sensor	Example: acA640-120gm: Sum of Offset X + Width < 659.
Offset Y + Autofunction AOI height < Height of camera sensor	Example: acA640-120gm: Sum of Offset Y+ Height < 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 AutoFunctionAOIOffsetX, AutoFunctionAOIOffsetY, AutoFunctionAOIWidth, and AutoFunctionAOIHeight 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 AutoFunctionAOIOffsetX parameter to 0, 2, 4, 6, 8, etc.

	Normally, the offset X, offset Y, 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 9.9.3 on page 310.

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 63.

9.14.4 Gain Auto

Gain Auto is the "automatic" counterpart to manually setting the GainRaw parameter. When the gain auto function is operational, the camera will automatically adjust the GainRaw 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 GainRaw 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 9.1 on page 244.
- the Auto Function Profile feature, see Section 9.14.7 on page 341.

The limits within which the camera will adjust the GainRaw parameter are defined by the AutoGainRawUpperLimit and the AutoGainRawLowerLimit parameters. The minimum and maximum allowed settings for the AutoGainRawUpperLimit and the AutoGainRawLowerLimit 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 AutoTargetValue parameter defines the target average gray value that the gain auto function will attempt to achieve when it is automatically adjusting the GainRaw 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:

To set the gain auto functionality:

1. Select the AutoFunctionAOISelector to AOI1.
2. Set the value of the following parameters for the AOI:
   - AutoFunctionAOIOffsetX
   - AutoFunctionAOIOffsetY,
   - AutoFunctionAOIWidth
   - AutoFunctionAOIHeight
3. Set the value of the AutoGainRawLowerLimit and AutoGainRawUpperLimit parameters.
4. Set the value of the AutoTargetValue parameter.
5. Determine the mode of operation by setting the GainAuto parameter to Once or the Continuous.

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 general information about auto functions, see Section 9.14 on page 328.

For more information about

- the pylon API and the pylon Viewer, see Section 3 on page 63.
- Auto Function AOIs and how to set them, see Section 9.14.3 on page 330.

9.14.5 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 6.4.3.2 on page 143.

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 ExposureTimeAbs 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 6.11 on page 192.
- the trigger width exposure mode, see Section 6.4.3.2 on page 143.
- the Auto Function Profile feature, see Section 9.14.7 on page 341.

The limits within which the camera will adjust the ExposureTimeAbs parameter are defined by the AutoExposureTimeAbsUpperLimit and the AutoExposureTimeAbsLowerLimit parameters. The current minimum and the maximum allowed settings for the AutoExposureTimeAbsUpperLimit parameter and the AutoExposureTimeAbsLowerLimit parameters depend on the minimum allowed and maximum possible exposure time for your camera model.

The AutoTargetValue parameter defines the target average gray value that the exposure auto function will attempt to achieve when it is automatically adjusting the ExposureTimeAbs 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 AutoExposureTimeAbsUpperLimit parameter is set to a sufficiently high value the camera's frame rate may be decreased.

To set the exposure auto functionality using Basler pylon:

1. Set the AutoFunctionAOISelector to AOI1.
2. Set the value of the AutoFunctionAOIOffsetX, AutoFunctionAOIOffsetY, AutoFunctionAOIWidth, and AutoFunctionAOIHeight parameters for the AOI.
3. Set the value of the AutoExposureTimeAbsLowerLimit and AutoExposureTimeAbsUpperLimit parameters.
4. Set the value of the AutoTargetValue parameter.
5. Determine the mode of operation by setting the GainAuto parameter to Once or the Continuous.

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 information about

- the pylon API and the pylon Viewer, see Section 3 on page 63
- Auto Function AOIs and how to set them, see Section 9.14.3 on page 330
- minimum allowed and maximum possible exposure time, see Section 6.11 on page 192.

For general information about auto functions, see Section 9.14 on page 328.

9.14.6 Gray Value Adjustment Damping

The gray value adjustment damping controls the rate by which pixel gray values are changed when the exposure auto function and/or the gain auto function 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 GrayValueAdjustmentDampingAbs parameter. The parameter can be set in a range from 0.0 to 0.78125.

The higher the value,

- 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.

9.14.7 Auto Function Profile

	The Auto Function Profile feature will only take effect if you use the gain auto function and the exposure auto function at the same time.

The auto function profile specifies how the gain and the exposure time will be balanced when the camera is making automatic adjustments.

If you want to use this feature, you must enable both the gain auto function and the exposure auto function and set both for the continuous mode of operation.

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.

All Basler ace GigE cameras support the following auto function profiles:

- Gain Minimum: Gain will be kept as low as possible during automatic adjustments.
- Exposure Minimum: Exposure time will be kept as low as possible during automatic adjustments.

By default, the Auto Function Profile feature minimizes gain.

To use the gain auto function and exposure auto function at the same time:

1. Set the value of the AutoFunctionProfile parameter to specify whether gain or exposure time will be minimized during automatic adjustments.
2. Set the value of the GainAuto parameter to Continuous.
3. Set the value of the ExposureAuto parameter to Continuous.

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 63.

9.14.8 Balance White Auto

Available for
Color cameras only

Balance White Auto is the "automatic" counterpart to manually setting the white balance.

Automatic white balancing is a two-step process. First, the BalanceRatioAbs 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 7.2 on page 212.

To set the balance white auto functionality:

1. Select the Auto Function AOI 2.
2. Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
3. Set the value of the BalanceWhiteAuto 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 63.
- Auto Function AOIs and how to set them, see Section 9.14.3 on page 330.

For general information about auto functions, see Section 9.14 on page 328.

9.14.8.1 Balance White Adjustment Damping

The balance white adjustment damping controls the rate by which the colors red, green, and blue are adjusted such that white objects in the camera's field of view appear white in the acquired images.

If an adjustment damping factor is used, the white balance 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 balance white adjustment damping is set to 0.976562. This is a setting where the damping control is as stable and quick as possible.

Setting the Adjustment Damping

The balance white adjustment damping is determined by the value of the BalanceWhiteAdjustmentDampingAbs parameter. The parameter can be set in a range from 0.0 to 0.976562.

The higher the value,

- the sooner the target value will be reached,
- the adaptation is realized over a smaller number of frames.

Examples:
0.9765 = Default value the camera starts with. There is a relatively immediate continuous adaptation to the target 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 balance white adjustment damping from within your application software by using the pylon API. The following code snippets illustrate using the API to set the balance white adjustment damping:

```
Camera.BalanceWhiteAdjustmentDampingRaw.SetValue(600);
Camera.GrayValueAdjustmentDampingAbs.SetValue(0.5859);
```

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

9.14.9 Using an Auto Function

To use an auto function:

1. Select an Auto Function AOI.
2. Assign the auto function you want to use to the selected Auto Function AOI.
3. Unassign the auto function you want to use from the other Auto Function AOI.
4. Set the position and size of the Auto Function AOI.
5. If necessary, set the lower and upper limits for the auto functions's parameter value.
6. If necessary, set the target value.
7. If necessary, set the GrayValueAdjustmentDampingAbs parameter.
8. If necessary, set the BalanceWhiteAdjustmentDampingAbs parameter.
9. If necessary, set the auto function profile to define priorities between auto functions.
10. 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.

9.14.10 Pattern Removal

Available for
acA3800-10

9.14.10.1 Monochrome Cameras

Images output by the monochrome acA3800-10gm cameras can display a superposed artifact pattern resembling a checker pattern.

You can suppress the formation of the "checker pattern" to a great extent by applying correction coefficients to the original pixel values. The Pattern Removal feature allows you to configure the correction coefficients.

The correction coefficients are automatically applied during each image acquisition and can’t be disabled. Correction coefficient values are **only** valid for the specific imaging conditions (see below) that were present when the correction coefficients were configured.

When Basler acA3800-10gm cameras are switched on for the first time, they wake up with default pattern removal correction values.

During normal operation these correction values are applied to all pixels of the captured images.

As these default correction values are not adapted to your final camera operating conditions (light conditions, optics settings), you must create pattern removal correction values that are created under the normal working conditions of the camera. You must therefore generate new correction coefficient values when you enable or change one or more of the relevant "imaging conditions": Among them are the following:

- Optical system: exchange of lens, change of aperture, change of focus
- Illumination: change of the type of illumination, change of the arrangement of light sources, change of brightness
- Camera settings and features: The checker pattern depends on several camera settings and features, in particular exposure time, Black Level, Digital Shift, Binning Horizontal, Binning Vertical, LUT, some image AOI-related settings (Width, Height, OffsetX, OffsetY, CenterX, CenterY).

Pattern removal correction values should be saved in a user set so that they are available after restart of the camera.

For information about

- how to create correction values for the pattern removal function, see below.
- configuration set, factory sets, and user sets, see from page 360 on.

Make sure the Sequencer feature and all auto functions except Pattern Removal Auto are disabled when generating new correction coefficients.

	We strongly recommend to generate new correction coefficients whenever you change the imaging conditions.

The Pattern Removal Auto Function and Its Operation

The pattern removal auto function differs in some respects from other auto functions: It does not employ any Auto Function areo of interest (Auto Function AOI). A "target" value does not exist. Instead, the auto function aims at generating correction coefficient values that will remove the checker pattern as far as possible. Only the "once" mode of operation is available to generate correction coefficient values.

Newly generated correction values will be stored in the camera's volatile memory (the active set) and will be lost if the camera is reset or if power is switched off. You can, however, save them in one of the user sets 1 thorough 3. The correction values will then be immediately available whenever you want to use them. In this case, however, make sure the camera is operated at exactly the imaging conditions that were present when the correction coefficients values were generated.

	We recommend not to use the Pattern Removal Auto Function when other auto functions are used unless the automatic changes are very limited and close to the imaging conditions for which the correction values were generated. A similar restriction applies when using Pattern Removal Auto Function with the Sequencer feature. Note that correction coefficient values can not be stored in sequencer sets.

Pattern Removal and Camera Startup

When the camera is switched on or reset, correction values from one of the user sets will be loaded into the active set if the user set was configured as user set default. Otherwise, factory-generated correction values will be loaded that are only appropriate for the imaging conditions chosen by the factory. Most likely, your imaging conditions will differ and you must therefore generate new correction values for your imaging conditions.

Generating Correction Values for the Pattern Removal Function

To generate correction values for the pattern removal function:

1. If possible, establish homogeneous illumination for the scene to be imaged.
2. Deactivate all camera settings and features (e.g. auto functions, sequencer) that would interfere with the generation of correction coefficient values.
3. Adjust the optical system, illumination, camera settings (e.g. exposure time, Digital Shift, Black Level) as required for the following image acquisitions. For best results, the image should display some average gray.
4. Set Pattern Removal Auto to Once.

5. Acquire three images to generate correction coefficient values. Between acquisitions, the scene, should ideally not change or at least as little as possible. You can use the "single frame" or "continuous frame" acquisition mode.

   The correction coefficient values are generated in a tree-step process during which they are refined with each step: After the first trigger, correction coefficient values are generated and are used for the second acquisition. Based on the second acquisition, new values are generated and are used for the third acquisition. Based on it, even further refined values are generated.

   These are the "final" ones that are used to perform pattern removal for all subsequent acquisitions until different correction coefficient values are loaded into the active set.

   After the correction coefficient values are generated, PatternRemovalAuto is automatically set to Off.

6. Save the created correction parameters in a user set that you can load afterwards for normal operation into the active set.
   If you do not save correction values in a user set, the adapted values will get lost and the default pattern removal correction values will be applied during the next image captures.

	Any time you make a change to the exposure time, light settings, and/or optics (lens), you must update your correction values for the pattern removal function. If ▪ you do not create new correction values for new light and optics settings and ▪ you do not save them in a user set, the old correction values, stored the last time, will be valid, and the old correction values might not be the suitable for your new light/optics settings.

Enabling the Pattern Removal Function Using the pylon API

You can enable the PatternRemovalAuto function from within your application software by using the Basler pylon API.

The following code snippet illustrates using the API to enable the pattern removal functionality:

```
Camera.PatternRemovalAuto.SetValue(PatternRemovalAuto_Once);
```

After three image captures the camera automatically sets the pattern removal function to off (PatternRemovalAuto_Off).

	We recommend not to use the Pattern Removal Auto Function when other auto functions are used unless the automatic changes are very limited and close to the imaging conditions for which the correction values were generated. A similar restriction applies when using Pattern Removal Auto Function with the Sequencer feature. Note that correction coefficient values can not be stored in sequencer sets.

9.14.10.2 Color Cameras

In color cameras with Aptina sensor, groups of four pixels each display the same characteristic as their monochrome counterparts, that is, a tendency to different response to light. The resulting artifact effect produces slight color shifts. These can be corrected by using the white balance feature. The need for correction applies to acA3800-10gc and acA4600-7gc cameras.

As with monochrome cameras, the artifact effect varies with certain "imaging conditions", that are defined by the optical system, the illumination, and several camera settings. Accordingly, to correct for artifact color shifts, you must perform white balance whenever at least one of the relevant imaging conditions changes. This means also that you may have to perform white balance when you normally would not, for example after having changed the lens focus.

9.15 Median Filter

Available for
acA1280-60, acA1300-60, acA1600-60

The cameras offer a Median Filter feature that, for example, can be used to reduce noise in images.

The median filter is a multi-directional 3x3 weighted median filter. The filter is compatible with mono and color cameras.

Setting the Median Filter

You can set the MedianFilter parameter from within your application software by using the Basler pylon API.

The following code snippet illustrates using the API to enable the median filter:

```
// Enable the median filter
camera.MedianFilter.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 63.

9.16 Event Notification

Available for
All models

Event notification is available on the camera. With event notification, 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. (*)
- The critical temperature of 77 °C has been reached (CriticalTemperatureEventData). (*)

  This event warns you that the camera is about to reach the internal over temperature condition.
- The over temperature of 80 °C has been reached (OverTemperatureEventData). (*)

  If the over temperature is reached, the camera enters the over temperature idle mode. In this mode, the camera will no longer acquire images but display test image 2.

  * Only available for certain cameras; see next pages.

An Example of Event Notification

An example related to the Frame Start Overtrigger event illustrates how works. The example assumes that your system is set for event notification (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 FrameStartOvertrigger 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

Available for	Not Available for
All models, exceptions see right.	acA640-300*, acA800--200*, acA1300-75*, acA1920-40*, acA1920-50*
* The camera models marked with an asterisk have been re-designed. They no longer use a central event queue that could possibly overflow. As a result, they don’t generate overflow events.	

Most cameras (exceptions, see table above) have 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 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 Notification

Event notification 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 SDKs delivered with the pylon Camera Software Suite, see Section 3.1.3 on page 64.

Event notification 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
Critical Temperature **	EventCriticalTemperatureEventData	EventCriticalTemperatureEventTimestamp
Over Temperature **	EventOverTemperatureEventData	EventOverTemperatureEventTimestamp
* Not available for acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50 ** Only available for: acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50		

Table 53: Parameter Names of Events and Supplementary Information

You can enable event notification 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.

9.17 Test Images

The 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.

Test image generation is done internally by the camera's logic and does not use the optics or the imaging sensor. Six test images are available; for mono cameras: 5 test images.

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 TestImageSelector 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 63.

9.17.1 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 129.

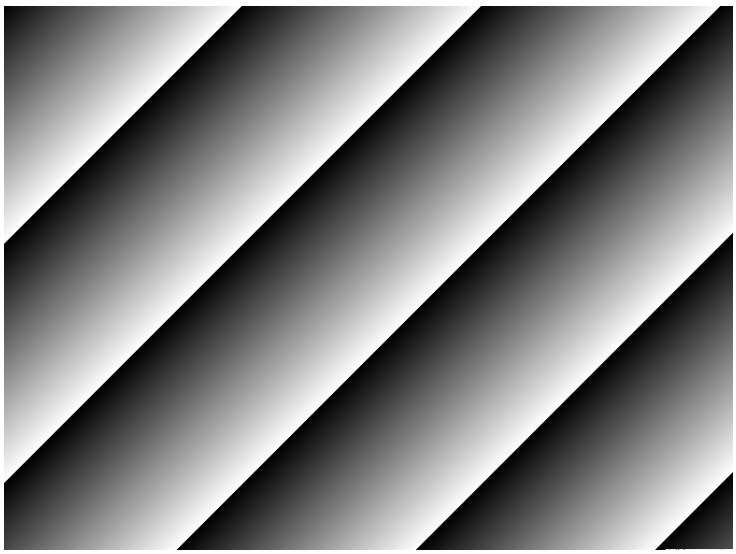


Fig. 129: Test Image One

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.

	When cameras with a special sensor reache a temperature of 80 °C (+176 °F), the so-called over temperature, it will automatically enter an over temperature idle mode. In this mode, the camera will no longer acquire images but will display test image 2. For more information about the over temperature idle mode and how to leave it, see Section 1.7.3.3 on page 58.

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.

(*) For acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50 camera models it is a 10-bit test image.

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; exception: see * above). 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 130, 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.

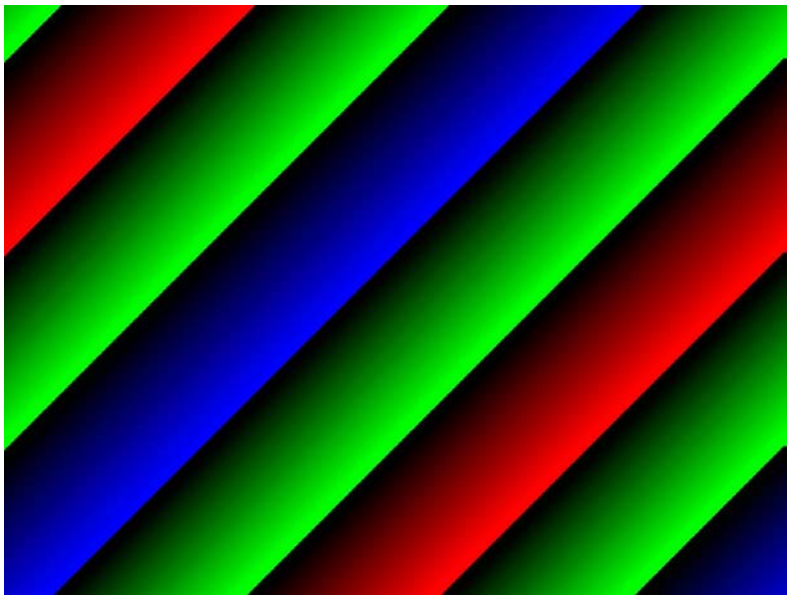


Fig. 130: Test Image Six

9.18 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 Information Parameter	Read or Write Status	Explanation
DeviceVendorName	read only	Contains the camera vendor’s name.
DeviceModelName	read only	Contains the model name of the camera.
DeviceManufacturerInfo	read only	Can contain some information about the camera manufacturer. This string usually indicates "none".
DeviceVersion	read only	Contains the device version number for the camera.
DeviceFirmwareVersion	read only	Contains the version of the firmware in the camera.
DeviceID	read only	Contains the serial number of the camera.
DeviceUserID	read / write	Used to assign a user defined name to a device. This name will be displayed in the Basler pylon Viewer and the Basler pylon IP Configurator. The name will also be visible in the "friendly name" field of the device information objects returned by pylon’s device enumeration procedure.
DeviceScanType	read only	Contains the scan type of the camera, for example, area scan.
SensorWidth	read only	Contains the physical width of the sensor in pixels.
SensorHeight	read only	Contains the physical height of the sensor.
WidthMax	read only	Indicates the camera’s maximum area of interest (AOI) width setting.
HeightMax	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 DeviceUserID 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 DeviceUserID:

```
// 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 DeviceUserID.

You can use the Basler pylon IP Configurator to read or write the DeviceUserID.

For more information about the pylon API, the pylon Viewer, and the pylon IP Configurator, see Section 3 on page 63.

9.19 User-Defined Values

The camera can store five "user defined values". These five values are 32-bit signed integer values that you can set and read as desired. They serve as convenient storage locations for the camera user and have no impact on the operation of the camera.

The values are designated as Value 1, Value 2, Value 3, Value 4, and Value 5.

Setting User-Defined Values

To set a user-defined value:

1. Set the UserDefinedValueSelector to Value1, Value2, Value3, Value4, or Value5.
2. Set the UserDefinedValue parameter to the desired value for the selected value.

You can use the pylon API to set the UserDefinedValueSelector and the UserDefinedValue 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 63.

9.20 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, factory sets, and user sets. In addition you can designate a startup set.

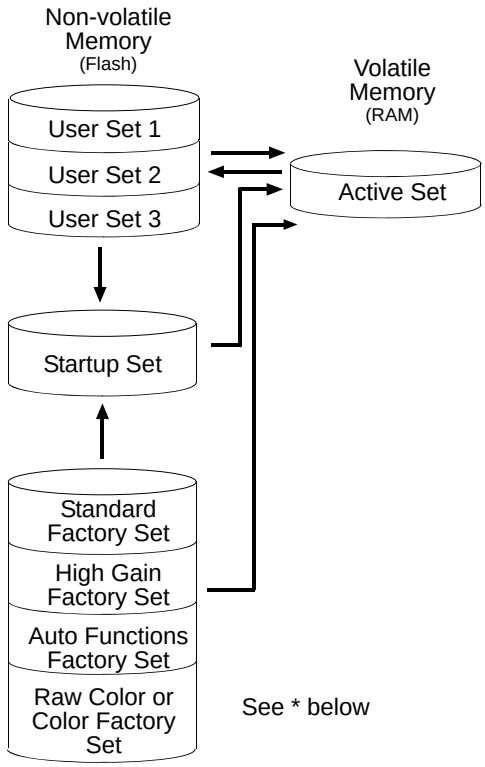


Fig. 131: Configuration Sets

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.

Factory Sets

When a camera is manufactured, numerous tests are performed on the camera and four factory optimized sets are determined. The four factory optimized sets are:

Factory Set [Availability in cameras, see table below]	**What is it?**	**Camera Models with Color Factory Set**	**Camera Models with Raw Color Factory Set**
Standard factory set	Factory set that the camera wakes up with. The basic settings of the standard factory set depends on the camera modell. See right. Gain is set to a low value, all auto functions are set to off.	Color enhancement **disabled**	Color enhancement **enabled**
High gain factory set	Similar to the standard factory set, but gain is set to + 6 dB.	x	x
Auto functions factory set	Similar to standard factory set, but gain auto and 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.	x	x

Factory Set [Availability in cameras, see table below]	**What is it?**	**Camera Models with Color Factory Set**	**Camera Models with Raw Color Factory Set**
Color factory set	Optimized to yield the best color fidelity with daylight lighting.	x	n.a.
Raw color factory set (*)	Color optimization settings are disabled.	n.a.	x
For information about which camera model has color factory set or a raw color factory, see table below. * A camera has either a color factory set or a raw color factory set available. Depends on the camera model. n.a. = not available			

Camera Models with Color Factory Set	**Camera Models with Raw Factory Set**
All other models	acA640-300, acA800-200, acA1300-75, acA1920-40, acA1920-50

The factory sets are saved in permanent files in the camera's non-volatile memory.

Factory sets

- cannot be changed
- are not lost when the camera is reset or switched off
- can 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.
- can be adapted and saved as user sets.

You can load one of the factory sets into the camera's active set.

For information about

- the color factory set and the raw color factory set, see Section 7.4.7.1 on page 238 and Section 7.4.7.2 on page 238
- auto functions, see Section 9.14 on page 328.
- loading a factory set into the active set, see Section 9.20.2 on page 363
- designating which set will be the startup set, see Section 9.20.3 on page 364.

User Sets

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 (e.g. a modified factory 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.

For information about

- loading a saved user set into the active set, see Section 9.20.2 on page 363
- designating which set will be the startup set, see Section 9.20.3 on page 364.

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 a factory 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.

9.20.1 Saving a User Set

To save the currently active set into a user set:

1. Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
2. Set the UserSetSelector parameter to User Set 1, User Set 2, or User Set 3.
3. Execute a UserSetSave 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 UserSetSelector parameter and execute the UserSetSave 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.

9.20.2 Loading a User Set or a Factory 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 a user set or a factory set into the camera's active set.

To load a factory set or a user set into the active set:

1. Set the UserSetSelector to User Set 1, User Set 2, User Set 3 or to one of the factory sets.
2. Execute a UserSetLoad command to load the selected set into the active set.

You can set the UserSetSelector parameter and execute the UserSetLoad 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 a factory 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 standard factory set (UserSetSelector_Default parameter) 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 set 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 63.

9.20.3 Designating the Startup Set

You can designate one of the following sets stored in the camera’s non-volatile memory to be the "startup set":

- one of the factory sets:
  the Standard factory set, the High Gain factory set, the Auto Functions factory set, the Color factory set (for models without GPIO) or the Raw Color factory set (for models with GPIO)]
- one of the user sets.

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.

To designate the startup set:

1. Set the UserSetDefaultSelector parameter to User Set 1, User Set 2, User Set 3 or to one of the factory sets.

You can set the UserSetDefaultSelector parameter 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 63.

10 Chunk Features

Available for
All models

10.1 What are Chunk Features?

In most cases, enabling a camera feature will 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 (see Section 10.2 on page 366).

10.2 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:

1. Set the ChunkModeActive parameter to true.

You can set the ChunkModeActive 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

- ChunkOffsetX parameter.
- ChunkOffsetY parameter.
- ChunkWidth parameter.
- ChunkHeight parameter.
- ChunkPixelFormat parameter.
- ChunkDynamicRangeMin
- ChunkDynamicRangeMax

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).

10.3 Frame Counter

The frame counter feature numbers frames sequentially as they are acquired. When the feature is enabled, a chunk is added to each frame containing the value of the counter.

The frame counter is a 32-bit value. The counter starts at 0 and increments by 1 for each acquired frame. The counter counts up to 4294967295 unless it is reset before (see below). After reaching the maximum value, the counter will reset to 0 and then continue counting.

Be aware that, if the camera is acquiring frames continuously and continuous capture is stopped, several numbers in the counting sequence may be skipped. This happens due to the internal image buffering scheme used in the camera.

The chunk mode must be active before you can enable the frame counter feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the frame counter chunk:

1. Use the ChunkSelector parameter to select the FrameCounter chunk.
2. Use the ChunkEnable 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.

To retrieve the frame counter information:

1. 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 63.

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:

1. Configure the frame counter reset by setting the counter selector to Counter2 and setting the counter event source to FrameStart.
2. Set the CounterResetSource parameter to Line1, software, or Off.
3. Execute the command if using software as the counter reset source.

You can set the frame CounterReset 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 63.
- about using line 1 as the source signal for a frame counter reset, see Section 5.10.1 on page 98.

10.4 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:

1. Use the ChunkSelector to select the TimeStamp chunk.
2. Use the ChunkEnable 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.

To retrieve the time stamp information:

1. Read the value of the ChunkTimeStamp parameter.

You can set the ChunkSelector and ChunkEnable 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 63.

10.5 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:

1. Use the ChunkSelector to select the TriggerInputCounter chunk.
2. Use the ChunkEnable 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.

To retrieve the trigger input counter information:

1. Read the value of the ChunkTriggerInputCounter 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 63.

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:

1. Configure the trigger input counter reset by setting the counter selector to Counter1 and setting the counter event source to FrameTrigger.
2. Set the counter reset source to Line1, software, or Off.
3. Execute the command if using software as the counter reset source.

You can set the TriggerInputCounterReset 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 63.
- using line 1 as the source signal for a trigger input counter reset, see Section 5.10.1 on page 98.

10.6 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 132, 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 high at the time of triggering.

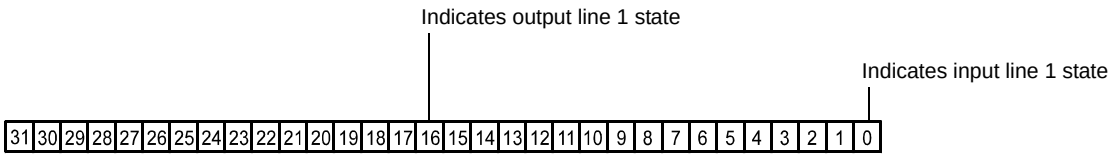


Fig. 132: Line Status All Parameter Bits (Camera Models **Without** GPIO)

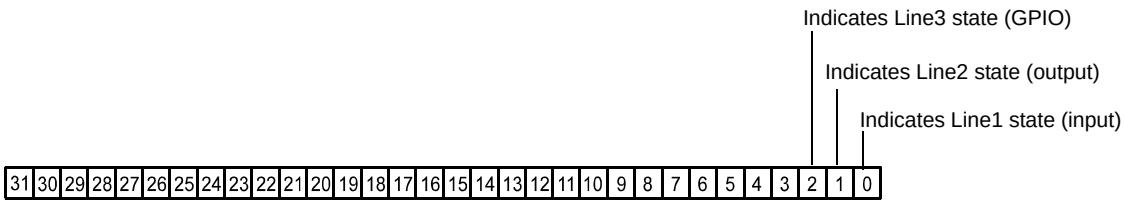


Fig. 133: Line Status All Parameter Bits (Camera Models **With** GPIO)

	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:

1. Use the ChunkSelector to select the LineStatusAll chunk.
2. Use the ChunkEnable 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.

To retrieve the line status all information:

1. Read the value of the ChunkLineStatusAll parameter.

You can set the ChunkSelector and ChunkEnable parameter values 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 63.

10.7 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 134, 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.

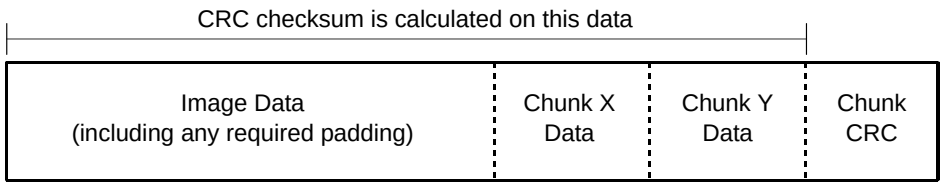


Fig. 134: CRC Checksum

	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:

1. Use the ChunkSelector to select the CRC chunk.
2. Use the ChunkEnable 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 63.

10.8 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 from page 273 on.

	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:

1. Use the ChunkSelector to select the Sequence Set Index chunk.
2. Use the ChunkEnable 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.

To retrieve the sequence set index information:

1. Read the value of the ChunkSequenceSetIndex parameter.

You can set the ChunkSelector and ChunkEnable parameter values 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.

11 Troubleshooting and Support

This chapter outlines the resources available to you, if you need help working with your camera.

11.1 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 the Basler 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.

11.2 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 the Basler website: www.baslerweb.com

11.3 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.
- ☐ Every time.
- ☐ Regularly when: _____
- ☐ Occasionally when: _____

8 How severe is the problem?

- ☐ Camera can still be used.
- ☐ Camera can be used after I take this action: _____
- ☐ Camera can no longer be used.

9 Did your application ever run without problems?

- ☐ Yes
- ☐ No

10 Parameter set

It is very important for Basler technical support to get a copy of the exact camera parameters that you were using when the problem occurred.

To make note of the parameters, use Basler's pylon Viewer.

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 and
- installing the network drivers,

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

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.
EnableResend - 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.

PacketTimeout - The PacketTimeout 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 PacketTimeout parameter is set to a longer time interval than the time interval set for the interpacket delay.

FrameRetention - The FrameRetention 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 63.

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 pylon for Windows* (AW000611).

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

- The threshold resend mechanism
- The timeout resend mechanism

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

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

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

EnableResend - Enables the packet resend mechanisms.

If the EnableResend 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.

ReceiveWindowSize - Sets the size of the receive window.

A.2.2 Threshold Resend Mechanism Parameters

The threshold resend request mechanism is illustrated in Figure 135 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.

DIAGRAM IS NOT DRAWN TO SCALE

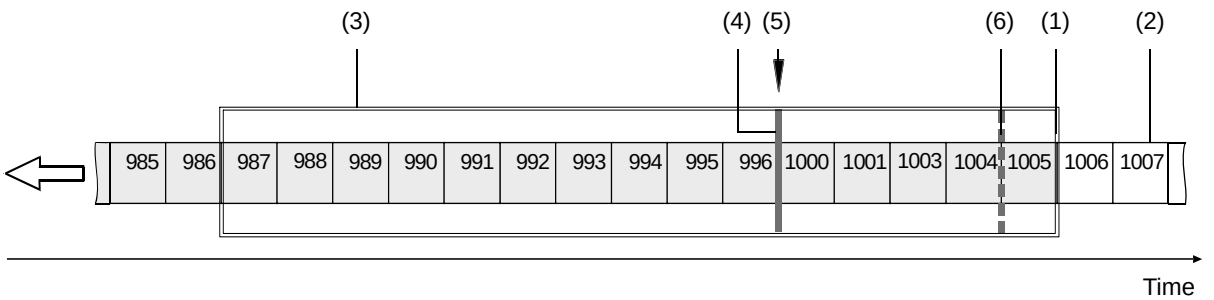


Fig. 135: Example of a Receive Window with Resend Request Threshold & Resend Request Batching Threshold

(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.

ResendRequestThreshold - This parameter determines the location of the resend request threshold within the receive window as shown in Figure 135. The parameter value is in per cent of the width of the receive window. In Figure 135 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 135). 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 135, 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.

ResendRequestBatching - This parameter determines the location of the resend request batching threshold in the receive window (Figure 135). 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 135 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 ResendRequestBatching 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 135). 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 136 where the following assumptions are made:

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

DIAGRAM IS NOT DRAWN TO SCALE

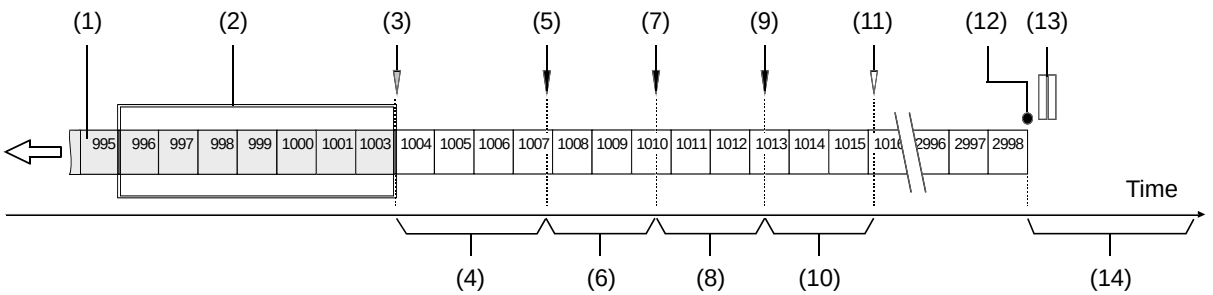


Fig. 136: Incomplete Stream of Packets and Part of the Resend Mechanism

(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 ResendTimeout 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 ResendResponseTimeout 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 ResendResponseTimeout 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 ResendResponseTimeout parameter.

(11) Because the maximum number of resend requests has been sent and the last resend responset 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 PacketTimeout parameter.

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

ResendTimeout - The ResendTimeout 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.

ResendRequestResponseTimeout - The ResendRequestResponseTimeout 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 MaximumNumberResendRequests 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.

PacketTimeout - The PacketTimeout 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 137 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 137). In addition, resend request batching will not occur.

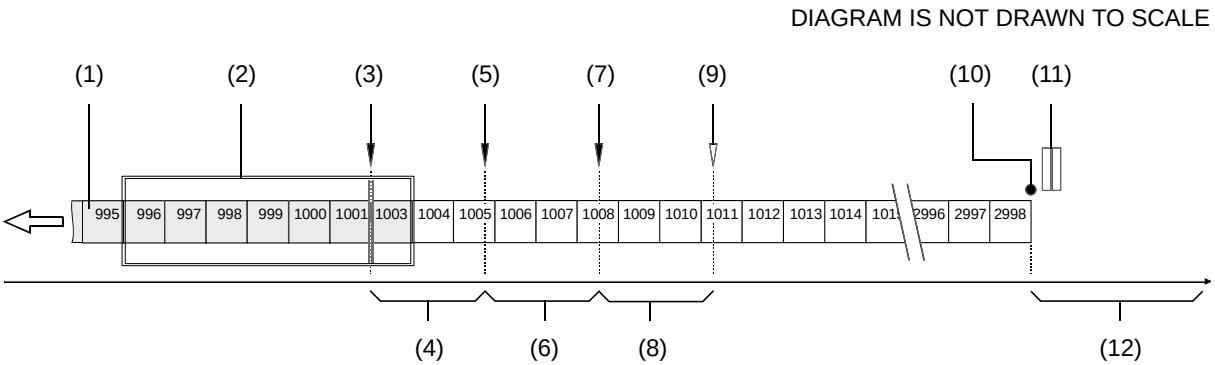


Fig. 137: Combination of Threshold Resend Mechanism and Timeout Resend Mechanism

(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 ResendResponseTimeout parameter.

(5) The ResendTimeout 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 ResendResponseTimeout 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 ResendResponseTimeout 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 PacketTimeout 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 63.

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.
   A **LAN Connection Properties** window will open.
3. Click the **Configure** button.
   An **Adapter Properties** window will open.
4. 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.

ReadTimeout - 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.

WriteTimeout - 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.

HeartbeatTimeout - 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 63.

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.

PayloadSize (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.

StreamChannelSelector (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.

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

Inter-PacketDelay (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.

FrameTransmissionDelay (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.

BandwidthAssigned (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:

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

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

Y = number of bytes in each packet

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

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

BandwidthReserve (read/write)

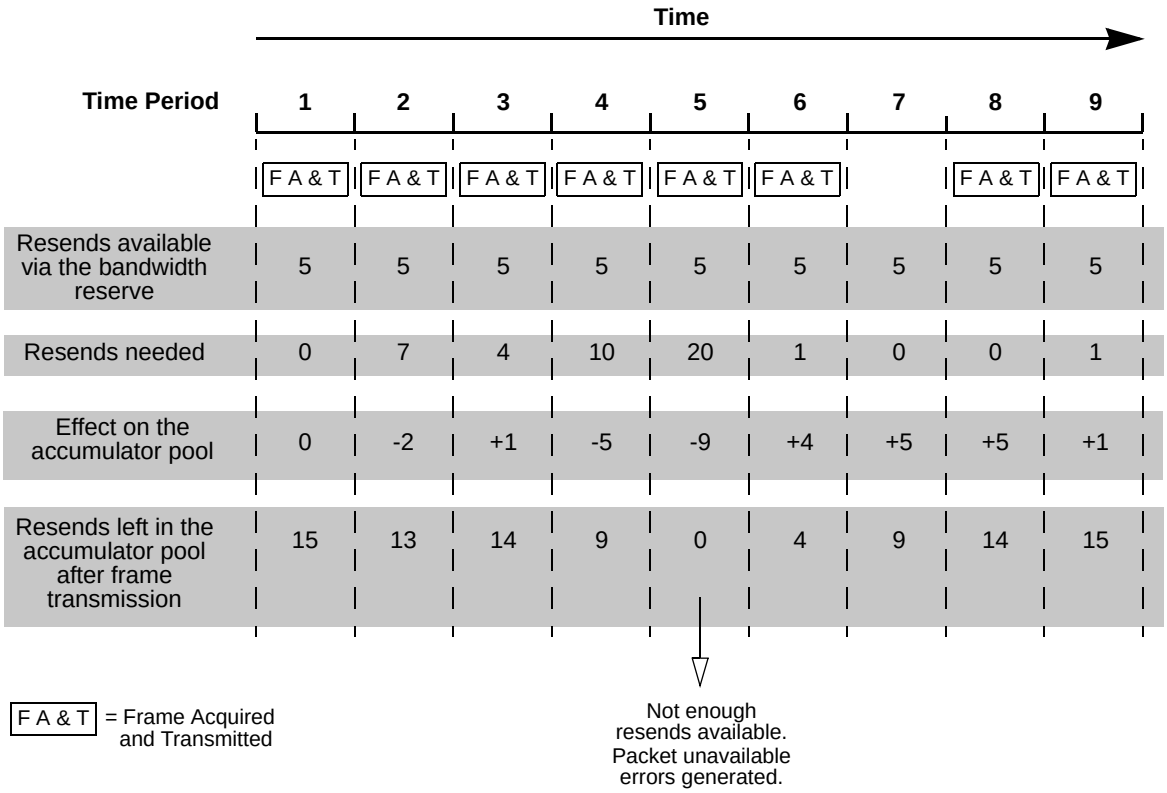
Used to reserve a portion of the assigned bandwidth for packet resends and for the transmission of control data between the camera and the host PC. The setting is expressed as a percentage of the Bandwidth Assigned parameter. For example, if the Bandwidth Assigned parameter indicates that 30 MByte/s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

BandwidthReserveAccumulation (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 BandwidthReserveAccumulation 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.

FrameMaxJitter (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.

DeviceMaxThroughput (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 AcquisitionFrameRateAbs 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 DeviceCurrentThroughput parameter indicates the bandwidth needed to transmit the actual image data and chunk data. The BandwidthAssigned 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.

ResultingFrameRate (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 AcquisitionFrameRateAbs parameter has been used to set the camera's frame rate, the ResultingFrameRate parameter will show the AcquisitionFrameRateAbs parameter setting. If software or hardware triggering is being used to control the camera's frame rate, the ResultingFrameRate 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 63.

Note

For **acA1600-60, acA1920-50 cameras**:

On the initial wake-up after delivery the Basler acA1600-60 and acA1920-50 cameras have default transport layer settings that don't allow to reach the specified maximum possible frame rate.

If you want to obtain the maximum possible frame rate, change the values of the default transport layer parameters in pylon Viewer as indicated in Table 54.

Parameter	Default Setting (Wake-up Value)	Value for Reaching 60 fps
PacketSize	1500	4000
BandwidthReserve	10	5

Table 54: Transport Layer Settings (acA1600-60, acA1920-50)

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 138.

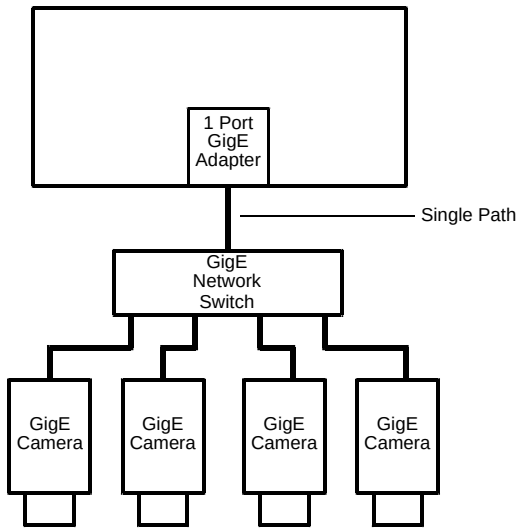


Fig. 138: Multiple Cameras on a Network

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 BandwidthAssigned 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-PacketDelay 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-PacketDelay parameter on each camera, you can check the sum of the BandwidthAssigned 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 **ReceiveDescriptors** parameter is set to its maximum value and the **InterruptModerationRate** parameter is set to **Extreme**.

  Also make sure the **Speedand 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 PacketSize 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 PacketSize 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 PacketSize 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 BandwidthReserve parameter for each camera.

The BandwidthReserve 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 PayloadSize parameter and the PacketSize parameter for each camera. You will also need to know the frame rate (in frames/s) at which each camera will operate.

$$\text{Bytes/Frame} = \left\lfloor \left\lceil \frac{\text{Payload Size}}{\text{Packet Size}} \right\rceil_1 \times \text{Packet Overhead} \right\rfloor + \lceil \text{Payload Size} \rceil_4 + \text{Leader Size} + \text{Trailer Size}$$

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

Where:

Packet Overhead = 72 (for a GigE network)
78 (for a 100 MBit/s network)

Leader Size = Packet Overhead + 36 (if chunk mode is not active)
Packet Overhead + 12 (if chunk mode is active)

Trailer Size = Packet Overhead + 8

$\lceil x \rceil_1$ means round up x to the nearest integer

$\lceil x \rceil_4$ means round up x to the nearest multiple of 4

Step 5 - Calculate “data bandwidth assigned” to each camera.

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

You can use the formula below to determine the actual amount of assigned bandwidth that is available for data transmission. To use the formula, you will need to know the current value of the BandwidthAssigned parameter and the BandwidthReserve 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-PacketDelay. If you increase the Inter-PacketDelay 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-packetDelay 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 BandwidthAssigned parameter. The value is in Byte/s. (Make sure that you determine the value of the BandwidthAssigned parameter after you have made any adjustments described in the earlier steps.)
2. Find the sum of the current BandwidthAssigned 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 6.12 on page 196.
- the AOI, see Section 9.5 on page 261.

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 5.7.1 on page 83 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 9.19 on page 359 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
AW000893**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 9.20 on page 360. Removed the Tungsten and Daylight 5000K Color Transformation Selector parameters from Section 8.4.4 on page 172. Added Section 9.8 on page 273 describing the sequencer feature. Added a note that auto functions will not work when the sequencer is enabled in Section 9.14 on page 328. Added the color factory set in Section 9.8 on page 273. Added Section 11.8 on page 323 describing the sequence set index chunk and modified Section 11.2 on page 310 accordingly.
AW000893**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 9.8 on page 273.

AW000893**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 9.1.1 on page 245
		  ▪ Section 9.2.1 on page 254
		▪ 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 76: 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 9.14.3 on page 330.
		▪ Integrated note on Sequence Set Index chunk in Section 9.8 on page 273.
		▪ Adapted vertical binning description for the acA 2500-14 gm in Section 9.9 on page 306.
		▪ Integrated two new events (acquisition start and frame start) in Section 9.16 on page 350.

Doc. ID Number	Date	Changes

AW000893**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 48 on page 85.
		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 9.1.1 on page 245
		  ▪ Section 9.2.1 on page 254
		  ▪ Section 9.5 on page 261
		▪ Integrated minimum output pulse width feature, see Section 5.11.2 on page 104.
		▪ Integrated note on the slowdown of the sequencer feature for the acA2500-14 in Section 9.8.
		Section 11
		Replaced Z-modem by X-modem in Section 11.7 on page 321.

Doc. ID Number	Date	Changes

AW000893**15**000	6 Jun 2012	**Section 5**
		▪ Replaced figures in Section 5.6 on page 80 by simplified versions.
		▪ Replaced figures in Section 5.7 on page 83 by simplified versions.
		▪ Added notes in Section 5.5 on page 78, Section 5.6.1 on page 80, Section 5.7.1 on page 74:
		  ▪ 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 9.6 on page 266 describing the stacked zone imaging feature.
		▪ Integrated the color binning feature for the acA2500-14gc (Section 9.9.1 on page 308).
		▪ Integrated section on binning's effect on the stacked zone imaging feature in Section 9.9.3 on page 310.

AW000893**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 47. **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 45 on page 82. **Section 6** ▪ Corrected description of Section 6.2.5.3 on page 110. ▪ Modified the Line Debouncer Time Abs parameter value to 20 µs in Section 6.1.2 on page 99. **Section 7** Inserted information in Section 6.12 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
AW000893**16**000	8 Aug 2012	**Section 10** ▪ Updated the minimum gain setting for the acA2000 and acA2040 models in Table 25 on page 205 and Table 26 on page 206. ▪ Inserted information in Section 9.6 on page 266 that the stacked zone imaging feature on the acA2000-50 and acA2040-25 increases the camera's frame rate. ▪ In the sequencer feature Section 9.8: ▪ 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 9.10.1 on page 313. ▪ Added Section 9.14.3.1 on page 331 describing the assignment of an auto function to an auto functio<<<< AOI. ▪ Added the gray value adjustment damping feature in Section 9.14.6 on page 340.
AW000893**17**000	15 Aug 2014	Updated the mail addresses and the Asian contact address (page 4). Removed the "in preparation" from the term "UL (in preparation) in the tables in Section 1.2 on page 2. Updated the sensor size of the acA1600-20gm (height: 1626 instead of 1628). Added warnings on avoiding dust on the sensor in Section 1.8 on page 59. Changed the name of the acA640-100gm/gc camera model to acA640-120gm/gc. Renaming throughout the manual: Changed ▪ pylon driver package to Basler pylon Camera Software Suite ▪ IP Configuration Tool to IP Configurator ▪ pylon Viewer Tool to pylon Viewer Made the appropriate changes throughout the manual to add the new acA1280-60gm/gc, acA1300-22gm, acA1300-60gm/gm/gmNIR, acA1600-60gm/gc, acA1920-25gm/gc, acA2000-50gm/gc - CMV2000-V3, acA2000-50gmNIR - CMV2000-V3, acA2040-25gm/gc - CMV2000-V3, and acA2040-25gmNIR - CMV4000-V3 camera models. Corrected the pixel data format for the acA2500-14 camera model throughout the manual (changed BG to GB). Added LZ4 licensing information in Section 1.5.2 on page 53. Removed absolute max. voltages from Table 19 on page 81, from Table 20 on page 84, from Section 5.7.2 on page 65, Table 20 on page 84, from Section 5.7.1 on page 83. Corrected the minimum operating voltage value to 10.8 V in Section 1.8 on page 59 and in Section 5.5 on page 78. Added information on the reverse X and reverse Y features in Section 7.1.1 on page 210. Removed sub-section "Pixel Data Formats for Mono Cameras" in Section 8. To be continued.

Doc. ID Number	Date	Changes
AW000893**17**000	15 Aug 2014	Adapted "Remove Parameter Limits" Section 9.3 on page 255. Added the "Error Codes" Section 9.7 on page 271. Adapted "Binning" Section 9.9 on page 306. Added the following sections in Section 9.9.3 on page 310: ▪ "Possible Image Distortion" on page 311. ▪ "Binning's Effect on Decimation" on page 311 Adapted "Decimation" Section 9.10 on page 313. Added the horizontal decimation feature Section 9.10.2 on page 315. Added general information on the mirror imaging features in Section 9.12 on page 319. Added the new reverse Y feature in Section 9.12.2 on page 321. Deleted "Some auto functions use their own individual Auto Function AOI and..." and modified "... some auto functions share a single Auto Function AOI", to "Some auto functions *always* share an Auto Function AOI" (Section 9.14.1 on page 328). Adapted the text in Section 9.14.3.2 on page 332, on page 275. Changed IR-cut filter to IR cut filter. Updated Section 9.19 on page 359: now 5 user defined values. Added note on page 401 on how to obtain the maximum frame rate for the acA1600-60.
AW000893**18**000	13 Feb 2015	Minor corrections. Corrected subpart J to subpart B in the FCC section at the back of the front page. Made the appropriate changes throughout the manual to add the new acA1920-50gm/gc, acA3800-10gm/gc, and acA4600-7gc camera models (prototype status). Removed the following models from the User's Manual: acA2000-50gm/gc - CMV2000-V3, acA2000-50gmNIR - CMV2000-V3, acA2040-25gm/gc - CMV2000-V3, and acA2040-25gmNIR - CMV4000-V3 camera models. Replaced plastic seal by plastic cap throughout the manual. Integrated precautions concerning SELV and LPS in Section 1.8 on page 59. Correction: Removed the color video output formats for NIR camera models in Section 1.2 on page 2. Updated Section 3.1.3 on page 64. Added Section 6.2.3 on page 104 (sync user output). Added note in Section 6.4.1.2 on page 137. Added the median filter feature for the acA1280-60, acA1300-60, and the acA1600-60 cameras in Section 9.15 on page 349. Restructured the reverse Y feature section (Section 9.12.2 on page 321).

Doc. ID Number	Date	Changes
AW000893**19**000	10 Apr 2015	Internal release
AW000893**20**000	8 Jun 2015	Integrated the I/O Control chapter sections into the Physical Interface chapter. Changed 30 V to 24 VDC in Figure 46 on page 82. Updated Section 9.1 on page 244 (gain). Corrected the black level raw parameter range of the acA1920-25 in Section 9.2.1 on page 254. It's the same as for the acA2500-14. Image Area of Interest (AOI) section: Corrected the increment examples for width and height (all camera models) on page 262. Rearranged the binning section (Section 9.9 on page 306).

Doc. ID Number	Date	Changes
AW000893**21**000	16 Oct 2015	Throughout the manual: ▪ Changed 'sensor size' to resolution. ▪ Changed camera parameter names to camel case (e.g. LineSelector parameter instead of Line Selector). ▪ Added information concerning the prototype cameras acA3800 and acA4600. ▪ Corrected the SELV and LPS precautions. ▪ Updated name of *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611). ▪ Added information about the following cameras: acA640-300, acA800-200, acA1300-75, acA1920-40 (prototype cameras). Modifications in technical specifications tables: ▪ Re-grouped the columns in the technical specifications table in chapter 1. ▪ Changed the expression "external trigger signal" to "hardware trigger" (Synchronization). ▪ Set Exposure Time Control instead of Exposure Time. ▪ Added the "Effective Sensor Diagonal" line for each camera model in the General Specifications table in chapter 1. ▪ Rectified the resolution values for the acA1920-50gc camera model. ▪ Rectified the camera power requirement values of the acA1920-50. ▪ Removed the CS-mount option for the acA3800-10 and acA4600-7 camera models. CS-mount is not available any more. ▪ Adapted the CS-mount availability for certain camera models (acA645-100, acA750-30, acA780-75, acA1280-60, acA1600-20, acA1600-60, acA1920-25). ▪ Adapted notice box on inappropriate code on page 60. Added the "Getting Information about the Temperature Status via Events" section on page 58. Modifications in Physical Interface chapter: ▪ Added "Temporal Performance of I/O Lines" section from page 93 on. ▪ Removed Response Time sections. ▪ Added Section 5.11.6 on page 111 ("Setting and Checking the State of All User Settable Synchronous Output Signals") Added note on "low level" code on page 121. Added description of the exposure time offset for the acA2000-50, acA2040-25 models. Also valid for the acA640-300, acA800-200, acA1300-75, acA1920-30 models: "Trigger Width Exposure Mode with Special Exposure Time Offset" on page 145. Corrected the maximum exposure time value for the acA2000-50 and acA2040-25 models to 10000000 µs (10 s; before 1 s). Added note on ERS mode, flash window and exposure time on page 163. Added note on min. exposure time required to be able to use a flash window in ERS mode (Section 6.7.2.1 on page 167). Updated Figure 87 on page 177. Corrected Exposure Active description, inlcuding Figure 88 on page 178: The Exposure Active signal goes high when the exposure for the first line in a frame begins and goes low when the exposure for the last line ends. Added information that the color enhancement feature should be used with the black level wake up value (Section 7.4.6 on page 236). Added pixel formats in the code snippet section on page 240.

Doc. ID Number	Date	Changes
AW000893**21**000 Continued from previous page	16 Oct 2015	Added note that the use of the Reverse X and Reverse Y features changes the Bayer color filter alignment. Correction: Changed $Gain_{dB}$ = 0.01 x 200 = 200 to $Gain_{dB}$ = 0.01 x 200 = 2 (Table 35 on page 247). Changed the pixel data formats of the acA1920-50gc from Bayer RG 8 and Bayer RG 12 (instead of BG). Added the raw color factory set description for cameras with GPIO and adapted the color factory set description to the fact that it is only available for cameras without GPIO. Added the Center X and Center Y feature description in Section 9.5.1 on page 265. Updated the Digital Shift feature description in Section 9.4 on page 257. Added new error code for over temperature (available for cameras with GPIO). Section 9.7 on page 271. Updated the list of parameters that are included in each sequence set on page 275. Added the "Scaling" feature description (9.11 on page 317). Updated the "Auto Function Profile" section on page 341. Added the "Balance White Adjustment Damping" section on page 343. Updated the "Pattern Removal" Section 9.14.10 on page 345. Adapted the event notification description - added two new events in Section 9.16 on page 350. Removed the mathematical expressions in the test image descriptions (Section 9.17). Updated Section 9.20: Modifications throughout the whole section: replaced Default Set Selector with Configuration Set Selector. Added Raw Color factory set for camera models with GPIO. Deleted section "Selecting a Factory Setup as the Default Set".

Index

A

B

C

D

E

F

G

H

I

L

M

O

P

R

S

T

U

V

W

Z