Basler pilot

USER’S MANUAL FOR GigE VISION CAMERAS

Document Number: AW000151
Version: 16   Language: 000 (English)
Release Date: 17 June 2009

For customers in the U.S.A.

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

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

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

For customers in Canada

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

Pour utilisateurs au Canada

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

Life Support Applications

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

Warranty Note

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

Contacting Basler Support Worldwide

Europe:

Basler AG
An der Strusbek 60 - 62
22926 Ahrensburg
Germany

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

bc.support.europe@baslerweb.com

Americas:

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

Tel.: +1-877-934-8472
Fax.: +1-610-280-7608

bc.support.usa@baslerweb.com

Asia:

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

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

bc.support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1 Specifications, Requirements, and Precautions

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

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

1.1 Models

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

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

- standard housing
- 90° head housing

The names for the housing variants other than the standard housing are appended to the camera's name, e.g. piA640-210gm/gc 90° head.

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

1.2 General Specifications

Specification	piA640-210gm/gc	piA1000-48gm/gc	piA1600-35gm/gc
Sensor Size (H x V pixels)	gm: 648 x 488 gc: 646 x 486	gm: 1004 x 1004 gc: 1000 x 1000	gm: 1608 x 1208 gc: 1604 x 1204
Sensor Type	Kodak KAI-0340M/CM	Kodak KAI-1020M/CM	Kodak KAI-2020M/CM
	Progressive scan CCD		
Optical Size	1/3"	2/3"	1"
Pixel Size	7.4 µm x 7.4 µm	7.4 µm x 7.4 µm	7.4 µm x 7.4 µm
Max. Frame Rate (at full resolution)	210 fps	48 fps	35 fps
Mono/Color	All models available in mono or color		
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)		
Pixel Data Formats	Mono Models: Mono 8 (equivalent to DCAM Mono 8) Mono 16 (equivalent to DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (equivalent to DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 (equivalent to DCAM Mono 8) Bayer GB 8 (equivalent to DCAM Raw 8) Bayer GB 16 (equivalent to DCAM Raw 16) Bayer GB 12 Packed YUV 4:2:2 Packed (equivalent to DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		
ADC Bit Depth	12 bits		
Synchronization	Via external trigger signal or via software		
Exposure Control	Programmable via the camera API		
Camera Power Requirements	+12 to +24 VDC, (min. +11.3 VDC, absolute max. +30.0 VDC), < 1% ripple		
	4.5 W @ 12 VDC	4.2 W @ 12 VDC	4.8 W @ 12 VDC
I/O Ports	2 opto-isolated input ports and 4 opto-isolated output ports		
Lens Adapter	C-mount		

Specification	piA640-210gm/gc	piA1000-48gm/gc	piA1600-35gm/gc
Size (L x W x H) (standard housing)	86.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 98.5 mm x 44 mm x 29 mm (with lens adapter and connectors)		
(90° head housing)	104.7 mm x 44 mm x 29 mm (without front module or connectors) 110 mm x 44 mm x 41.8 mm (with front module and connectors)		
Weight (standard housing)	~ 220 g (typical)		
(90° head housing)	~ 240 g (typical)		
Conformity	CE, FCC, GenICam, GigE Vision; IP30		

Table 1: General Specifications

Specification	piA1900-32gm/gc
Sensor Size (H x V pixels)	gm: 1928 x 1084 gc: 1926 x 1082
Sensor Type	Kodak KAI-2093M/CM
	Progressive scan CCD
Optical Size	1"
Pixel Size	7.4 µm x 7.4 µm
Max. Frame Rate (at full resolution)	32 fps
Mono/Color	All models available in mono or color
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)
Pixel Data Formats	Mono Models: Mono 8 (equivalent to DCAM Mono 8) Mono 16 (equivalent to DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (equivalent to DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 (equivalent to DCAM Mono 8) Bayer GB 8 (equivalent to DCAM Raw 8) Bayer GB 16 (equivalent to DCAM Raw 16) Bayer GB 12 Packed YUV 4:2:2 Packed (equivalent to DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	12 bits
Synchronization	Via external trigger signal or via software
Exposure Control	Programmable via the camera API
Camera Power Requirements	+12 to +24 VDC, (min. +11.3 VDC, absolute max. +30.0 VDC), < 1% ripple
	4.4 W @ 12 VDC
I/O Ports	2 opto-isolated input ports and 4 opto-isolated output ports
Lens Adapter	C-mount

Specification	**piA1900-32gm/gc**
Size (L x W x H) (standard housing)	86.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 98.5 mm x 44 mm x 29 mm (with lens adapter and connectors)
(90° head housing)	104.7 mm x 44 mm x 29 mm (without front module or connectors) 110 mm x 44 mm x 41.8 mm (with front module and connectors)
Weight (standard housing)	~ 220 g (typical)
(90° head housing)	~ 240 g (typical)
Conformity	CE, FCC, GenICam, GigE Vision, IP30

Table 2: General Specifications

Note

The sensor characteristics of the piA1900-32gm/gc cameras do not entirely conform to the quality standards generally adhered to by Basler. The sensitivity to light for clusters of up to six contiguous pixels may deviate significantly from the sensitivities of normal pixels.

Specification	**piA2400-12gm/gc**	**piA2400-17gm/gc**
Sensor Size (H x V pixels)	gm: 2456 x 2058 gc: 2454 x 2056	
Sensor Type	Sony ICX625ALA/AQA	
	Progressive scan CCD	
Optical Size	2/3"	
Pixel Size	3.45 µm x 3.45 µm	
Max. Frame Rate (at full resolution)	12 fps	17 fps
Mono/Color	All models available in mono or color	
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)	
Pixel Data Formats	Mono Models: Mono 8 (equivalent to DCAM Mono 8) Mono 16 (equivalent to DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (equivalent to DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed Color Models: Mono 8 (equivalent to DCAM Mono 8) Bayer BG 8 (equivalent to DCAM Raw 8) Bayer BG 16 (equivalent to DCAM Raw 16) Bayer BG 12 Packed YUV 4:2:2 Packed (equivalent to DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed	
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal or via software	
Exposure Control	Programmable via the camera API	
Camera Power Requirements	+12 to +24 VDC, (min. +11.3 VDC, absolute max. +30.0 VDC), < 1% ripple	
	5.4 W @ 12 VDC	5.9 W @ 12 VDC
I/O Ports	2 opto-isolated input ports and 4 opto-isolated output ports	
Lens Adapter	C-mount	

Specification	piA2400-12gm/gc	piA2400-17gm/gc
Size (L x W x H)		
(standard housing)	86.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 98.5 mm x 44 mm x 29 mm (with lens adapter and connectors)	
(90° head housing)	104.7 mm x 44 mm x 29 mm (without front module or connectors) 110 mm x 44 mm x 41.8 mm (with front module and connectors)	
Weight		
(standard housing)	~ 220 g (typical)	
(90° head housing)	~ 240 g (typical)	
Conformity	CE, FCC, GenICam, GigE Vision, IP30	

Table 3: General Specifications

1.3 Spectral Response for Mono Cameras

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

Note

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

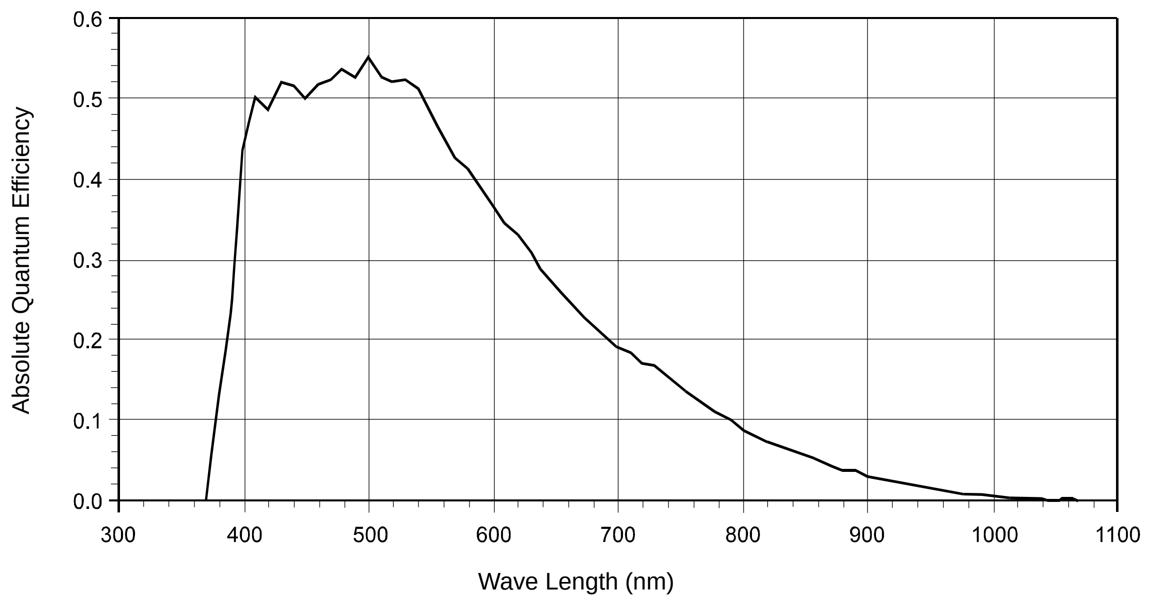


Fig. 1: piA640-210gm Spectral Response

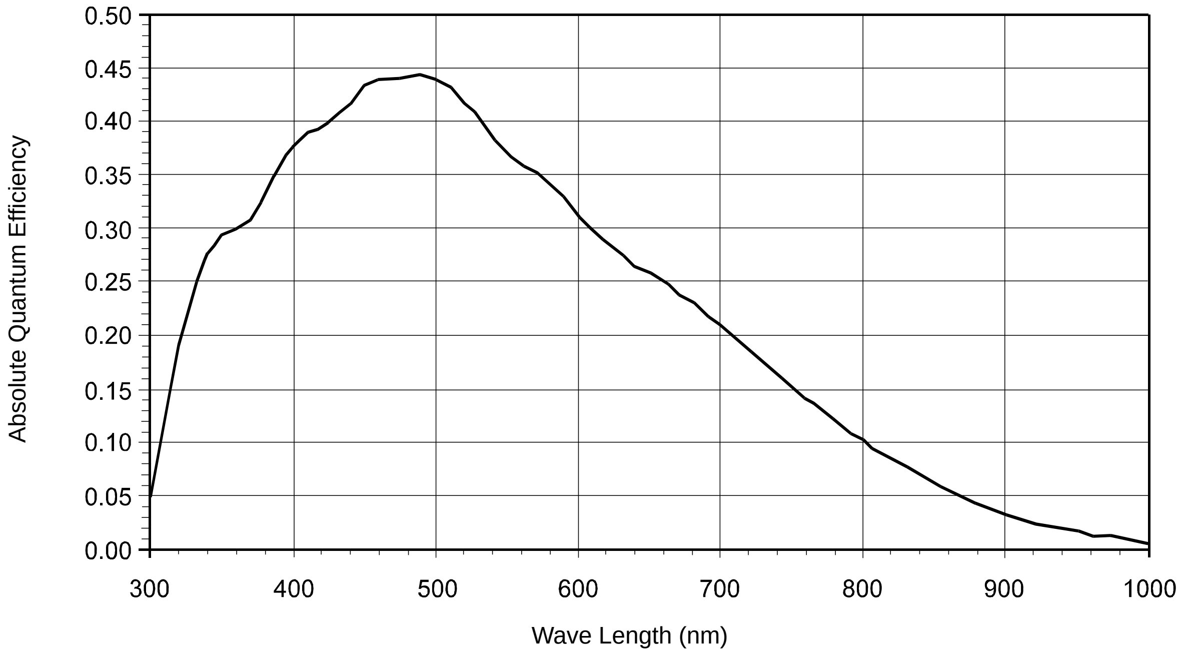


Fig. 2: piA1000-48gm Spectral Response

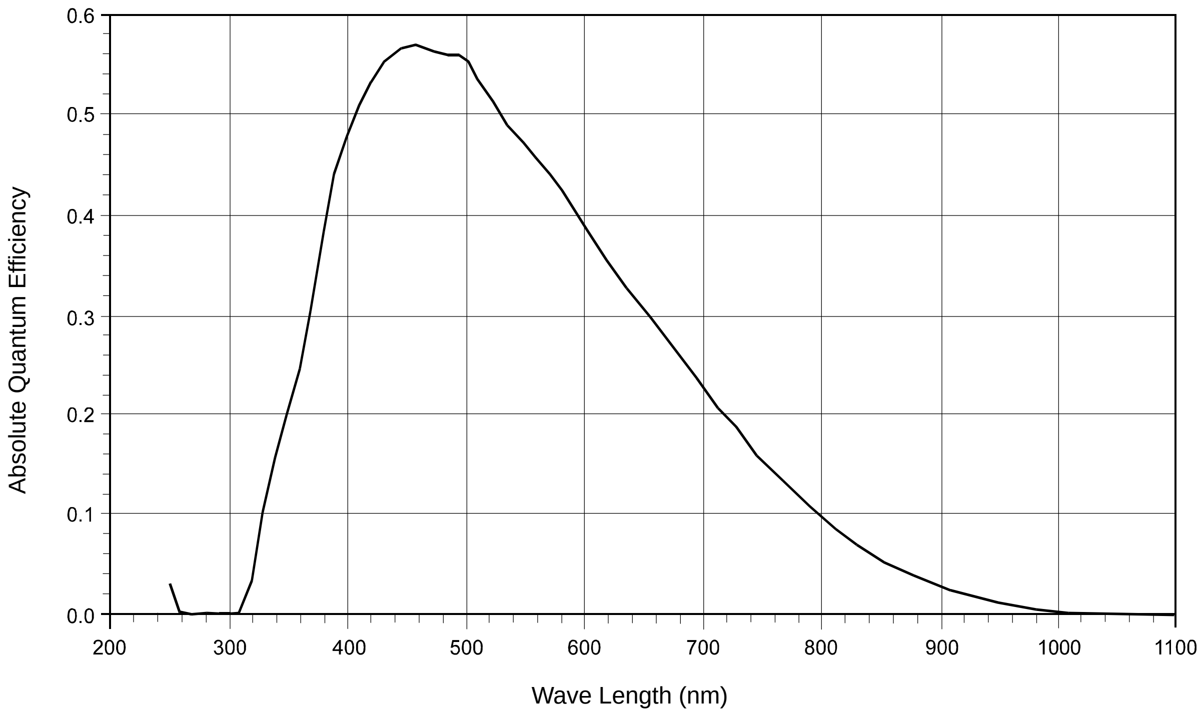


Fig. 3: piA1600-35gm Spectral Response

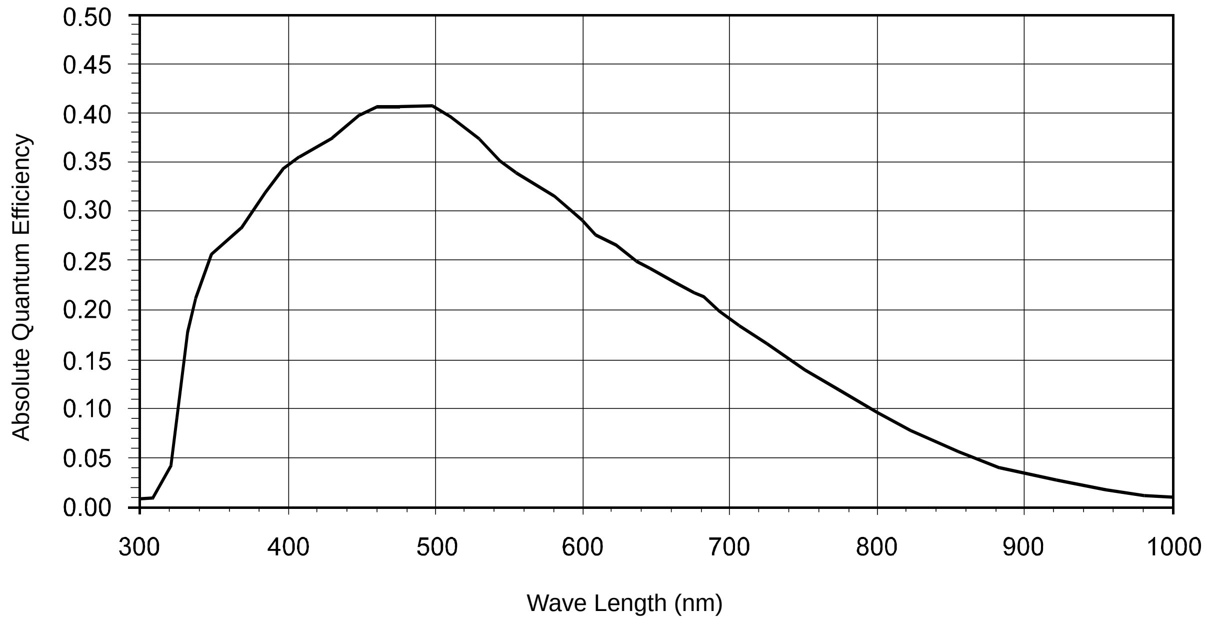


Fig. 4: piA1900-32gm Spectral Response

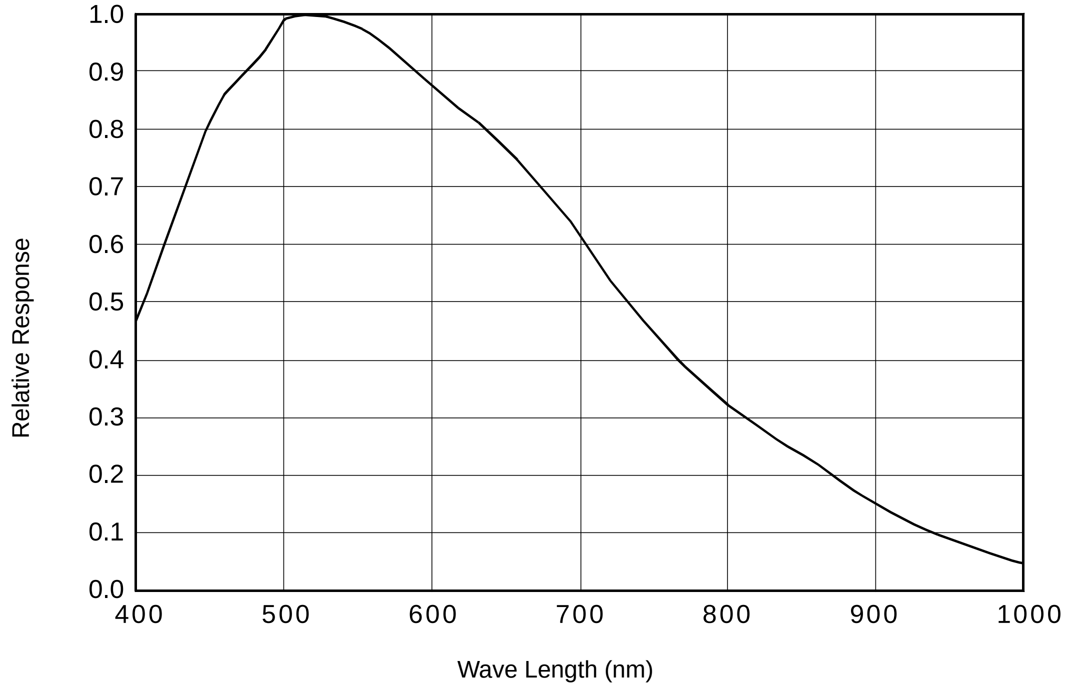


Fig. 5: piA2400-12gm and piA2400-17gm Spectral Response

1.4 Spectral Response for Color Cameras

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

Note

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

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

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

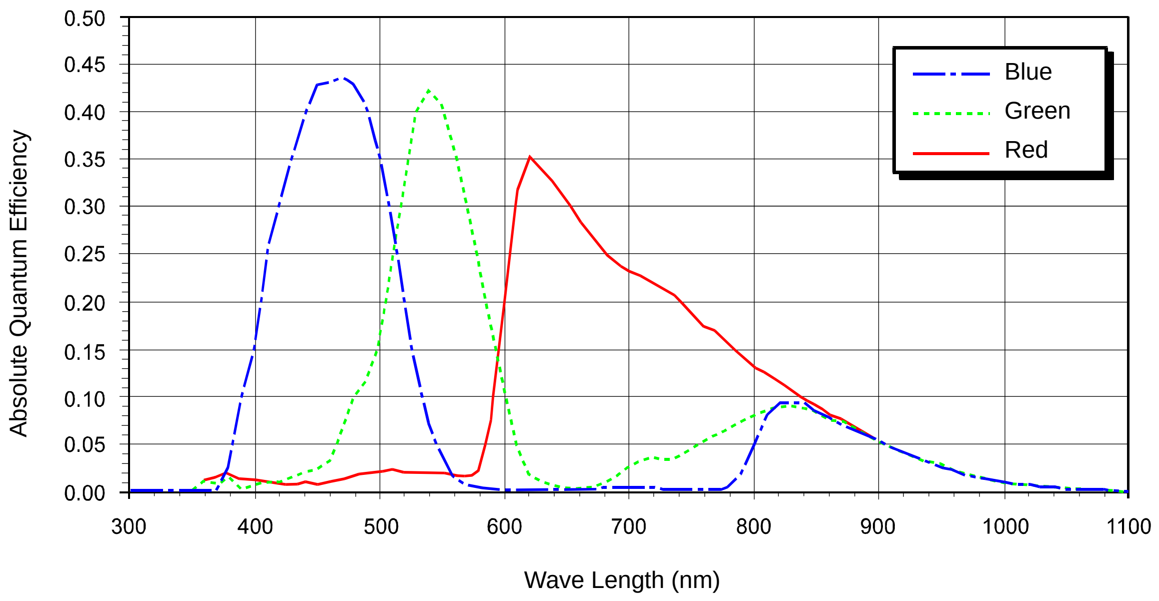


Fig. 6: piA640-210gc Spectral Response

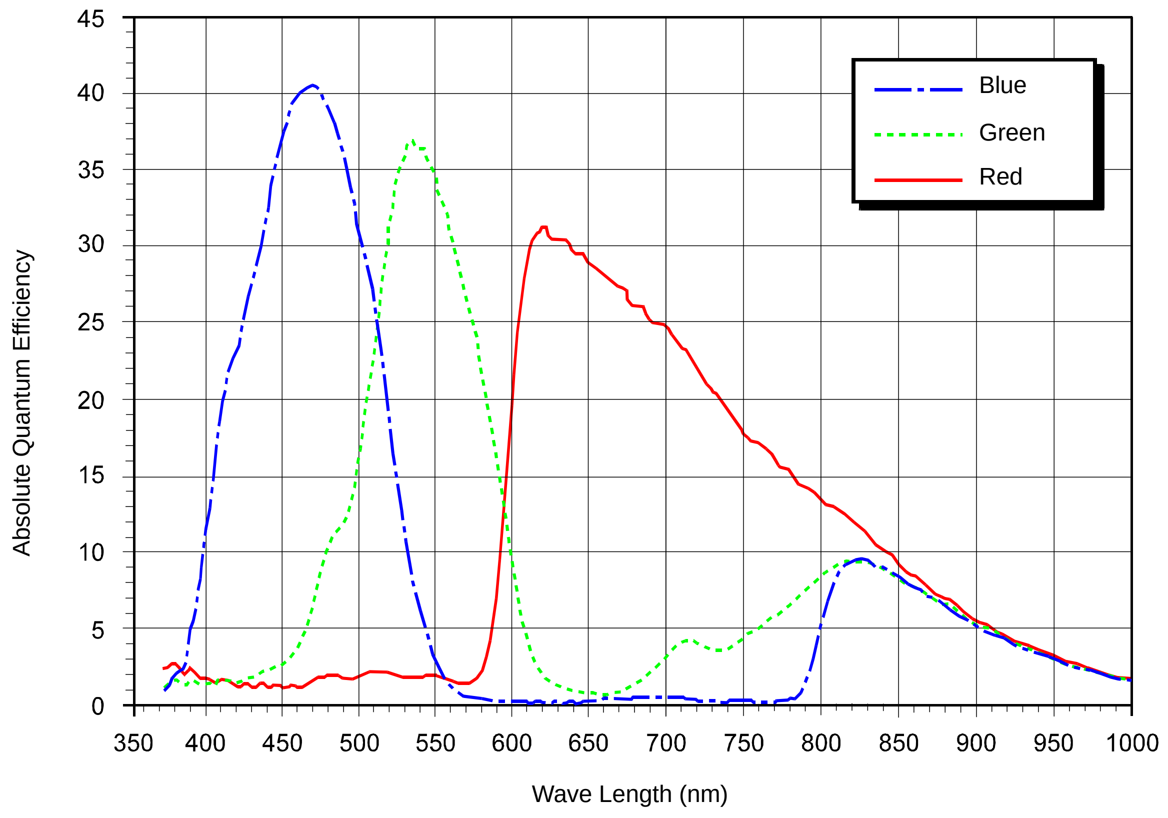


Fig. 7: piA1000-48gc Spectral Response

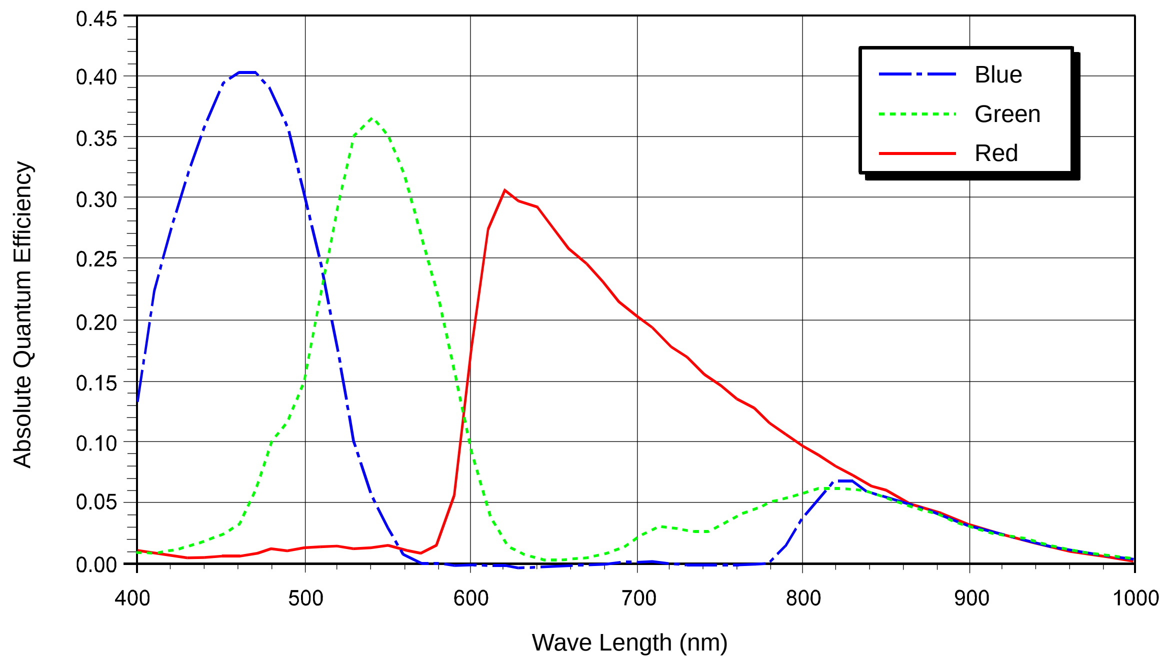


Fig. 8: piA1600-35gc Spectral Response

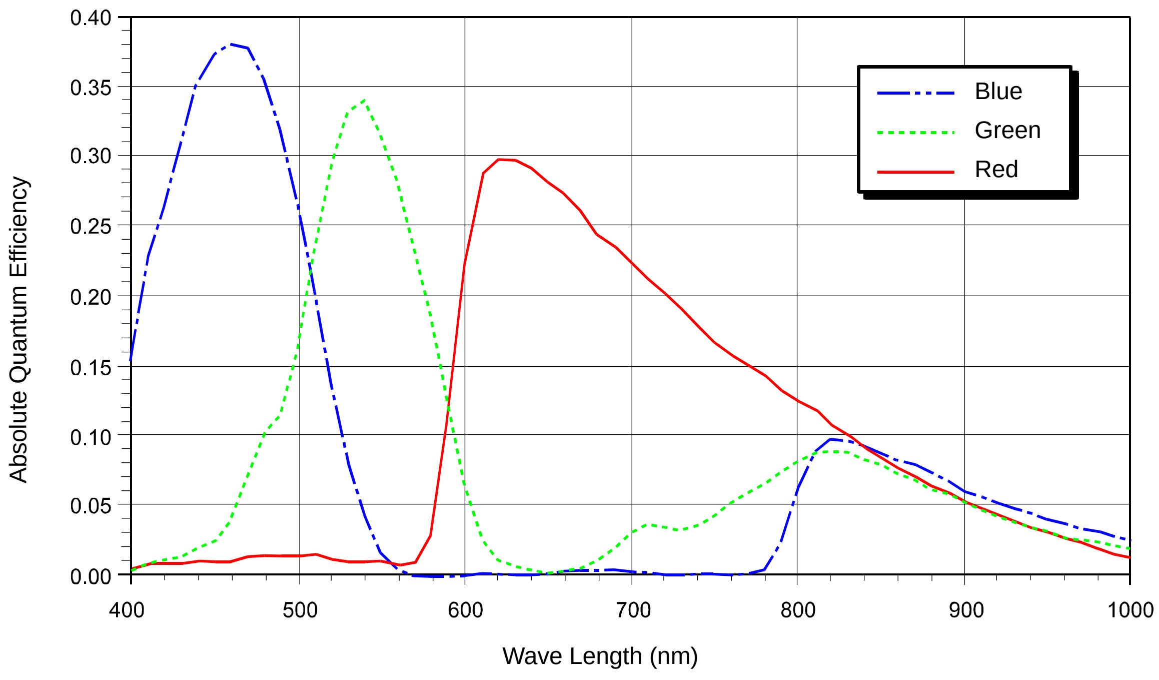


Fig. 9: piA1900-32gc Spectral Response

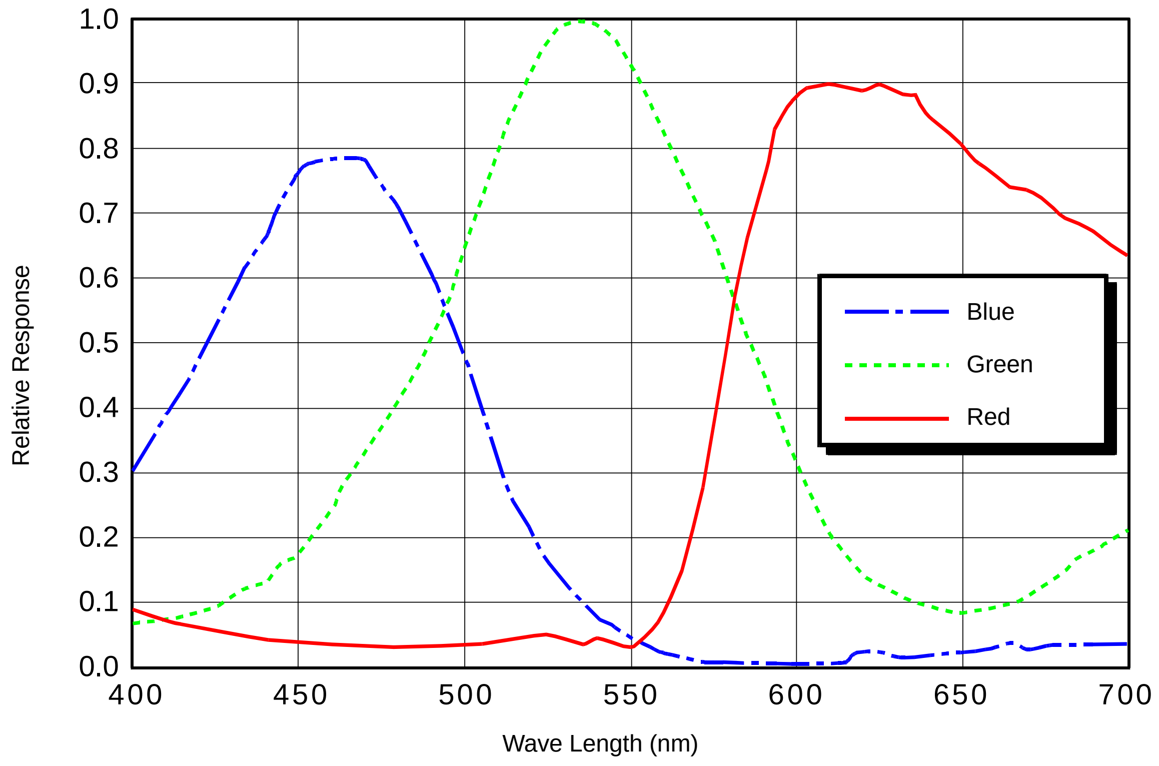


Fig. 10: piA2400-12gc and piA2400-17gc Spectral Response

1.5 Mechanical Specifications

1.5.1 Standard Housing

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

1.5.1.1 Camera Dimensions and Mounting Points

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

The camera’s dimensions in millimeters are as shown in the drawings below.

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

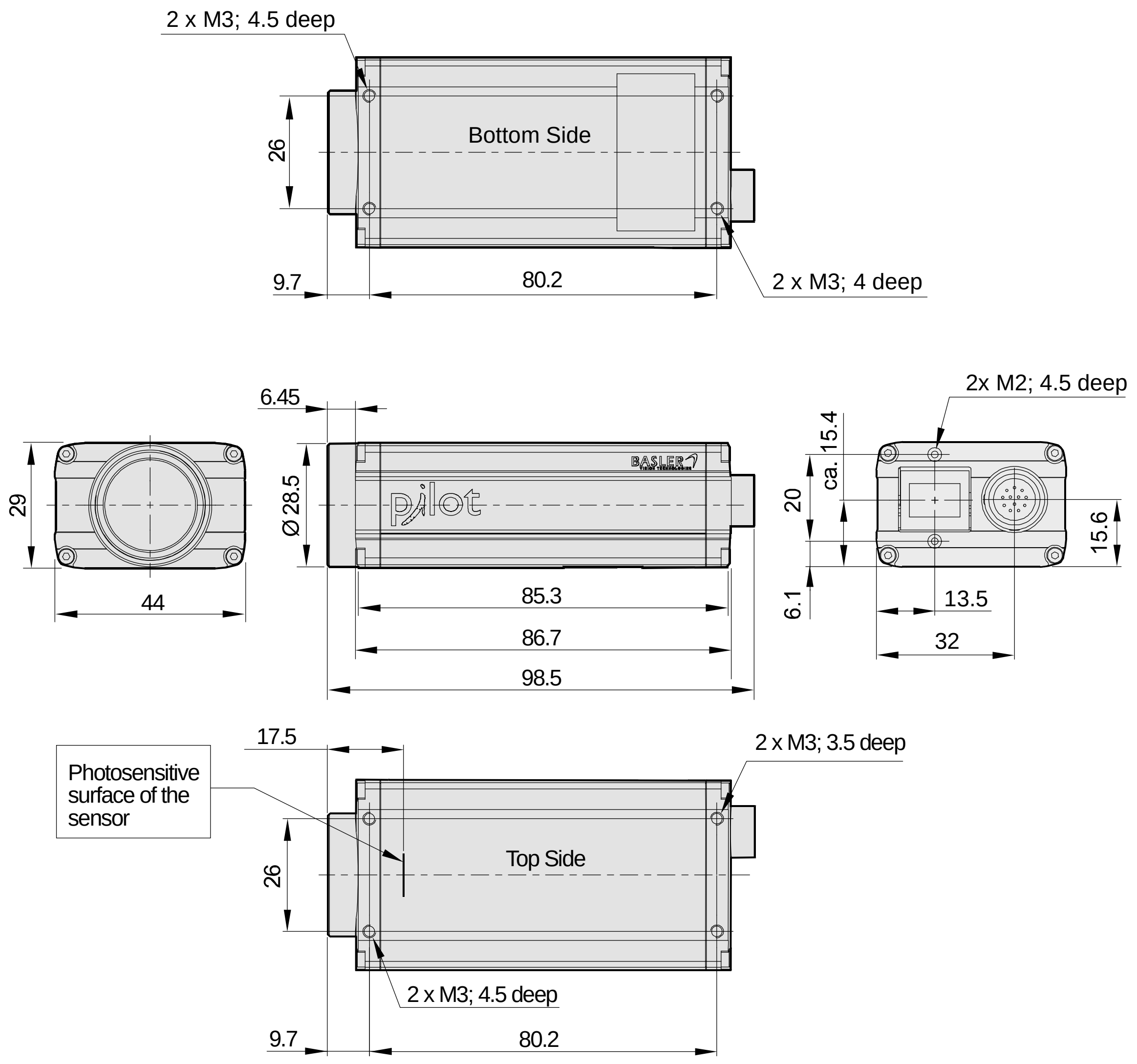
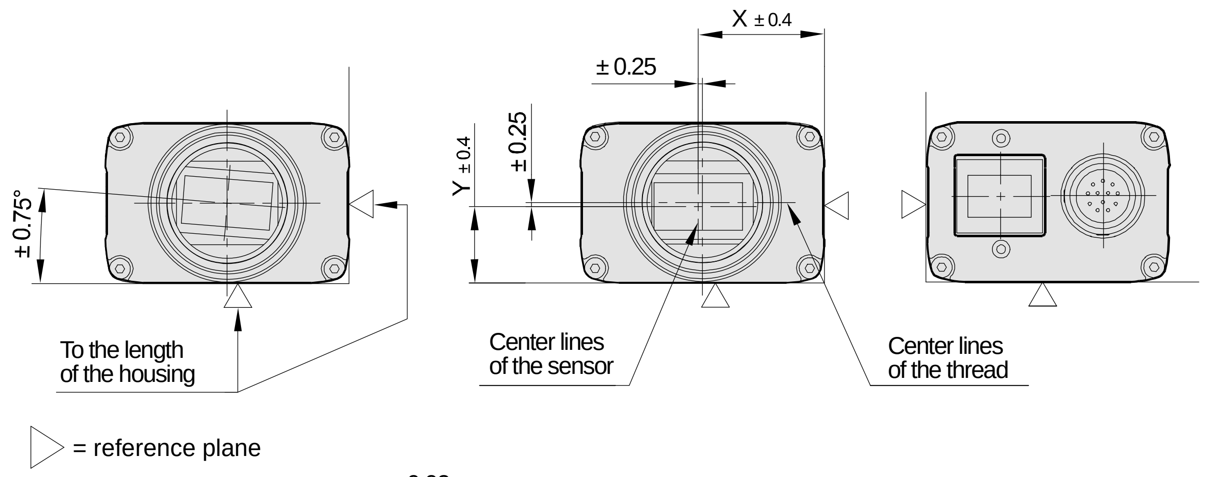
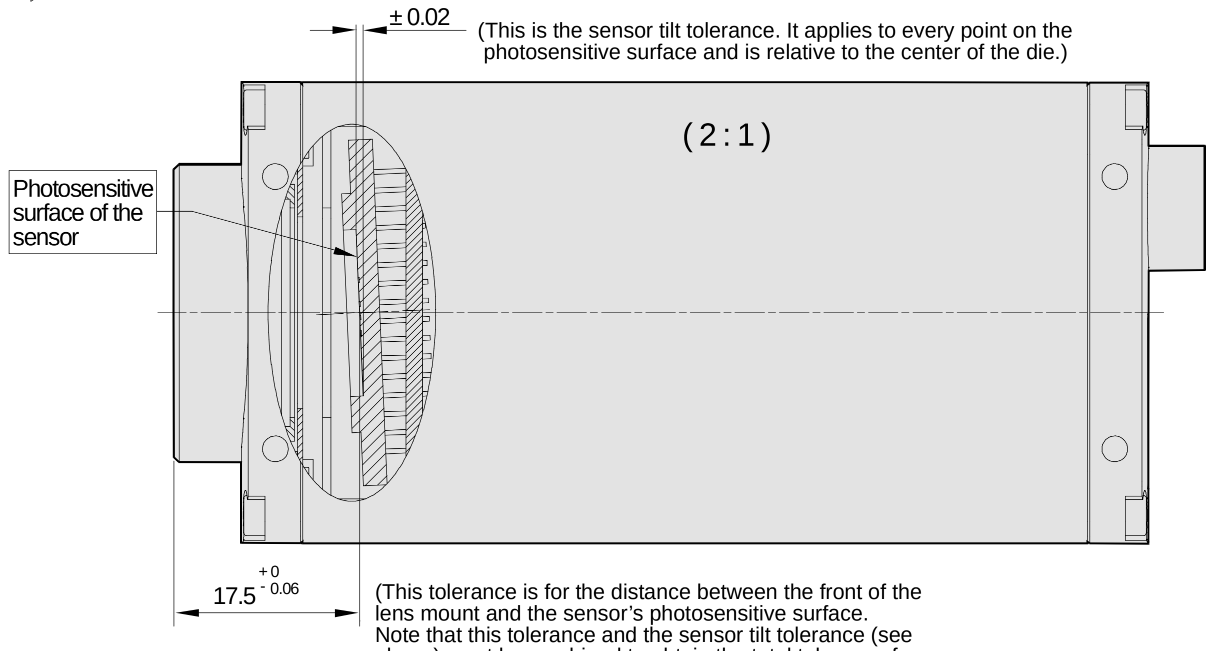


Fig. 11: Mechanical Dimensions (in mm)

1.5.1.2 Sensor Positioning Accuracy

The sensor positioning accuracy is as shown in the drawings below.


X ± 0.4
± 0.25
Y ± 0.4
± 0.25
± 0.75°
To the length of the housing
Center lines of the sensor
Center lines of the thread
= reference plane



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

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

Maximum Sensor Tilt Angle (Degrees)					
Camera	**Tilt X**	**Tilt Y**	**Camera**	**Tilt X**	**Tilt Y**
piA640-210gm/gc	0.48	0.63	**piA1900-32gm/gc**	0.16	0.29
piA1000-48gm/gc	0.31	0.31	**piA2400-12gm/gc**	0.27	0.32
piA1600-35gm/gc	0.19	0.26	**piA2400-17gm/gc**	0.27	0.32

Fig. 12: Sensor Positioning Accuracy (in mm Unless Otherwise Noted)

1.5.2 90° Head Housing

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

1.5.2.1 Camera Dimensions and Mounting Points

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

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

The dimensions in millimeters for cameras with 90° head housing are as shown in Figure 13.

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

Note

For optimum accuracy in the positioning of the camera's optical axis, we recommend using the front module reference plane (see the figure in the Sensor Positioning Accuracy section) as mounting surface.

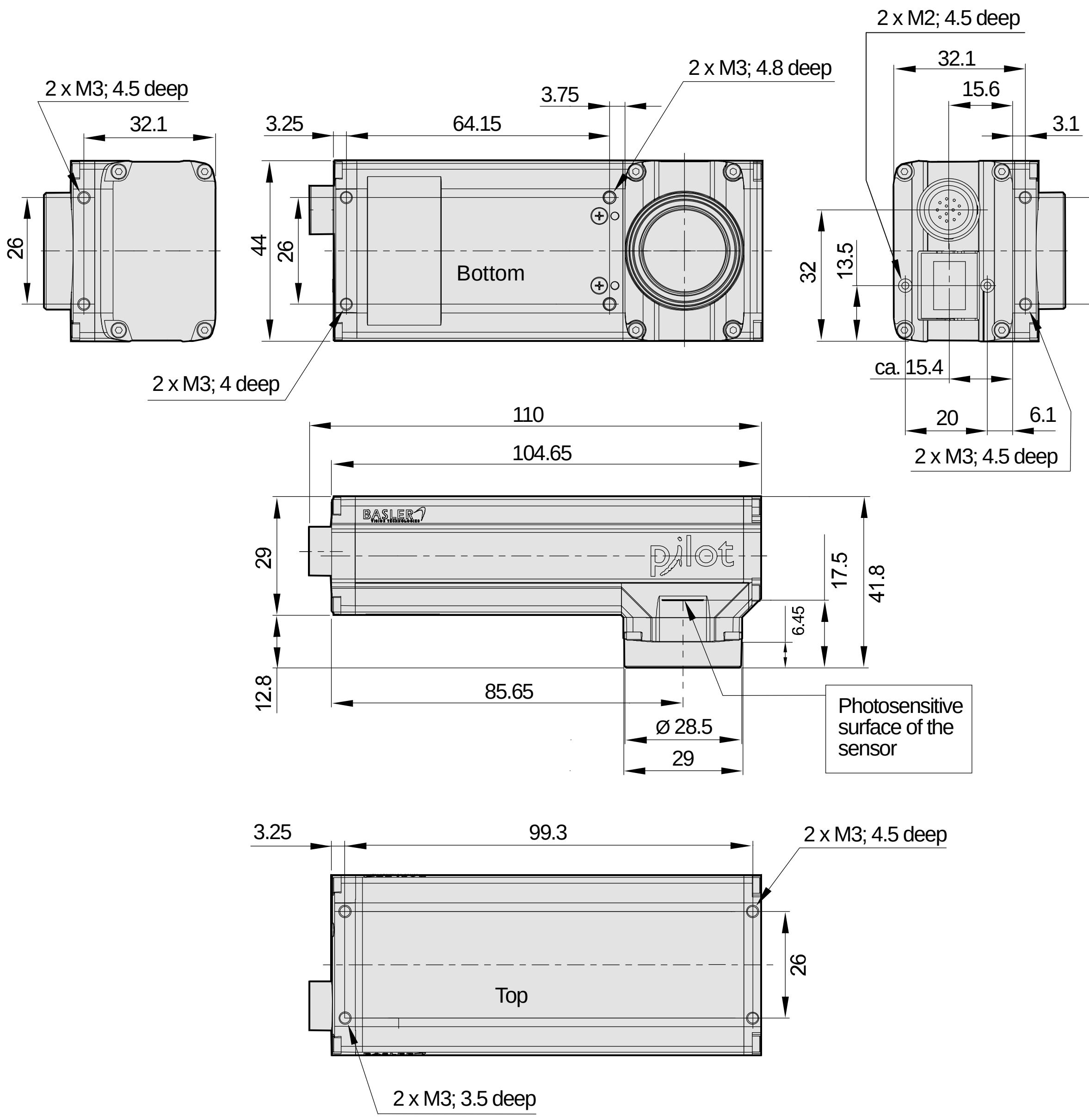
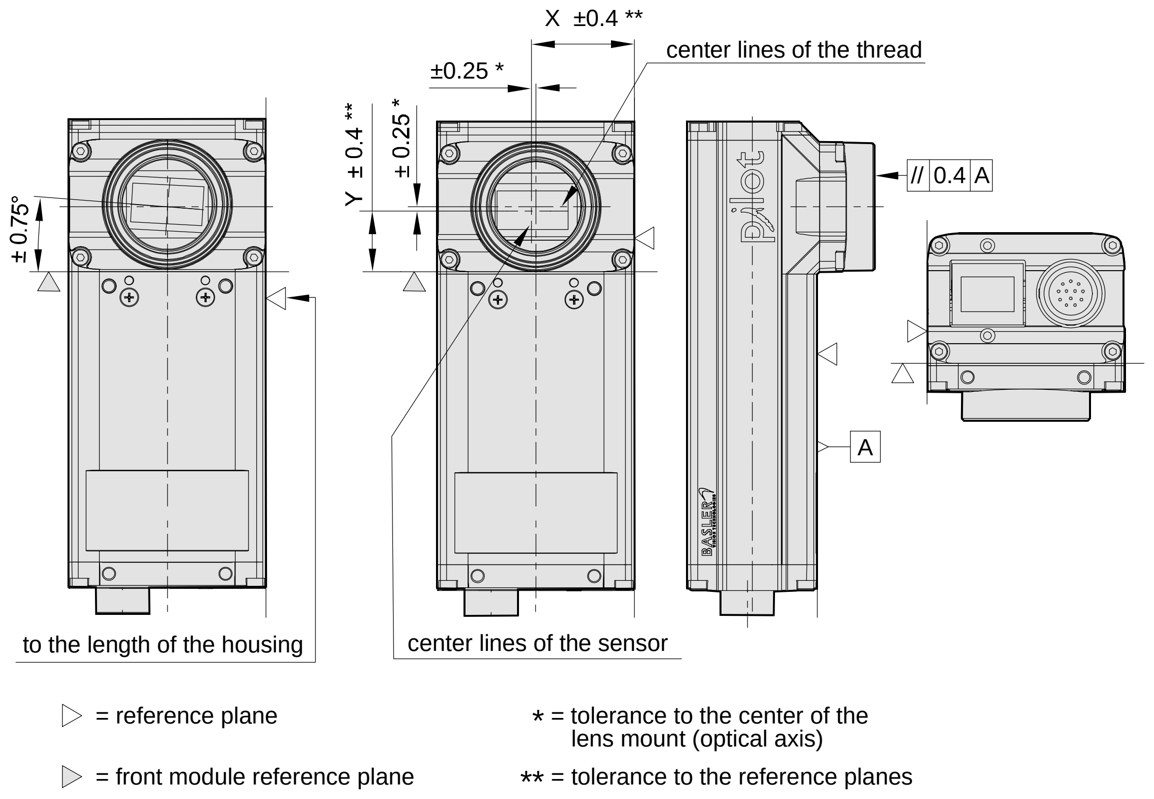
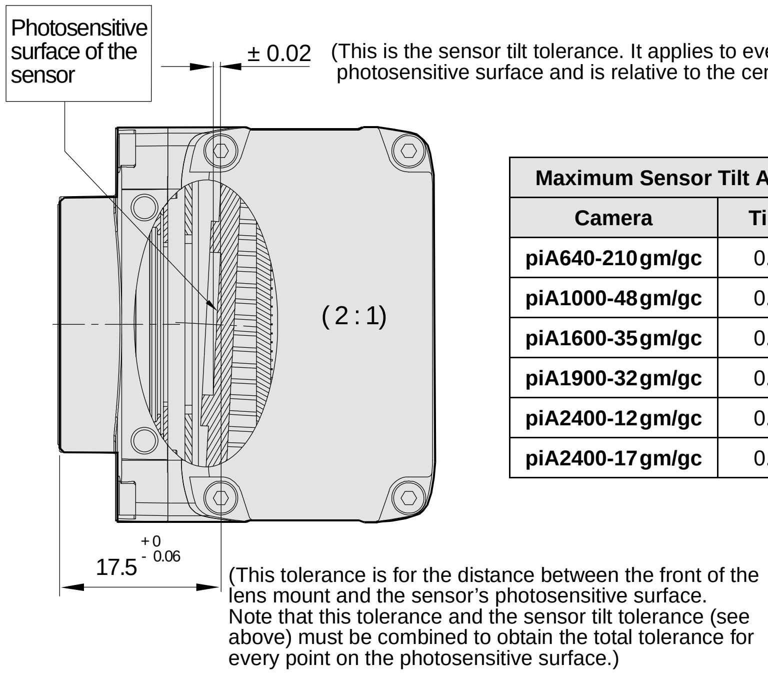


Fig. 13: Mechanical Dimensions (in mm) for Cameras With 90° Head Housing

1.5.2.2 Sensor Positioning Accuracy

The sensor positioning accuracy for cameras equipped with 90° head housing is as shown in Figure 14.


X ±0.4 **
±0.25 *
center lines of the thread
Y ± 0.4 **
± 0.25 *
± 0.75°
// 0.4 A
A
to the length of the housing
center lines of the sensor
▷ = reference plane
▷ = front module reference plane
* = tolerance to the center of the lens mount (optical axis)
** = tolerance to the reference planes


Photosensitive surface of the sensor
± 0.02
(This is the sensor tilt tolerance. It applies to every point on the photosensitive surface and is relative to the center of the die)
(2 : 1)
17.5 +0 -0.06
(This tolerance is for the distance between the front of the lens mount and the sensor's photosensitive surface. Note that this tolerance and the sensor tilt tolerance (see above) must be combined to obtain the total tolerance for every point on the photosensitive surface.)

Maximum Sensor Tilt Angle (Degrees)		
Camera	**Tilt X**	**Tilt Y**
piA640-210gm/gc	0.48	0.63
piA1000-48gm/gc	0.31	0.31
piA1600-35gm/gc	0.19	0.26
piA1900-32gm/gc	0.16	0.29
piA2400-12gm/gc	0.27	0.32
piA2400-17gm/gc	0.27	0.32

Fig. 14: Sensor Positioning Accuracy for Cameras With 90° Head Housing (in mm unless otherwise noted)

1.5.3 Maximum Thread Length on Color Cameras

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

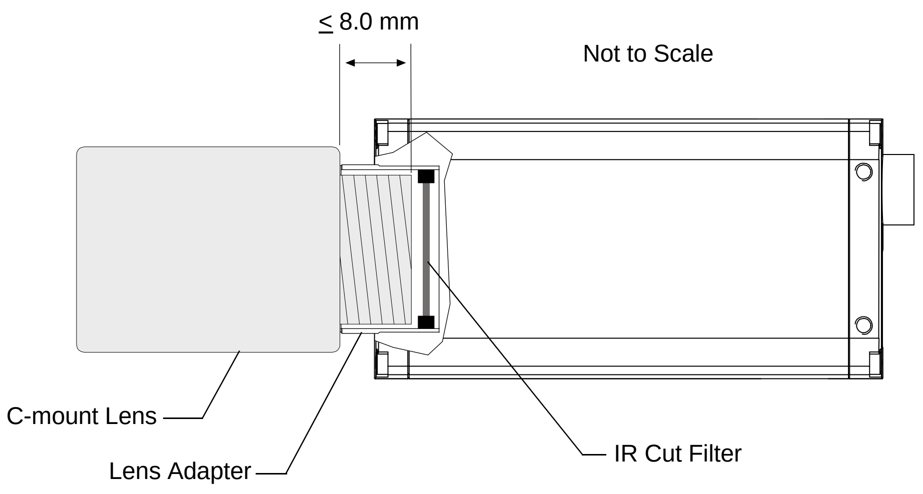


Fig. 15: Maximum Lens Thread Length on Color Cameras

Note

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

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

1.5.4 Mechanical Stress Test Results

Pilot cameras were submitted to an independent mechanical testing laboratory and subjected to the stress tests listed below. The mechanical stress tests were performed on selected camera models with standard housing. 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 4: Mechanical Stress Tests

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

1.6 Software Licensing Information

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

1.7 Avoiding EMI and ESD Problems

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

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

- Always use high quality shielded cables. The use of high quality cables is one of the best defenses against EMI and ESD.
- Try to use camera cables that are the correct length and try to run the camera cables and power cables parallel to each other. Avoid coiling camera cables. If the cables are too long, use a meandering path rather then coiling the cables.
- Avoid placing camera cables parallel to wires carrying high-current, switching voltages such as wires supplying stepper motors or electrical devices that employ switching technology. **Placing camera cables near to these types of devices may cause problems with the camera.**
- Attempt to connect all grounds to a single point, e.g., use a single power outlet for the entire system and connect all grounds to the single outlet. This will help to avoid large ground loops. (Large ground loops can be a primary cause of EMI problems.)
- Use a line filter on the main power supply.
- Install the camera and camera cables as far as possible from devices generating sparks. If necessary, use additional shielding.
- Decrease the risk of electrostatic discharge by taking the following measures:
  - Use conductive materials at the point of installation (e.g., floor, workplace).
  - Use suitable clothing (cotton) and shoes.
  - Control the humidity in your environment. Low humidity can cause ESD problems.

Note

The Basler application note called *Avoiding EMI and ESD in Basler Camera Installations* provides much more detail about avoiding EMI and ESD.
The application note can be downloaded at:
www.baslerweb.com/indizes/download_index_en_31412.html

1.8 Environmental Requirements

1.8.1 Temperature and Humidity

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

1.8.2 Heat Dissipation

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

- In all cases, you should monitor the temperature of the camera housing and make sure that the temperature does not exceed 50 °C. Keep in mind that the camera will gradually become warmer during the first 1.5 hours of operation. After 1.5 hours, the housing temperature should stabilize and no longer increase.
- If your camera is mounted on a substantial metal component in your system, this may provide sufficient heat dissipation.
- The use of a fan to provide air flow over the camera is an extremely efficient method of heat dissipation. The use of a fan provides the best heat dissipation.

1.9 Precautions

CAUTION

Avoid Dust on the Sensor

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

CAUTION

Lens Thread Length is Limited

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

For more specific information about the lens thread length, see Section 1.5.3 on page 20.

CAUTION

Voltage Outside of Specified Range Can Cause Damage

If the voltage of the power to the camera is greater than +30.0 VDC damage to the camera can result. If the voltage is less than +11.3 VDC, the camera may operate erratically.

CAUTION

An Incorrect Plug Can Damage the 12-pin Connector

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

CAUTION

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.

Warranty Precautions

To ensure that your warranty remains in force:

Do not remove the camera's serial number label

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

Do not open the camera housing

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

Keep foreign matter outside of the camera

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

Avoid Electromagnetic fields

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

Transport Properly

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

Clean Properly

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

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

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

Read the manual

Read the manual carefully before using the camera!

2 Software and Hardware Installation

The information you will need to install and operate the camera is included in the Installation and Setup Guide for Cameras Used with Basler’s pylon API (AW000611xx000).

You can download the Installation and Setup Guide for Cameras Used with Basler’s pylon API from the Basler website: www.baslerweb.com/indizes/download_index_en_19627.html.

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

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

3 Tools for Changing Camera Parameters

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

3.1 The pylon Viewer

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

The pylon Viewer is included in Basler’s pylon Driver Package. You can download the pylon package from the Basler website: www.baslerweb.com/beitraege/beitrag_en_71708.html.

For more information about using the viewer, see the installation and Setup Guide for Cameras Used with Basler’s pylon API, (AW000611xx000). You can download the guide from the Basler website: www.baslerweb.com/indizes/download_index_en_19627.html.

3.2 The IP Configuration Tool

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

The IP Configuration Tool is included in Basler’s pylon Driver Package. You can download the pylon package from the Basler website: www.baslerweb.com/beitraege/beitrag_en_71708.html.

For more information about using IP Configuration Tool, see the installation and Setup Guide for Cameras Used with Basler’s pylon API, (AW000611xx000). You can download the guide from the Basler website: www.baslerweb.com/indizes/download_index_en_19627.html.

3.3 The pylon API

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

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

The SDK is included in Basler’s pylon Driver Package. You can download the pylon package from the Basler website: www.baslerweb.com/beitraege/beitrag_en_71708.html.

For more information about installing pylon software, see the installation and Setup Guide for Cameras Used with Basler’s pylon API (AW000611xx000). You can download the guide from the Basler website: www.baslerweb.com/indizes/download_index_en_19627.html.

4 Basler Network Drivers and Parameters

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

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

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

Note

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

For more information about compatible Intel chipsets, see the Installation and Setup Guide for Cameras Used with Basler’s pylon API, (AW000611xx000).

For more information about installing the network drivers, see the Installation and Setup Guide for Cameras Used with Basler’s pylon API, (AW000611xx000).

4.1 The Basler Filter Driver

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

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

The parameters associated with the filter driver are described below.

Enable Resend - Enables or disables the packet resend mechanism.

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

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

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

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

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

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

// Packet Timeout/FrameRetention
Camera_t::StreamGrabber_t StreamGrabber ( Camera.GetStreamGrabber(0) );
StreamGrabber.PacketTimeout.SetValue( 40 );
StreamGrabber.FrameRetention.SetValue( 200 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference (AW000131xx000).

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 on page 29.

4.2 The Basler Performance Driver

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

For more information about compatible Intel chipsets, see the Installation and Setup Guide for Cameras Used with Basler's pylon API, (AW000611xx000).

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

- The threshold resend mechanism
- The timeout resend mechanism

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

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

Note

The parameter default values will provide for the following:

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

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

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

General Parameters

Enable Resend - Enables the packet resend mechanisms.

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

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

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

Threshold Resend Mechanism Parameters

The threshold resend request mechanism is illustrated in Figure 16 where the following assumptions are made:

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

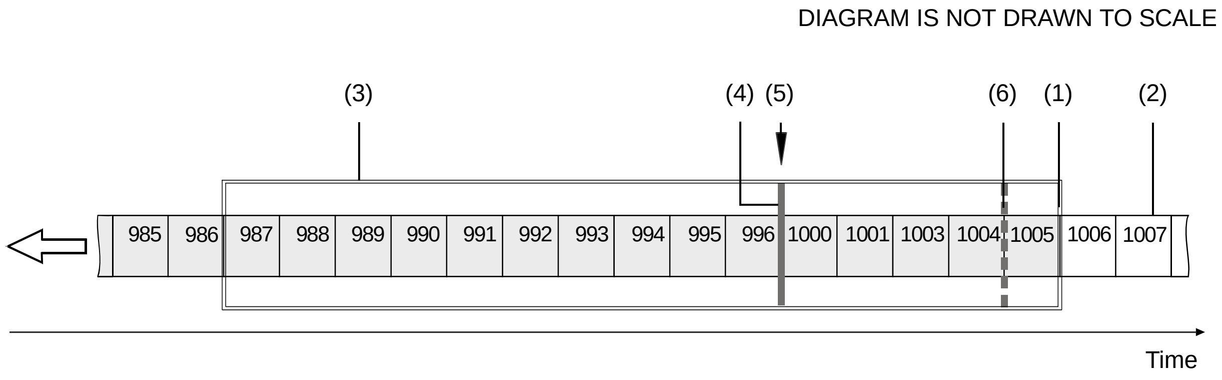


Fig. 16: Example of a Receive Window with Resend Request Threshold & Resend Request Batching Threshold

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

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

(3) Receive window of the performance driver.

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

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

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

Resend Request Threshold - This parameter determines the location of the resend request threshold within the receive window as shown in Figure 16. The parameter value is in per cent of the width of the receive window. In Figure 16 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 16). 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 16, packets 987 to 1005 are within the receive window and packets 997 to 999 and 1002 were detected as missing. In the situation shown, a resend request is sent to the camera for each of the missing consecutive packets 997 to 999. The resend requests are sent after packet 996 - the last packet of the intact sequence of packets - has advanced beyond the resend request threshold and before packet 1000 - the next packet in the stream of packets - can advance beyond the resend request threshold. Similarly, a resend request will be sent for missing packet 1002 after packet 1001 has advanced beyond the resend request threshold and before packet 1003 can advance beyond the resend request threshold.

Resend Request Batching - This parameter determines the location of the resend request batching threshold in the receive window (Figure 16). 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 16 the resend request batching threshold is set at 80% of the span.

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

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

If the Resend Request Batching parameter is set to 0, no batching will occur and a resend request will be sent for each single missing packet. For other settings, consider an example: Suppose the Resend Request Batching parameter is set to 80 referring to a span between the resend request threshold and the front end of the receive window that can hold five packets (Figure 16). In this case 4 packets (5 x 80%) will fit into the span between the resend request threshold and the resend request batching threshold. Accordingly, the maximum number of consecutive missing packets that can be batched is 5 (4 + 1).

Timeout Resend Mechanism Parameters

The timeout resend mechanism is illustrated in Figure 17 where the following assumptions are made:

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

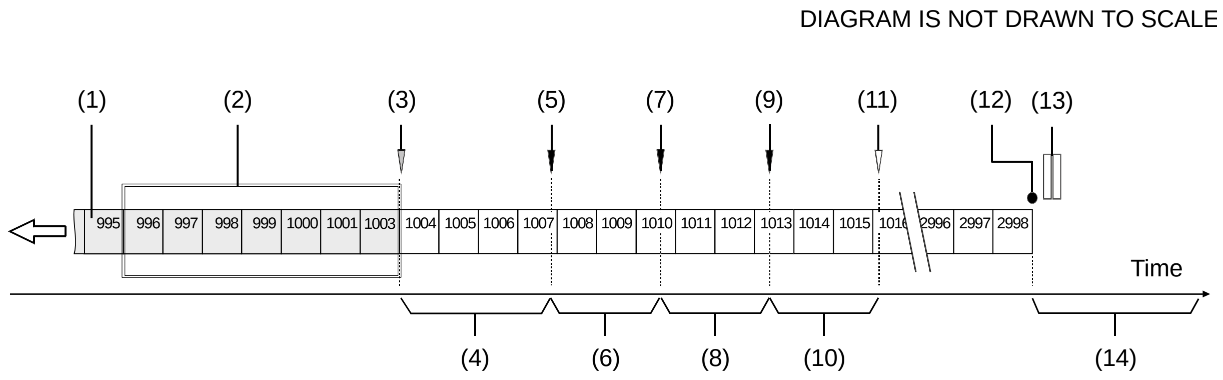


Fig. 17: Incomplete Stream of Packets and Part of the Resend Mechanism

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

(2) Receive window of the performance driver.

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

(4) Interval defined by the Resend Timeout parameter.

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

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

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

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

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

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

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

(12) End of the frame.

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

(14) Interval defined by the Packet Timeout parameter.

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

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

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

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

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

Threshold and Timeout Resend Mechanisms Combined

Figure 18 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 18). In addition, resend request batching will not occur.

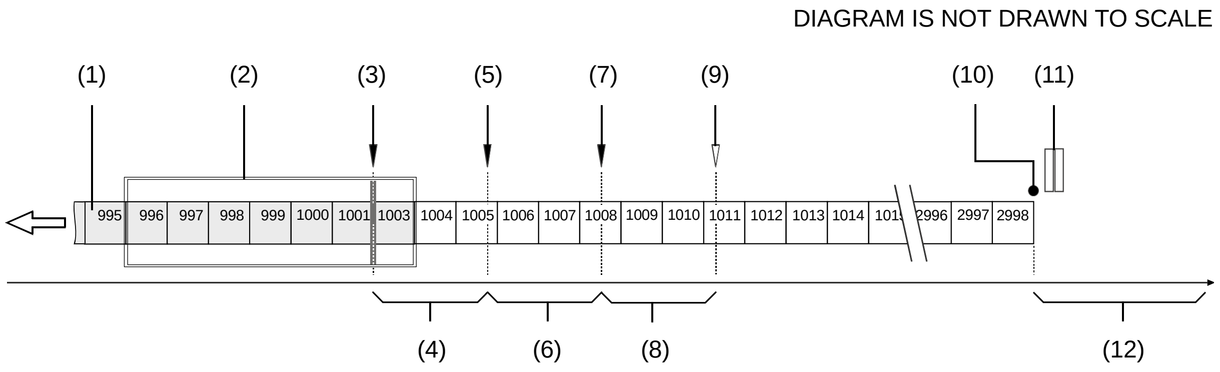


Fig. 18: Combination of Threshold Resend Mechanism and Timeout Resend Mechanism

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

(2) Receive window of the performance driver.

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

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

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

(6) Interval defined by the Resend Response Timeout parameter

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

(8) Interval defined by the Resend Response Timeout parameter

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

(10) End of the frame.

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

(12) Interval defined by the Packet Timeout parameter.

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

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

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

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

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

Adapter Properties

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

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

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

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

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

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

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

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

To access the advanced properties for an adapter:

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

Note

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

4.3 Transport Layer Parameters

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

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

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

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

Note

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

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

```
 // Read/Write Timeout
Camera_t::TlParams_t TlParams( Camera.GetTLNodeMap() );
TlParams.ReadTimeout.SetValue(500);  // 500 milliseconds
TlParams.WriteTimeout.SetValue(500); // 500 milliseconds

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

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

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

5 Network Related Camera Parameters and Managing Bandwidth

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

5.1 Network Related Parameters in the Camera

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

Payload Size (read only)

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

Stream Channel Selector (read/write)

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

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

Packet Size (read/write)

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

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

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

Inter-packet Delay (read/write)

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

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

Frame Transmission Delay (read/write)

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

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

Bandwidth Assigned (read only)

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

In essence, the bandwidth assigned is calculated this way:

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

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

Y = number of bytes in each packet

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

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

Bandwidth Reserve (read/write)

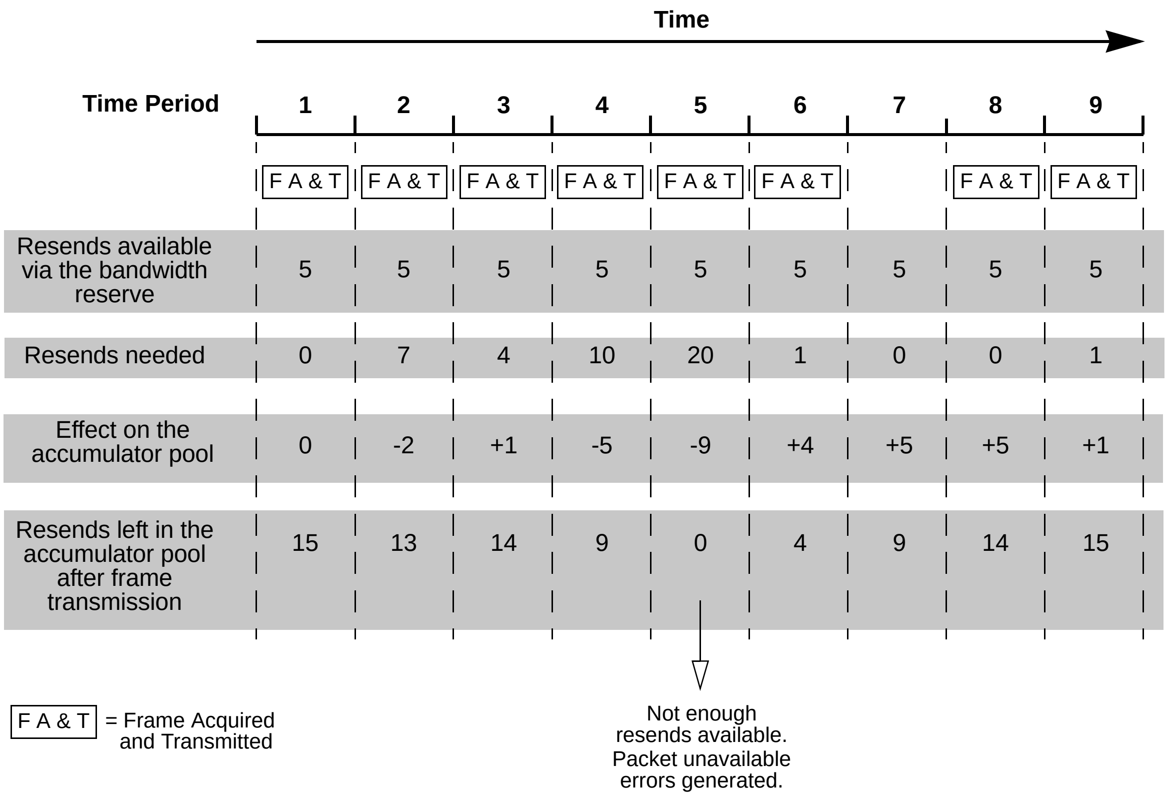
Used to reserve a portion of the assigned bandwidth for packet resends and for the transmission of control data between the camera and the host PC. The setting is expressed as a percentage of the Bandwidth Assigned parameter. For example, if the Bandwidth Assigned parameter indicates that 30 MByte/s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

Bandwidth Reserve Accumulation (read/write)

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

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

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


Time
F A & T = Frame Acquired and Transmitted
Not enough resends available. Packet unavailable errors generated.

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

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

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

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

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

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

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

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

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

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

Frame Max Jitter (read only)

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

Device Max Throughput (read only)

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

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

Device Current Throughput (read only)

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

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

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

Resulting Frame Rate (read only)

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

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

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

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

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

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

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

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

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

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


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

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

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

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

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

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 on page 29.

5.2 Managing Bandwidth When Multiple Cameras Share a Single Network Path

If you are using a single camera on a GigE network, the problem of managing bandwidth is simple. The network can easily handle the bandwidth needs of a single camera and no intervention is required. A more complicated situation arises if you have multiple cameras connected to a single network adapter as shown in Figure 19.

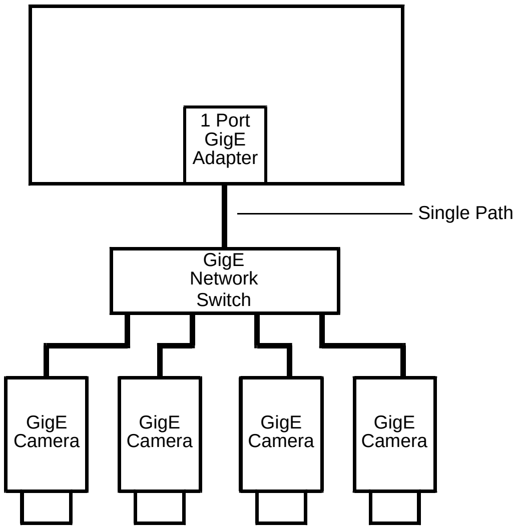


Fig. 19: Multiple Cameras on a Network

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

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

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

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

5.2.1 A Procedure for Managing Bandwidth

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

The objectives of the procedure are:

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

Step 1 - Improve the Network Performance.

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

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

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

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

  If possible, set the number of receive descriptors to a maximum value and set the number of CPU interrupts to a low value.

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

Contact Basler technical support if you need further assistance.

Step 2 - Set the Packet Size parameter on each camera as large as possible.

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

Unless you have already set the packet size for your network adapter during the installation of the Basler pylon software, check the documentation for your adapter to determine the maximum packet size (sometimes called “frame” size) that the adapter can handle. Many adapters can handle what is known as “jumbo packets” or "jumbo frames". These are packets with a maximum size of 16 kB. Once you have determined the maximum size packets the adapter can handle, make sure that the adapter is set to use the maximum packet size.

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

Now that you have set the adapter and switch, you can determine the largest packet size the network can handle. The device with the smallest maximum packet size determines the maximum allowed packet size for the network. For example, if the adapter can handle 8 kB packets and the switch can handle 6 kB packets, then the maximum for the network is 6 kB packets.

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

Tip

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

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

Step 3 - Set the Bandwidth Reserve parameter for each camera.

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

Step 4 - Calculate the "data bandwidth needed" by each camera.

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

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

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

Data Bandwidth Needed = Bytes/Frame x Frames/s

Where:

Packet Overhead = 72 (for a GigE network)

78 (for a 100 MBit/s network)

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

Packet Overhead + 12 (if chunk mode is active)

Trailer Size = Packet Overhead + 8

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

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

Step 5 - Calculate “data bandwidth assigned” to each camera.

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

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

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

Step 6 - For each camera, compare the data bandwidth needed with the data bandwidth assigned.

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

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

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

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

Note

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

Step 7 - Check that the total bandwidth assigned is less than the network capacity.

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

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

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

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

For more information about the camera's maximum allowed frame transmission rate, see Section 8.9 on page 102.

For more information about the AOI, see Section 11.6 on page 163.

6 Camera Functional Description

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

6.1 Overview

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

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

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

Accumulated charges are read out of the sensor when exposure ends. At readout, the accumulated charges are transported from the sensor’s light-sensitive elements (pixels) to its vertical shift registers (see Figure 20 on page 58). The charges from the bottom line of pixels in the array are then moved to two horizontal shift registers as shown in the figure. Charges from the left half of the line are moved to the left horizontal shift register and charges from the right half of the line are moved to the right horizontal shift register. The left horizontal shift register shifts out charges from left to right, that is, pixel 1, pixel 2, pixel 3, and so on. The right horizontal shift register shifts out charges from right to left, that is, pixel n, pixel n-1, pixel n-2, and so on (where n is the last pixel in a line).

As the charges move out of the horizontal shift registers, 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). For optimal digitization, gain and black level can be adjusted by setting camera parameters. After each voltage has been amplified and digitized, it passes through an FPGA and into an image buffer. As the pixel data passes through the FPGA, it is reordered so that the pixel data for each line will be transmitted from the camera in ascending order from pixel 1 through pixel n. All shifting is clocked according to the camera’s internal data rate. Shifting continues in a line-by-line 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.

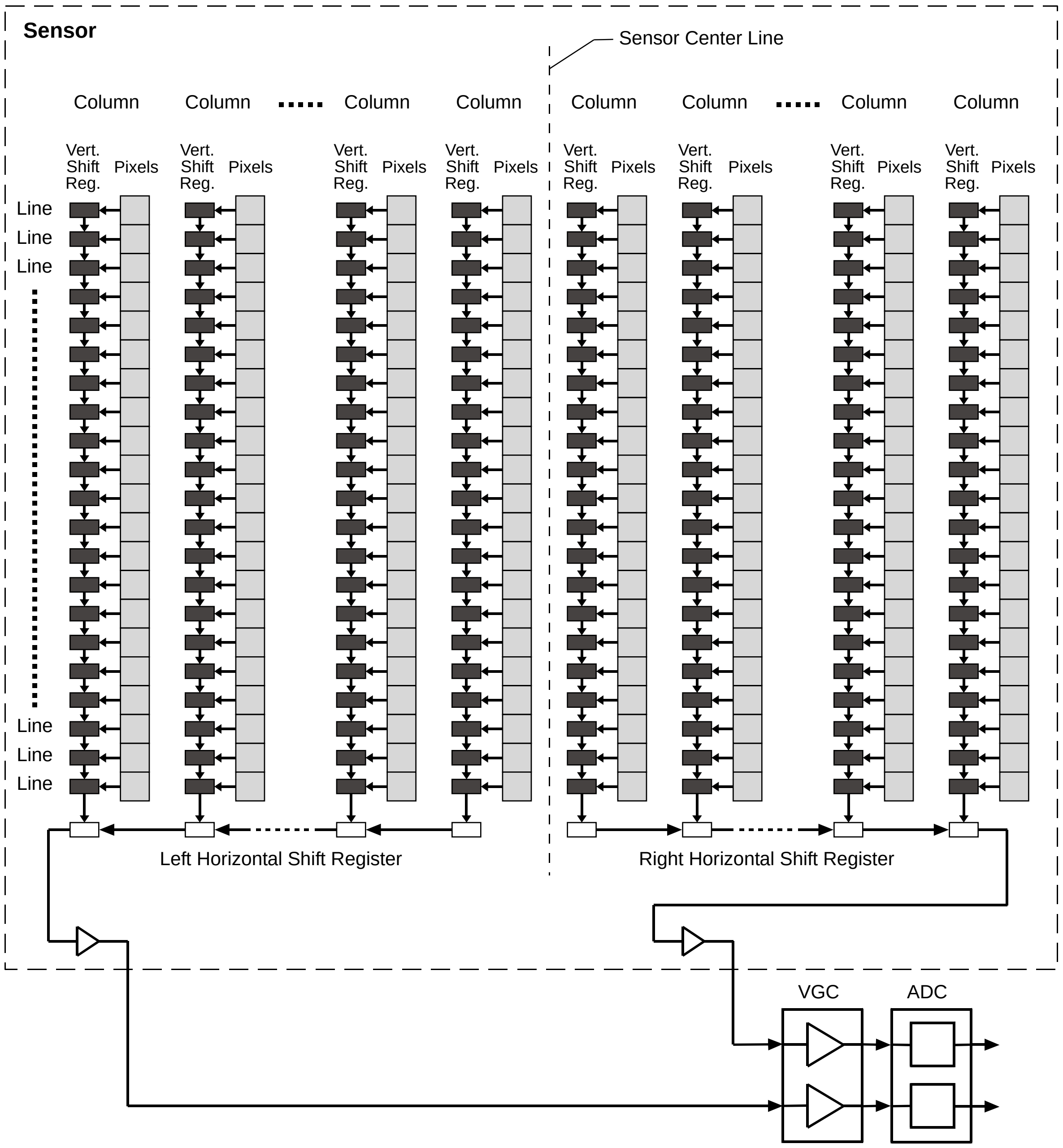


Fig. 20: CCD Sensor Architecture

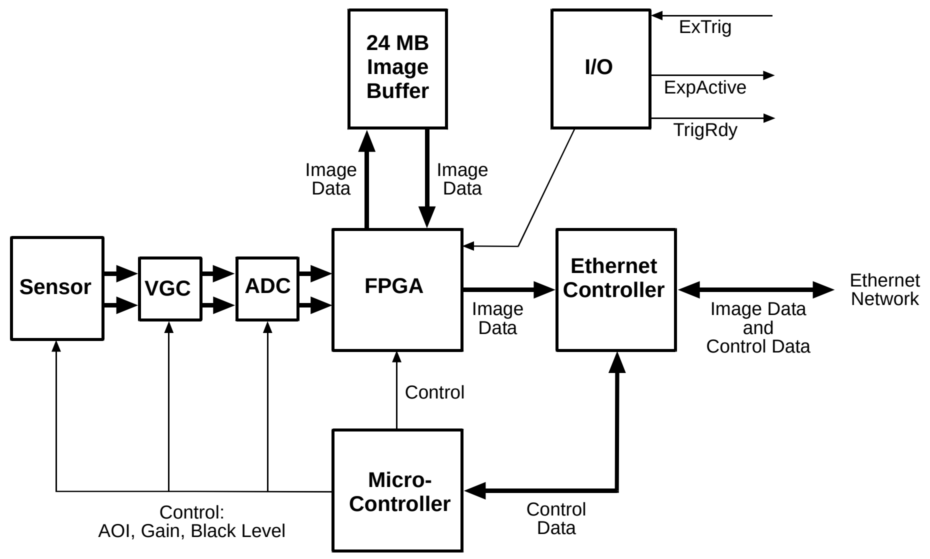


Fig. 21: Camera Block Diagram

7 Physical Interface

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

7.1 General Description of the Connections

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

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

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

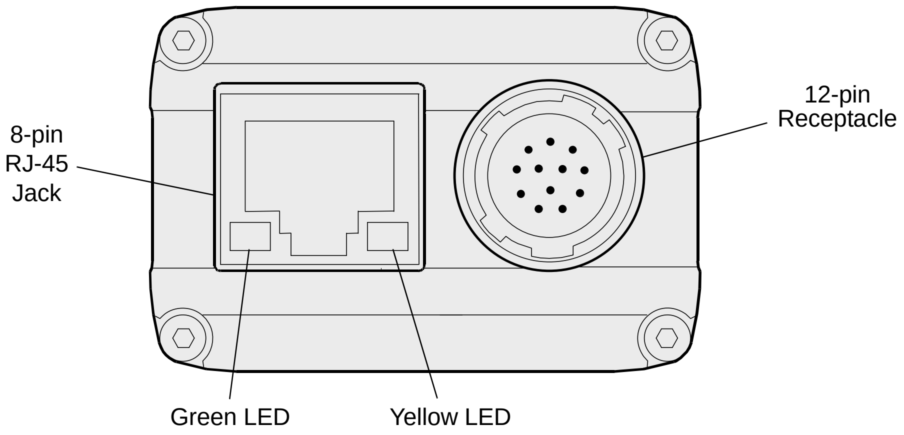


Fig. 22: Camera Connectors and LED

7.2 Connector Pin Assignments and Numbering

7.2.1 12-pin Receptacle Pin Assignments

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

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

Table 5: Pin Assignments for the 12-pin Receptacle

Note

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

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

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

7.2.2 RJ-45 Jack Pin Assignments

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

7.2.3 Pin Numbering

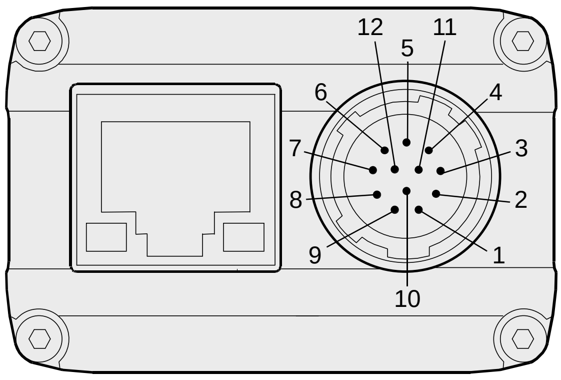


Fig. 23: Pin Numbering for the 12-pin Receptacle

7.3 Connector Types

7.3.1 8-pin RJ-45 Jack

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

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

Green and Yellow LEDs

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

7.3.2 12-pin Connector

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

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

7.4 Cabling Requirements

7.4.1 Ethernet Cables

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

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

Close proximity to strong magnetic fields should be avoided.

7.4.2 Standard Power and I/O Cable

Note

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

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

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

A single cable is used to connect power to the camera and to connect to the camera's I/O lines as shown in Figure 24.

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

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

Close proximity to strong magnetic fields should be avoided.

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

CAUTION

An Incorrect Plug Can Damage the 12-pin Connector

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

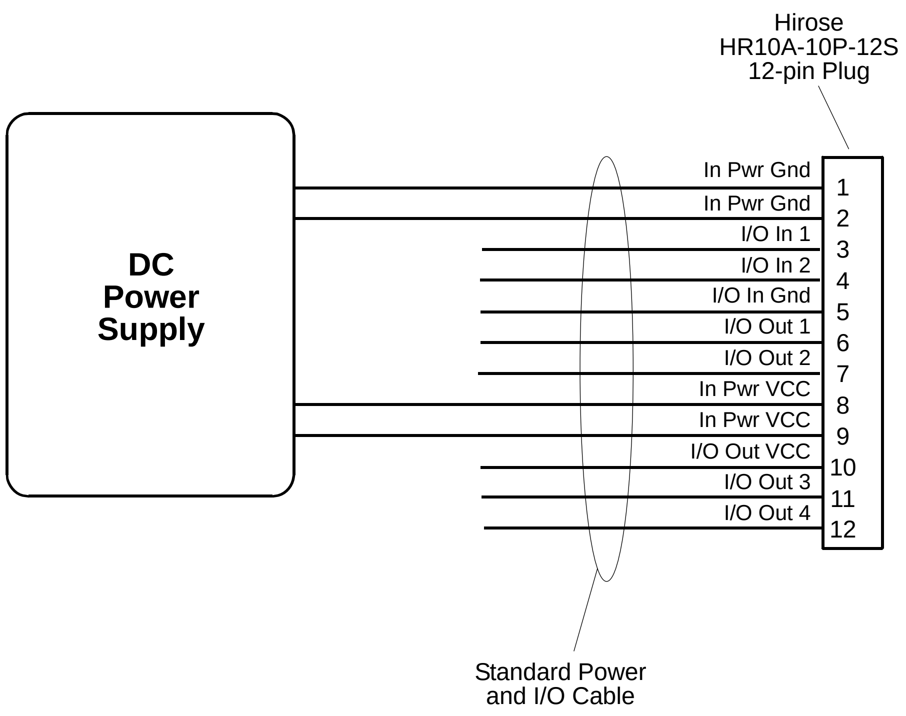


Fig. 24: Standard Power and I/O Cable

Note

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

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

7.4.3 PLC Power and I/O Cable

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

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

Close proximity to strong magnetic fields should be avoided.

Note

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

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

Basler offers PLC power and I/O cables with 3 m and 10 m lengths. Each cable is terminated with a 12-pin Hirose plug (HR10A-10P-12S) on the end that connects to the camera. The other end is unterminated. Contact your Basler sales representative to order the cables.

7.5 Camera Power

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

.

CAUTION

Voltage Outside of Specified Range Can Cause Damage

If the voltage of the power to the camera is greater than +30.0 VDC damage to the camera can result. If the voltage is less than +11.3 VDC, the camera may operate erratically.

CAUTION

An Incorrect Plug Can Damage the 12-pin Connector

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

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

Voltage	Significance
< +11.3 VDC	The camera may operate erratically.
+12 to +24 VDC	Recommended operating voltage; < 1 % ripple required. Make sure to use a power supply that supplies power in this voltage range.
+30.0 VDC	Absolute maximum; the camera may be damaged when the absolute maximum is exceeded.

Table 6: Voltage Requirements for the Camera Power VCC

For more information about the 12-pin connector and the power and I/O cables see Section 7.2 on page 62, Section 7.3 on page 64, and Section 7.4 on page 65.

7.6 Ethernet GigE Device Information

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

7.7 Input and Output Lines

7.7.1 Input Lines

7.7.1.1 Voltage Requirements

Note

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

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

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

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

Table 7: Voltage Requirements for the I/O Input When Using the Standard Power and I/O Cable

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

The following voltage requirements apply to the input of the PLC power and I/O cable. The PLC power and I/O cable will adjust the voltages to the levels required at the camera's I/O input (see Table 5).

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

Table 8: Voltage Requirements for the I/O Input When Using a PLC Power and I/O Cable

7.7.1.2 Line Schematic

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

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

Figure 25 shows an example of a typical circuit you can use to input a signal into the camera.

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

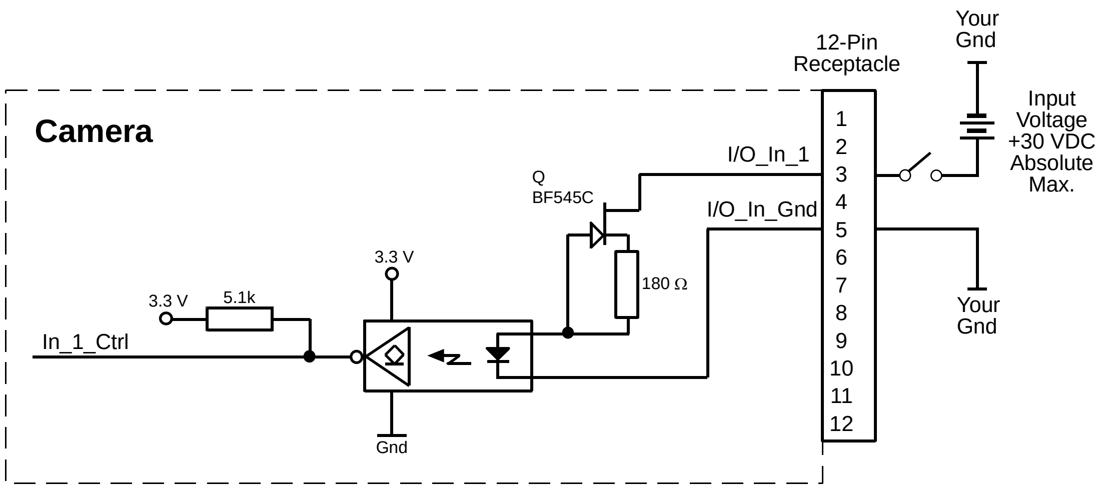


Fig. 25: Typical Input Circuit

For more information about input line pin assignments and pin numbering, see Section 7.2 on page 62.

For more information about how to use an ExTrig signal to control acquisition start, see Section 8.3 on page 84.

For more information about configuring the input lines, see Section 10.1 on page 137.

7.7.2 Output Lines

7.7.2.1 Voltage Requirements

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

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

Table 9: Voltage Requirements for the I/O Output VCC

7.7.2.2 Line Schematic

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

As shown in the I/O schematic, each output line is opto-isolated. See the previous section for the recommended operating voltage. The absolute maximum voltage is +30.0 VDC. The maximum current allowed through an output circuit is 100 mA.

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

Figure 26 shows a typical circuit you can use to monitor an output line with a voltage signal. The circuit in Figure 26 is monitoring output line 1.

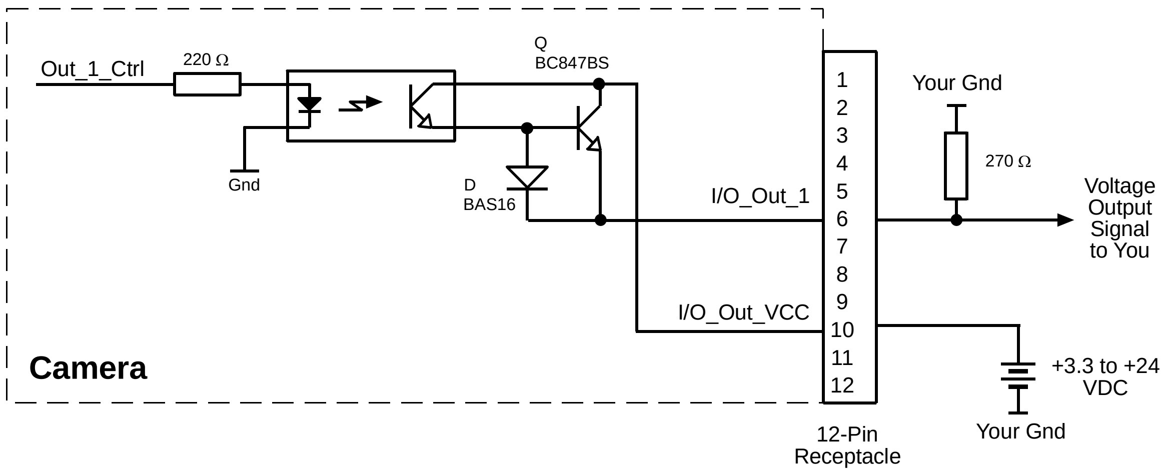


Fig. 26: Typical Voltage Output Circuit

Figure 27 shows a typical circuit you can use to monitor an output line with an LED or an opto-coupler. In this example, the voltage for the external circuit is +24 VDC. Current in the circuit is limited by an external resistor. The circuit in Figure 27 is monitoring output line 1.

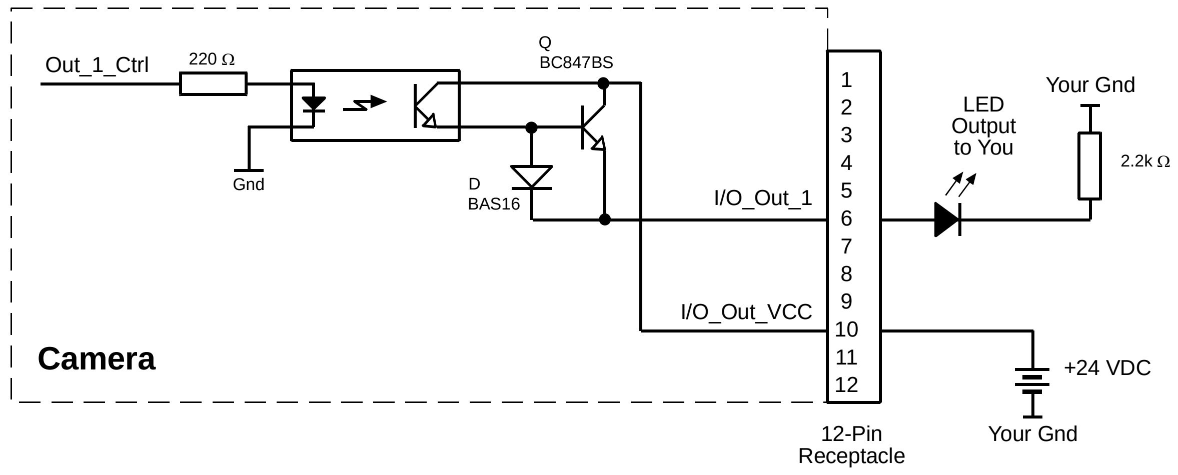


Fig. 27: Typical LED Output Signal at +24 VDC for the External Circuit (Example)

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

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

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

For more information about output line pin assignments and pin numbering, see Section 7.2 on page 62.

For more information about the exposure active signal, see Section Section 8.7 on page 98.

For more information about the trigger ready signal, see Section Section 8.6 on page 96.

For more information about assigning camera output signals to physical output lines, see Section 10.2.1 on page 139.

7.7.3 Output Line Response Time

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

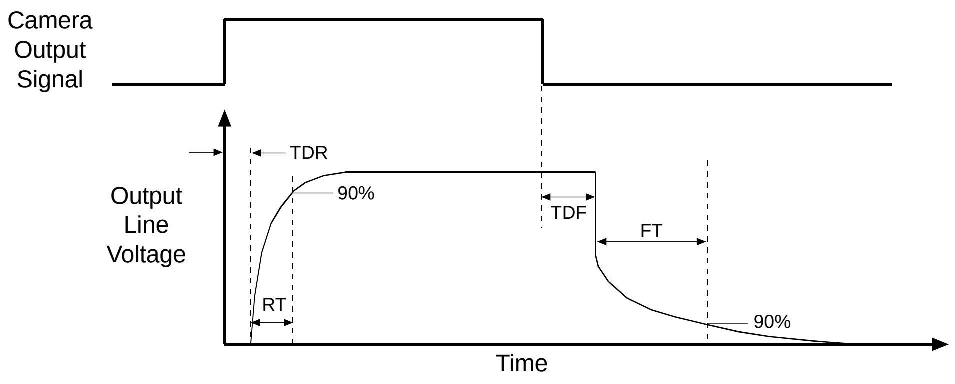


Fig. 28: Output Line Response Times

Time Delay Rise (TDR) = 1.5 µs

Rise Time (RT) = 1.3 - 5.0 µs

Time Delay Fall (TDF) = 1 - 20 µs

Fall Time (FT) = 1 - 5 µs

Note

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

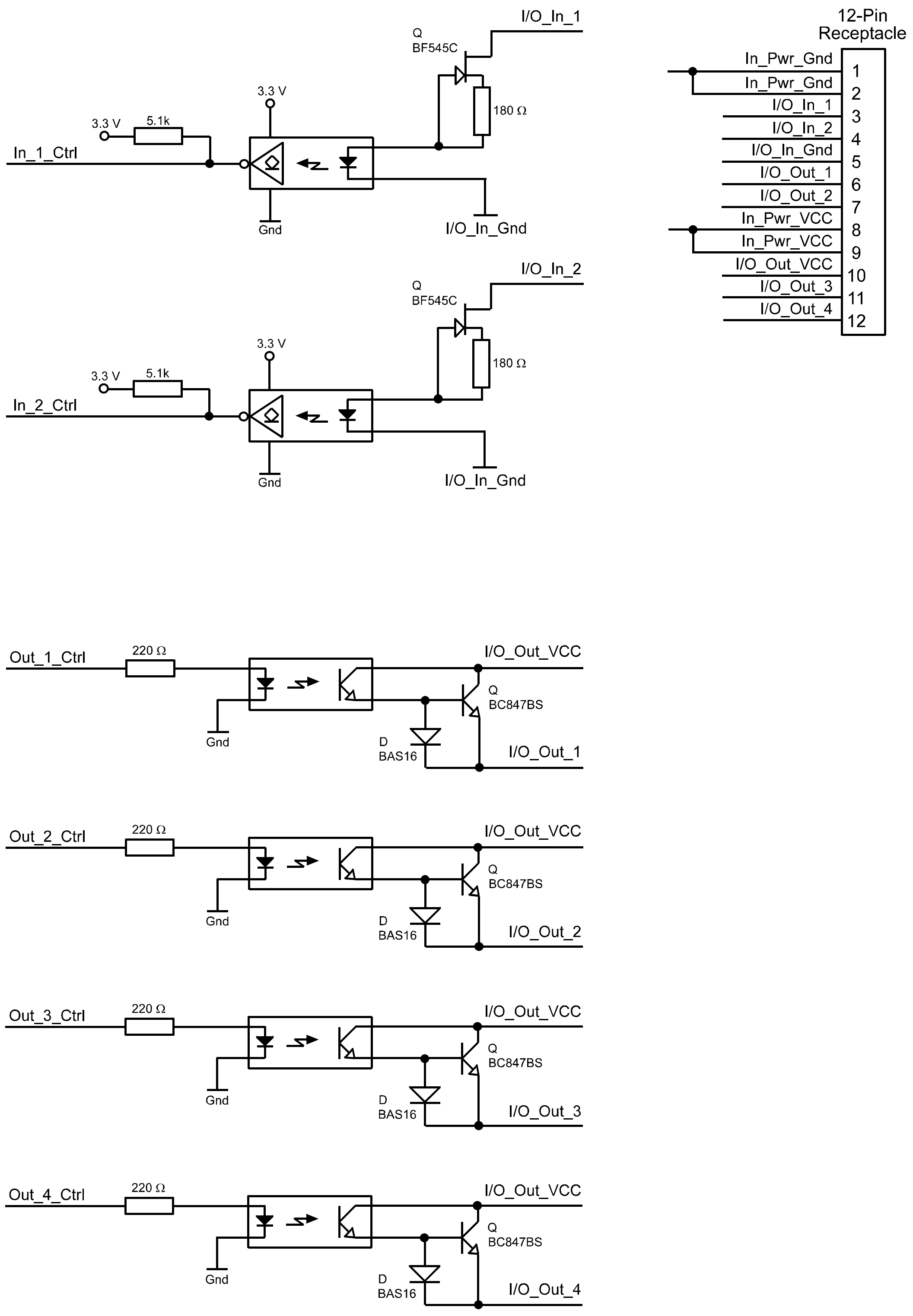


Fig. 29: I/O Line Schematic

8 Image Acquisition Control

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

8.1 Controlling Image Acquisition with Parameters Only (No Triggering)

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

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

8.1.1 Switching Off Triggering

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

8.1.2 Acquiring One Image at a Time

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about the camera’s exposure time parameter, see Section 8.4 on page 91.

8.1.3 Acquiring Images Continuously (Free-run)

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

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

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about the camera's exposure time parameter, see Section 8.4 on page 91.

For more information about determining the maximum allowed acquisition frame rate, see Section 8.9 on page 102.

Note

The explanations in Section 8.1.2 and Section 8.1.3 are intended to give you a basic idea of how parameters alone can be used to control image acquisition. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

8.2 Controlling Image Acquisition with a Software Trigger

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

Image acquisition starts when the Trigger Software command is executed. The exposure time for each image is determined by the value of the camera's exposure time parameter. Figure 30 illustrates image acquisition with a software trigger.

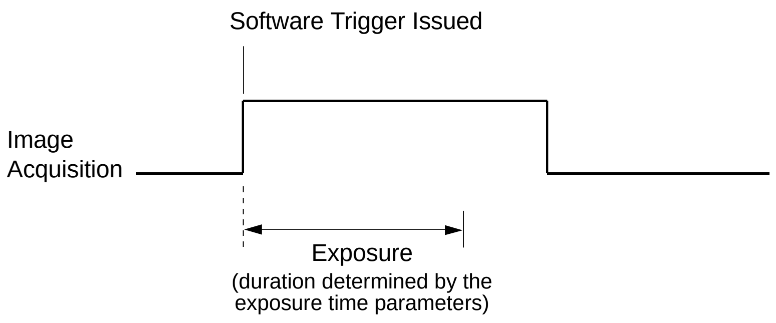


Fig. 30: Image Acquisition with a Software Trigger

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

8.2.1 Enabling the Software Trigger Feature

To enable the software trigger feature:

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

8.2.2 Acquiring a Single Image by Applying One Software Trigger

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about the camera's exposure time parameter, see Section 8.4 on page 91.

8.2.3 Acquiring Images by Applying a Series of Software Triggers

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

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

If you are acquiring images using a series of software triggers, you must avoid acquiring images at a rate that exceeds the maximum allowed with the current camera settings. You can use the Acquisition Status feature to determine when the camera is ready to be triggered for the next image acquisition.

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

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

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

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

// how to set and test the Acquisition Frame Rate
```

```
Camera.AcquisitionFrameRateAbs.SetValue( 60.0 );
double resultingFrameRate = Camera.ResultingFrameRateAbs.GetValue( );

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about the camera's exposure time parameter, see Section 8.4 on page 91.

For more information about determining the maximum allowed acquisition frame rate, see Section 8.9 on page 102.

Note

The explanations in Section 8.2.2 and Section 8.2.3 are intended to give you a basic idea of how the use of a software trigger works. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

8.3 Controlling Image Acquisition with a Hardware Trigger

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

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

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

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

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

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

In order for the camera to detect a transition from low to high, the ExTrig signal must be held high for at least 100 nanoseconds. In order for the camera to detect a transition from high to low, the ExTrig signal must be held low for at least 100 nanoseconds.

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

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

For more information about setting the camera for hardware triggering and selecting the input line to receive the ExTrig signal, see Section 8.3.2 on page 87.

For more information about determining the maximum allowed acquisition frame rate, see Section 8.9 on page 102.

8.3.1 Exposure Modes

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

Timed Exposure Mode

When timed mode is selected, the exposure time for each image is determined by the value of the camera’s exposure time parameter. If the camera is set for rising edge triggering, the exposure time starts when the ExTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExTrig signal falls. Figure 31 illustrates timed exposure with the camera set for rising edge triggering.

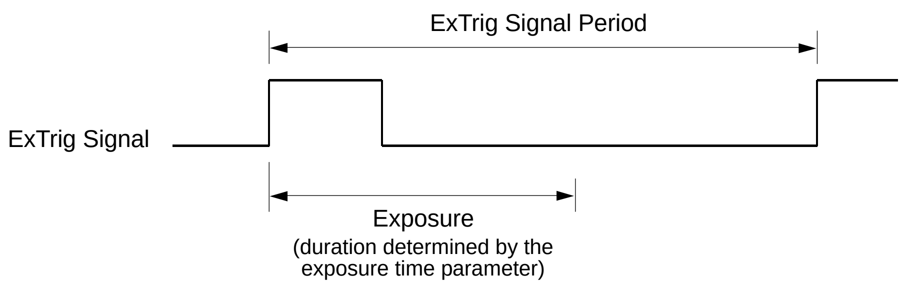


Fig. 31: Timed Exposure with Rising Edge Triggering

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure will be directly controlled by the ExTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExTrig signal rises and continues until the ExTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExTrig signal falls and continues until the ExTrig signal rises. Figure 32 illustrates trigger width exposure with the camera set for rising edge triggering.

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

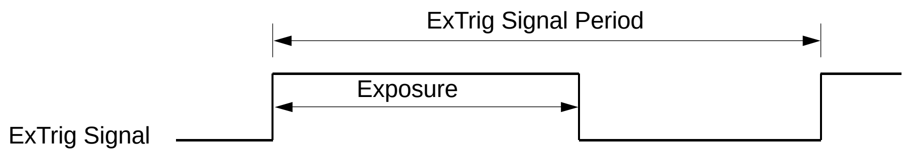


Fig. 32: Trigger Width Exposure with Rising Edge Triggering

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

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

If you are using the trigger width exposure mode and the camera is operating with overlapped exposures, there is something you must keep in mind. If the action of the ExTrig signal would end the current exposure while readout of the previously acquired image is still taking place, the camera will automatically continue the exposure until readout of the previous image is complete. This situation is illustrated Figure 31 for rising edge operation. On the first cycle of the ExTrig signal shown in the figure, the signal rises and falls while readout is taking place. Normally you would expect exposure to take place only when the ExTrig signal is high. But since the signal falls while the previous frame is still reading out, the camera automatically extends exposure until the readout is complete. On the second cycle of the ExTrig signal shown in the figure, the signal rises during previous frame readout, but falls after the readout is complete. This is a normal situation and exposure would be determined by the high time of the ExTrig signal as you would expect.

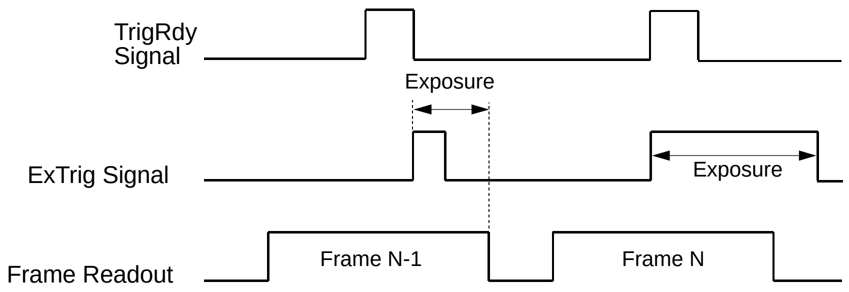


Fig. 33: Trigger Width Exposure Mode with Overlapped Exposure

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

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

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

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

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

For more information about the pylon viewer, see Section 3.1 on page 29.

For more information about the camera's exposure time parameter, see Section 8.4 on page 91.

For more information about overlapped exposure, see Section 8.5 on page 94.

For more detailed information about using the trigger width exposure mode with overlapped exposure, refer to the application notes called "Using a Specific External Trigger Signal with Overlapped Exposure" (AW000565xx000). The application notes are available in the downloads section of the Basler website: www.baslerweb.com.

8.3.2 Setting the Camera for Hardware Triggering

To set the camera for hardware triggering:

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

8.3.3 Acquiring a Single Image by Applying One Hardware Trigger Transition

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about the Trigger Ready signal, see Section 8.6 on page 96.

For more information about the camera's exposure time parameter, see Section 8.4 on page 91.

8.3.4 Acquiring Images by Applying a Series of Hardware Trigger Transitions

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

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

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about the Trigger Ready signal, see Section 8.6 on page 96.

For more information about the camera’s exposure time parameter, see Section 8.4 on page 91.

Note

The explanations in Section 8.3.3 and Section 8.3.4 are intended to give you a basic idea of how the use of a hardware trigger works. For a more complete description, refer to the Basler pylon Programmer’s Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

8.4 Exposure Time Parameters

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

The exposure time must not be set below a minimum specified value. The minimum exposure time for each camera model is shown in Table 10.

The maximum exposure time that can be set is also shown in Table 10.

Camera Model	Minimum Allowed Exposure Time	Maximum Possible Exposure Time
piA640-210gm/gc	28 µs	10000000 µs
piA1000-48gm/gc	24 µs	10000000 µs
piA1600-35gm/gc	50 µs	10000000 µs
piA1900-32gm/gc	69 µs	10000000 µs
piA2400-12gm/gc	45 µs	10000000 µs
piA2400-17gm/gc	29 µs	10000000 µs

Table 10: Minimum Allowed Exposure Time and Maximum Possible Exposure Time

Note

Exposure time can not only be manually set (see below), but can also be automatically adjusted.

Exposure Auto is an auto function and the "automatic" counterpart to manually setting an "absolute" exposure time. The exposure auto function automatically adjusts the Auto Exposure Time Abs parameter value.

In contrast to the manually set "absolute" exposure time, the automatically adjusted "absolute" exposure time is not restricted to multiples of the current exposure time base. The automatic adjustment is not available when trigger width exposure mode is selected.

For more information about auto functions, see Section 11.11.1 on page 196.

For more information about the Exposure Auto function, see Section 11.11.3 on page 205.

For information on parameter settings for obtaining the maximum possible exposure time, see Section 8.4.1 on page 92.

8.4.1 Setting the Exposure Time Using "Raw" Settings

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

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

By default, the Exposure Time Base is fixed at 20 µs. Typically, the exposure time is adjusted by setting only the Exposure Time Raw parameter.

The Exposure Time Raw parameter value can range from 1 to 4095. So if the value is set to 100, for example, the exposure time will be 100 x 20 µs or 2000 µs.

Settings for Obtaining the Maximum Possible Exposure Time

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

Changing the Exposure Time Base

By default, the Exposure Time Base is fixed at 20 µs, and the exposure time is normally adjusted by setting the value of the Exposure Time Raw parameter. However, if you require an exposure time that is longer than what you can achieve by changing the value of the Exposure Time Raw parameter alone, the Exposure Time Base Abs parameter can be used to change the exposure time base.

The Exposure Time Base Abs parameter value sets the exposure time base in µs. The exposure time base can be changed in 1 µs increments and the default is 20 µs.

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

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

Camera.ExposureTimeBaseAbs.SetValue( 200 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

8.4.2 Setting the Exposure Time Using "Absolute" Settings

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

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

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

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

Setting the Absolute Exposure Time Parameter

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

8.5 Overlapping Exposure and Sensor Readout

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

In regard to this image acquisition process, there are two common ways for the camera to operate: with “non-overlapped” exposure and with “overlapped” exposure. In the non-overlapped mode of operation, each time an image is acquired, the camera completes the entire exposure/readout process before acquisition of the next image is started. This situation is illustrated in Figure 34.

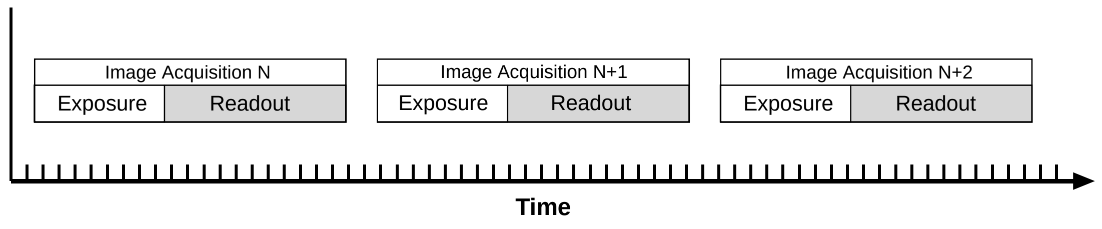


Fig. 34: Non-overlapped Exposure

While operating in a non-overlapped fashion is perfectly normal and is appropriate for many situations, it is not the most efficient way to operate the camera in terms of acquisition frame rate. On this camera, however, it is allowable to begin exposing a new image while a previously acquired image is being read out. This situation is illustrated in Figure 35 and is known as operating the camera with “overlapped” exposure.

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

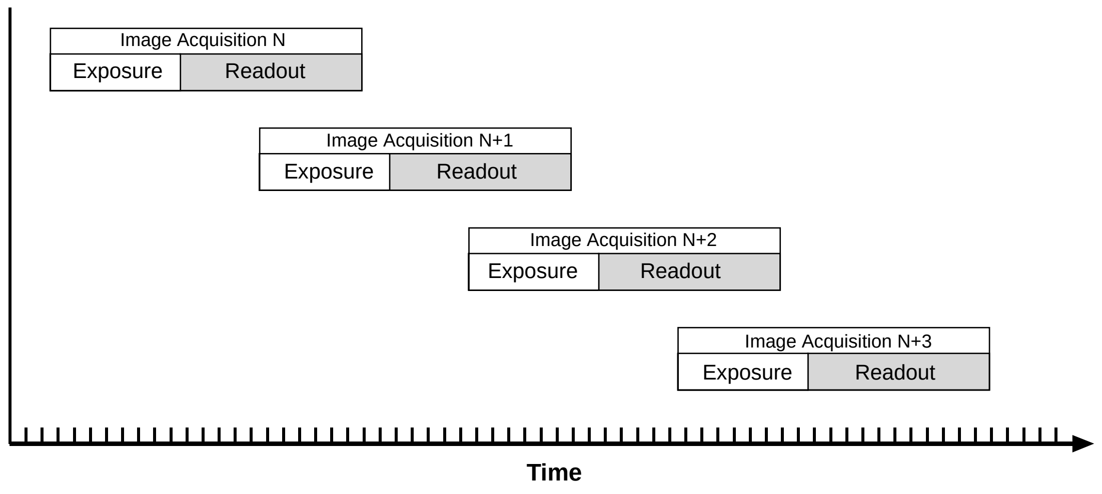


Fig. 35: Overlapped Exposure

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

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

You can calculate the readout time for a captured image by using the formula on page 99.

8.5.1 Guidelines for Overlapped Operation

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

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

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

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

For more information about using the Trigger Ready signal, see Section 8.6 on page 96.

For more detailed guidelines about using an external trigger signal with the trigger width exposure mode and overlapped exposure, refer to the application notes called "Using a Specific External Trigger Signal with Overlapped Exposure" (AW000565xx000). The application notes are available in the downloads section of the Basler website: www.baslerweb.com.

8.6 Trigger Ready Signal

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

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

The camera supplies a “Trigger Ready” (TrigRdy) output signal you can use to ensure that these conditions are met when you are using a hardware trigger signal to trigger image acquisition. When you are acquiring images, the camera automatically calculates the earliest moment that it is safe to trigger each new acquisition. The trigger ready signal will go high when it is safe to trigger an acquisition, will go low when the acquisition has started, and will go high again when it is safe to trigger the next acquisition (see Figure 36). The camera calculates the rise of the trigger ready signal based on the current exposure time parameter setting, the current size of the area of interest, and the time it will take to readout the captured pixel values from the sensor.

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

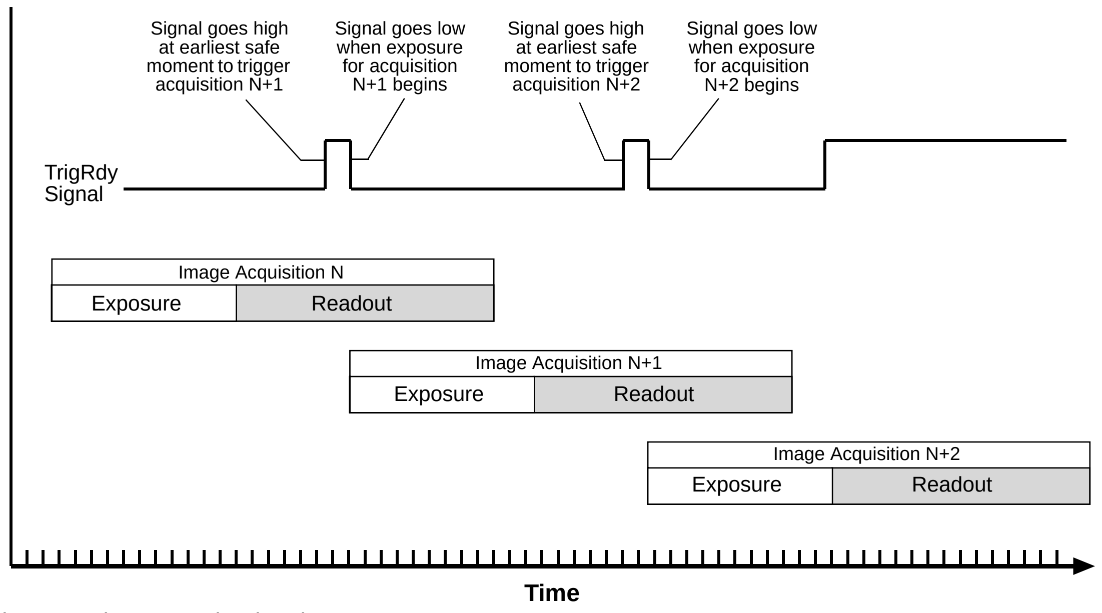

Signal goes high at earliest safe moment to trigger acquisition N+1
Signal goes low when exposure for acquisition N+1 begins
Signal goes high at earliest safe moment to trigger acquisition N+2
Signal goes low when exposure for acquisition N+2 begins
TrigRdy Signal
Image Acquisition N
Exposure
Readout
Image Acquisition N+1
Exposure
Readout
Image Acquisition N+2
Exposure
Readout
Time

Fig. 36: Trigger Ready Signal

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

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

Note

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

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

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

For more information about changing the assignment of camera output signals to physical output lines, see Section 10.2.1 on page 139.

For more information about the electrical characteristics of the camera's output lines, see Section 7.7.2 on page 73.

8.7 Exposure Active Signal

The camera’s “exposure active” (ExpAc) signal goes high when the exposure time for each image acquisition begins and goes low when the exposure time ends as shown in Figure 37. This signal can be used as a flash trigger and is also useful when you are operating a system where either the camera or the object being imaged is movable. For example, assume that the camera is mounted on an arm mechanism and that the mechanism can move the camera to view different portions of a product assembly. Typically, you do not want the camera to move during exposure. In this case, you can monitor the ExpAc signal to know when exposure is taking place and thus know when to avoid moving the camera.

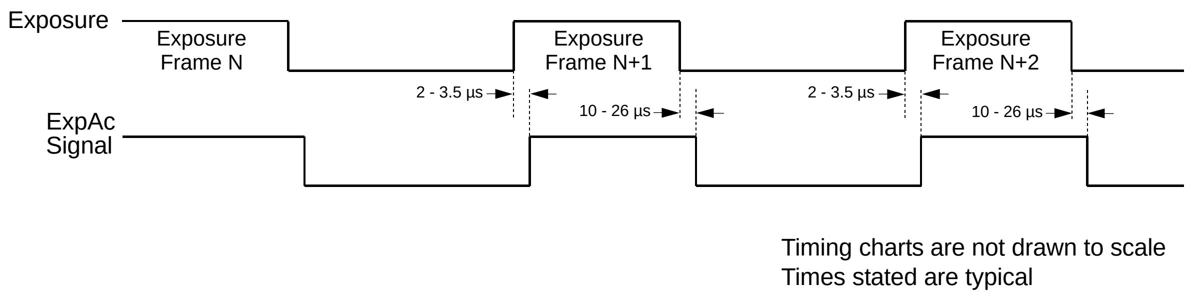


Fig. 37: Exposure Active Signal

Note

When you use the exposure active signal, be aware that there is a delay in the rise and the fall of the signal in relation to the start and the end of exposure. See Figure 37 for details.

By default, the ExpAc signal is assigned to physical output line 1 on the camera. However, the assignment of the ExpAc signal to a physical output line can be changed.

For more information about changing the assignment of camera output signals to physical output lines, see Section 10.2.1 on page 139.

For more information about the electrical characteristics of the camera’s output lines, see Section 7.7.2 on page 73.

8.8 Acquisition Timing Chart

Figure 38 shows a timing chart for image acquisition and transmission. The chart assumes that exposure is triggered by an ExTrig signal with rising edge activation and that the camera is set for the timed exposure mode.

As Figure 38 shows, there is a slight delay between the rise of the ExTrig signal and the start of exposure. After the exposure time for an image acquisition is complete, the camera begins reading out the acquired image data from the sensor into a buffer in the camera. When the camera has determined that a sufficient amount of image data has accumulated in the buffer, it will begin transmitting the data from the camera to the host PC.

This buffering technique avoids the need to exactly synchronize the clock used for sensor readout with the data transmission over your Ethernet network. The camera will begin transmitting data when it has determined that it can safely do so without over-running or under-running the buffer. This buffering technique is also an important element in achieving the highest possible frame rate with the best image quality.

The **exposure start delay** is the amount of time between the point where the trigger signal transitions and the point where exposure actually begins.

The **frame readout time** is the amount of time it takes to read out the data for an acquired image from the sensor into the image buffer.

The **frame transmission time** is the amount of time it takes to transmit the acquired image from the buffer in the camera to the host PC via the network.

The **transmission start delay** is the amount of time between the point where the camera begins reading out the acquired image data from the sensor to the point where it begins transmitting the data for the acquired image from the buffer to the host PC.

Note that, if the averaging feature is used, the concept of the transmission start delay, as described above, does not apply. In this case, the acquired images are not transmitted individually but will be used for creating an averaged image which is transmitted.

The exposure start delay varies from camera model to camera model. The table below shows the exposure start delay for each camera model:

Camera Model	Exposure Start Delay
piA640-210gm/gc	23.64 µs
piA1000-48gm/gc	24.64 µs
piA1600-35gm/gc	65.98 µs
piA1900-32gm/gc	101.45 µs
piA2400-12gm/gc	66.60 µs
piA2400-17gm/gc	32.06 µs

Table 11: Exposure Start Delays

Note that, if the debouncer feature is used, the debouncer setting for the input line must be added to the exposure start delays shown in Table 11 to determine the total start delay. For example, assume that you are using an piA640-210 camera and that you have set the cameras for hardware triggering. Also assume that you have selected input line 1 to accept the hardware trigger signal and that you have set the Line Debouncer Time Abs parameter for input line 1 to 5 µs. In this case:

Total Start Delay = Start Delay from Table 11+ Debouncer Setting

Total Start Delay = 12.44 µs+ 5 µs

Total Start Delay = 17.44 µs

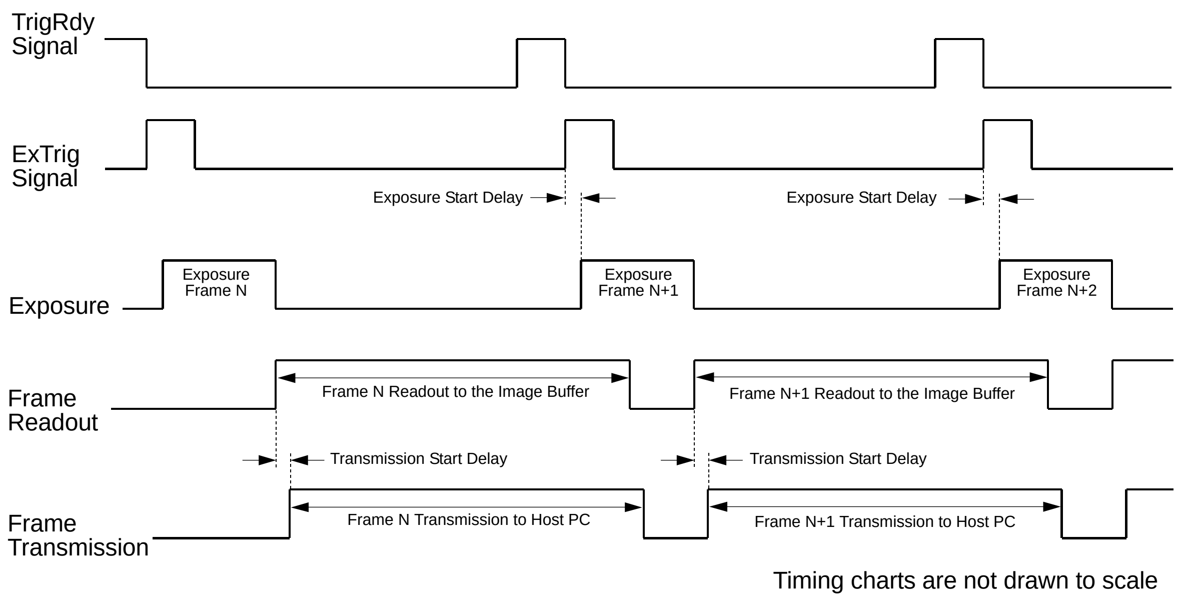


Fig. 38: Exposure Start Controlled with an ExTrig Signal

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

$$\text{Frame Readout Time} = (\text{AOI Height} \times C_1\ \mu s) + C_2\ \mu s$$

Where the values for the constants C_1 and C_2 are from the table in Section 8.9 on page 102 and AOI height is the height of the acquired frames as determined by the AOI settings.

For more information about the AOI height, see Section 11.6 on page 163.

For more information about the averaging feature, see Section 11.9 on page 173.

You can calculate an approximate frame transmission time by using this formula:

$$\sim \text{Frame Transmission Time} = \frac{\text{Payload Size Parameter Value}}{\text{Device Current Throughput Parameter Value}}$$

Note that this is an approximate frame transmission time. Due to the nature of the Ethernet network, the transmission time could vary. Also note that the frame transmission cannot be less than the frame readout time. So if the frame transmission time formula returns a value that is less than the readout time, the approximate frame transmission time will be equal to the readout time.

Due to the nature of the Ethernet network, the transmission start delay can vary from frame to frame. The start delay, however, is of very low significance when compared to the transmission time.

For more information about the Payload Size and Device Current Throughput parameters, see Section 5.1 on page 43.

8.9 Maximum Allowed Acquisition Frame Rate

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

- The amount of time it takes to read the data for an acquired image (known as a frame) from the image sensor to the frame buffer. This time varies depending on the height of the frame. Shorter frames take less time to read out of the sensor. The frame height is determined by the camera's AOI 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.

Note

When the averaging feature is used, an increased acquisition frame rate can be achieved if the frame transmission is the most limiting factor. The acquired images are not transmitted individually but will be used for creating an averaged image. The averaged image will be transmitted at an output frame rate which will be subject to the frame transmission time and will be lower than the acquisition frame rate.

To determine the maximum allowed acquisition frame rate with your current camera settings, you can read the value of the camera's Resulting Frame Rate parameter. This parameter indicates the camera's current maximum allowed frame rate taking into account the AOI, exposure time, bandwidth settings, and whether the averaging feature is enabled .

For more information about AOI settings, see Section 11.6 on page 163.

For more information about the Resulting Frame Rate parameter, see Section 5.1 on page 43.

For more information about the averaging feature, see Section 11.9 on page 173.

Increasing the Maximum Allowed Frame Rate

You may find that you would like to acquire frames at a rate higher than the maximum allowed with the camera's current settings. In this case, you must first use the three formulas described below to determine what factor is restricting the maximum frame rate the most. Next, you must try to make that factor less restrictive:

- You will often find that the sensor readout time is most restrictive factor. Decreasing the AOI height for the acquired frames will decrease the sensor readout time and will make this factor less restrictive.
- If you are using normal exposure times and you are using the camera at it's maximum resolution, your exposure time will not normally be the most restrictive factor on the frame rate. However, if you are using long exposure times or small areas of interest, it is quite possible to

find that your exposure time is the most restrictive factor on the frame rate. In this case, you should lower your exposure time. (You may need to compensate for a lower exposure time by using a brighter light source or increasing the opening of your lens aperture.)

- The frame transmission time will not normally be a restricting factor. But if you are using multiple cameras and you have set a small packet size or a large inter-packet delay, you may find that the transmission time is restricting the maximum allowed rate. In this case, you could increase the packet size or decrease the inter-packet delay. If you are using several cameras connected to the host PC via a network switch, you could also use a multiport network adapter in the PC instead of a switch. This would allow you to increase the Ethernet bandwidth assigned to the camera and thus decrease the transmission time.

For more information about AOI settings, see Section 11.6 on page 163.

For more information on the settings that determine the bandwidth assigned to the camera, see Section 5.2 on page 50.

Formula 1:

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

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

Where:

AOI Height = the height of the acquired frames as determined by the AOI settings.

The constants C_1 and C_2 depend on the camera model as shown in the table below:

	C_1	C_2
piA640-210gm/gc	8.76 µs	521.17 µs
piA1000-48gm/gc	13.39 µs	7423.76 µs
piA1600-35gm/gc	20.52 µs	3873.2 µs
piA1900-32gm/gc*	0 µs	31021.26 µs
piA2400-12gm/gc	26.19 µs	26210.09 µs
piA2400-17gm/gc	20.94 µs	15413.35 µs

* Note: The maximum frame rate of the piA1900-32gm/gc is limited to 32 fps.

Formula 2:

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

$$\text{Max. Frames/s} = \frac{1}{\text{Exposure time in µs} + C_3}$$

Where:

The constant C_3 depends on the camera model as shown in the table below:

	C_3
piA640-210gm/gc	46.99 µs
piA1000-48gm/gc	95.57 µs
piA1600-35gm/gc	79.64 µs
piA1900-32gm/gc	139.38 µs
piA2400-12gm/gc	102.40 µs
piA2400-17gm/gc	81.92 µs

For more information about setting the exposure time, see Section 8.4 on page 91.

Formula 3:

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

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

Note

When the averaging feature is used, the above formula is replaced by the related formula in the "Averaging" section, which may permit a higher maximum acquisition frame rate.

For the related formula when the averaging feature is used, see Section 11.9 on page 173.

Example

Assume that you are using a piA640-210gm camera set for an exposure time of 2000 µs and for 600 x 400 resolution. Also assume that you have checked the value of the Device Current Throughput parameter, the Payload Size parameters and found them to be 110000000 and 240000 respectively, and the averaging feature is not used.

Formula 1:

$$\text{Max Frames/s} = \frac{1}{(400 \times 8.76\ \mu s) + 521.17\ \mu s}$$

Max Frames/s = 248.4 frames/s

Formula 2:

$$\text{Max Frames/s} = \frac{1}{2000\ \mu s + 46.99\ \mu s}$$

Max Frames/s = 488.5 frames/s

Formula 3:

$$\text{Max Frames/s} = \frac{110000000}{240000}$$

Max Frames/s = 458.3 frames/s

Formula one returns the lowest value. So in this case, the limiting factor is the sensor readout time and the maximum allowed acquisition frame rate would be 248.4 frames per second.

9 Pixel Data Formats

By selecting a pixel data format, you determine the format (layout) of the image data transmitted by the camera. This section provides detailed information about the available pixel data formats.

9.1 Setting the Pixel Data Format

The setting for the camera's Pixel Format parameter determines the format of the pixel data that will be output from the camera. The available pixel formats depend on the camera model and whether the camera is monochrome or color. Table 12 lists the pixel formats available on each monochrome camera model and Table 13 lists the pixel formats available on each color camera model.

Mono Camera Model	Mono 8	Mono 16	Mono 12 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
piA640-210gm	●	●	●	●	●
piA1000-48gm	●	●	●	●	●
piA1600-35gm	●	●	●	●	●
piA1900-32gm	●	●	●	●	●
piA2400-12gm	●	●	●	●	●
piA2400-17gm	●	●	●	●	●

Table 12: Pixel Formats Available on Monochrome Cameras (• = format available)

Color Camera Model	Mono 8	Bayer GB 8	Bayer BG 8	Bayer GB 16	Bayer BG 16	Bayer GB 12 Packed	Bayer BG 12 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
piA640-210gc	●	●		●		●		●	●
piA1000-48gc	●	●		●		●		●	●
piA1600-35gc	●	●		●		●		●	●
piA1900-32gc	●	●		●		●		●	●
piA2400-12gc	●		●		●		●	●	●
piA2400-17gc	●		●		●		●	●	●

Table 13: Pixel Formats Available on Color Cameras (• = format available)

Details of the monochrome formats are described in Section 9.2 on page 109 and details of the color formats are described in Section 9.3 on page 115.

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

```
Camera.PixelFormat.SetValue( PixelFormat_Mono8 );
Camera.PixelFormat.SetValue( PixelFormat_Mono12Packed );
Camera.PixelFormat.SetValue( PixelFormat_Mono16 );
Camera.PixelFormat.SetValue( PixelFormat_YUV422Packed );
Camera.PixelFormat.SetValue( PixelFormat_YUV422_YUYV_Packed );
Camera.PixelFormat.SetValue( PixelFormat_BayerGB8 );
Camera.PixelFormat.SetValue( PixelFormat_BayerGB16 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

9.2 Pixel Data Formats for Mono Cameras

9.2.1 Mono 8 Format (Equivalent to DCAM Mono 8)

When a monochrome camera is set for the Mono 8 pixel data format, it outputs 8 bits of brightness data per pixel.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono 8 output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data
B_0	Brightness value for P_0
B_1	Brightness value for P_1
B_2	Brightness value for P_2
B_3	Brightness value for P_3
B_4	Brightness value for P_4
•	•
•	•

Byte	Data
•	•
•	•
B_{m-4}	Brightness value for P_{n-4}
B_{m-3}	Brightness value for P_{n-3}
B_{m-2}	Brightness value for P_{n-2}
B_{m-1}	Brightness value for P_{n-1}
B_m	Brightness value for P_n

With the camera set for Mono 8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	Indicates This Signal Level (Decimal)
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

9.2.2 Mono 16 Format (Equivalent to DCAM Mono 16)

When a monochrome camera is set for the Mono16 pixel data format, it outputs 16 bits of brightness data per pixel with 12 bits effective. The 12 bits of effective pixel data fill from the least significant bit. The four unused most significant bits are filled with zeros.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono16 output. Note that the data is placed in the image buffer in **little endian format**.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data
B_0	Low byte of brightness value for P_0
B_1	High byte of brightness value for P_0
B_2	Low byte of brightness value for P_1
B_3	High byte of brightness value for P_1
B_4	Low byte of brightness value for P_2
B_5	High byte of brightness value for P_2
B_6	Low byte of brightness value for P_3
B_7	High byte of brightness value for P_3
B_8	Low byte of brightness value for P_4
B_9	High byte of brightness value for P_4
•	•
•	•
•	•
B_{m-7}	Low byte of brightness value for P_{n-3}
B_{m-6}	High byte of brightness value for P_{n-3}
B_{m-5}	Low byte of brightness value for P_{n-2}
B_{m-4}	High byte of brightness value for P_{n-2}
B_{m-3}	Low byte of brightness value for P_{n-1}
B_{m-2}	High byte of brightness value for P_{n-1}
B_{m-1}	Low byte of brightness value for P_n
B_m	High byte of brightness value for P_n

When the camera is set for Mono 16, the pixel data output is 16 bit data of the “unsigned short (little endian)” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below. Note that for 16 bit data, you might expect a value range from 0x0000 to 0xFFFF. However, with the camera set for Mono16 only 12 bits of the 16 bits transmitted are effective. Therefore, the highest data value you will see is 0x0FFF indicating a signal level of 4095.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

Note

When a camera that is set for Mono 16 has only 12 bits effective, the leader of transmitted frames will indicate Mono 12 as the pixel format.

9.2.3 Mono 12 Packed Format

When a monochrome camera is set for the Mono 12 Packed pixel data format, it outputs 12 bits of brightness data per pixel. Every three bytes transmitted by the camera contain data for two pixels.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono 12 Packed output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data	
B_0	P_0 bits 11 ... 4	
B_1	P_1 bits 3 ... 0	P_0 bits 3 ... 0
B_2	P_1 bits 11 ... 4	
B_3	P_2 bits 11 ... 4	
B_4	P_3 bits 3 ... 0	P_2 bits 3 ... 0
B_5	P_3 bits 11 ... 4	
B_6	P_4 bits 11 ... 4	
B_7	P_5 bits 3 ... 0	P_4 bits 3 ... 0
B_8	P_5 bits 11 ... 4	
B_9	P_6 bits 11 ... 4	
B_{10}	P_7 bits 3 ... 0	P_6 bits 3 ... 0
B_{11}	P_7 bits 11 ... 4	
•	•	
•	•	•
•	•	
B_{m-5}	P_{n-3} bits 11 ... 4	
B_{m-4}	P_{n-2} bits 3 ... 0	P_{n-3} bits 3 ... 0
B_{m-3}	P_{n-2} bits 11 ... 4	
B_{m-2}	P_{n-1} bits 11 ... 4	
B_{m-1}	P_n bits 3 ... 0	P_{n-1} bits 3 ... 0
B_m	P_n bits 11 ... 4	

When a monochrome camera is set for Mono 12 Packed, the pixel data output is 12 bit data of the “unsigned” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

9.2.4 YUV 4:2:2 Packed Format (Equivalent to DCAM YUV 4:2:2)

When a monochrome camera is set for the YUV 4:2:2 Packed pixel data format, the camera transmits Y, U, and V values in a fashion that mimics the output from a color camera set for YUV 4:2:2 Packed.

The Y value transmitted for each pixel is an actual 8 bit brightness value similar to the pixel data transmitted when a monochrome camera is set for Mono 8. The U and V values transmitted will always be zero. With this format, a Y value is transmitted for each pixel, but the U and V values are only transmitted for every second pixel.

The order of the pixel data for a received frame in the image buffer in your PC is similar to the order of YUV 4:2:2 Packed output from a color camera.

For more information about the YUV 4:2:2 Packed format on color cameras, see Section 9.3.8 on page 129.

9.2.5 YUV 4:2:2 (YUYV) Packed Format

When a monochrome camera is set for the YUV 4:2:2 (YUYV) Packed pixel data format, the camera transmits Y, U, and V values in a fashion that mimics the output from a color camera set for YUV 4:2:2 (YUYV) Packed.

The Y value transmitted for each pixel is an actual 8 bit brightness value similar to the pixel data transmitted when a monochrome camera is set for Mono 8. The U and V values transmitted will always be zero. With this format, a Y value is transmitted for each pixel, but the U and V values are only transmitted for every second pixel.

The order of the pixel data for a received frame in the image buffer in your PC is similar to the order of YUV 4:2:2 (YUYV) Packed output from a color camera.

For more information about the YUV 4:2:2 (YUYV) Packed format on color cameras, see Section 9.3.9 on page 132.

9.3 Pixel Data Output Formats for Color Cameras

9.3.1 The Bayer Color Filter

The sensor used in color models of the camera is equipped with an additive color separation filter known as a Bayer filter. The pixel data output formats available on color cameras are related to the Bayer pattern, so you need a basic knowledge of the Bayer filter to understand the pixel formats. With the Bayer filter, each individual pixel is covered by a micro-lens that allows light of only one color to strike the pixel. The pattern of the Bayer filter used on the camera is as shown in Figure 39 (the alignment of the Bayer filter with repect 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.)

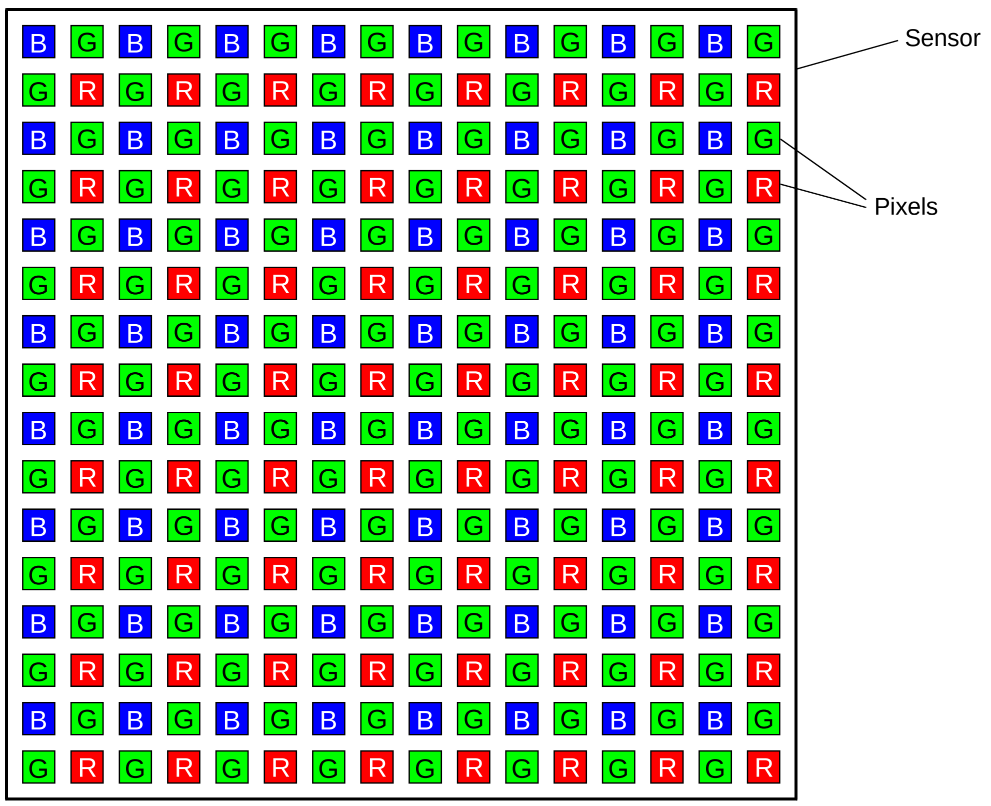

Sensor
Pixels
B G B G B G B G B G B G B G B G
G R G R G R G R G R G R G R G R

Fig. 39: Bayer Filter Pattern

9.3.1.1 Color Filter Alignment

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

Color Camera Model	Filter Alignment
piA640-210	GB
piA1000-48	GB
piA1600-35	GB
piA1900-32	GB
piA2400-12	BG
piA2400-17	BG

Table 14: Bayer Filter to Sensor Alignment

Bayer GB alignment means that pixel zero and pixel one of the first line in each image transmitted will be green and blue respectively. And for the second line transmitted, pixel zero and pixel one will be red and green respectively. Since the pattern of the Bayer filter is fixed, you can use this information to determine the color of all of the other pixels in the image.

Bayer BG alignment means that pixel zero and pixel one of the first line in each image transmitted will be blue and green respectively. And for the second line transmitted, pixel zero and pixel one 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.

Because the size and position of the area of interest on color cameras must be adjusted in increments of 2, the color filter alignment will remain the same regardless of the camera's area of interest (AOI) settings.

The Pixel Color Filter parameter indicates the current alignment of the camera's Bayer filter to the pixels in the images captured by a color camera. You can tell how the current AOI is aligned to the Bayer filter by reading the value of the Pixel Color Filter parameter.

For more information about the camera's AOI feature, see Section 11.6 on page 163.

9.3.2 Bayer GB 8 Format (Equivalent to DCAM Raw 8)

When a color camera is set for the Bayer GB 8 pixel data format, it outputs 8 bits of data per pixel and the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 8 bits of red data. For each pixel covered with a green lens, you get 8 bits of green data. And for each pixel covered with a blue lens, you get 8 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "GB" in the name Bayer GB 8 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel zero will be green, pixel one will be blue, pixel two will be green, pixel three will be blue, etc. For odd lines in the images, pixel zero will be red, pixel one will be green, pixel two will be red, pixel three will be green, etc.

For more information about the Bayer filter, see Section 9.3.1 on page 115.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer GB 8 output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

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

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Green value for P_0
B_1	Blue value for P_1
B_2	Green value for P_2
B_3	Blue value for P_3
B_4	Green value for P_4
B_5	Blue value for P_5
•	•
•	•
•	•
B_{m-5}	Green value for P_{n-5}
B_{m-4}	Blue value for P_{n-4}
B_{m-3}	Green value for P_{n-3}
B_{m-2}	Blue value for P_{n-2}
B_{m-1}	Green value for P_{n-1}
B_m	Blue value for P_n

Odd Lines	
Byte	**Data**
B_0	Red value for P_0
B_1	Green value for P_1
B_2	Red value for P_2
B_3	Green value for P_3
B_4	Red value for P_4
B_5	Green value for P_5
•	•
•	•
•	•
B_{m-5}	Red value for P_{n-5}
B_{m-4}	Green value for P_{n-4}
B_{m-3}	Red value for P_{n-3}
B_{m-2}	Green value for P_{n-2}
B_{m-1}	Red value for P_{n-1}
B_m	Green value for P_n

With the camera set for Bayer GB 8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

9.3.3 Bayer BG 8 Format (Equivalent to DCAM Raw 8)

When a color camera is set for the Bayer BG 8 pixel data format, it outputs 8 bits of data per pixel and the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 8 bits of red data. For each pixel covered with a green lens, you get 8 bits of green data. And for each pixel covered with a blue lens, you get 8 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "BG" in the name Bayer BG 8 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel one will be blue, pixel two will be green, pixel three will be blue, pixel four will be green, etc. For odd lines in the images, pixel one will be green, pixel two will be red, pixel three will be green, pixel four will be red, etc.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer BG 8 output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

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

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Blue value for P_0
B_1	Green value for P_1
B_2	Blue value for P_2
B_3	Green value for P_3
B_4	Blue value for P_4
B_5	Green value for P_5
²	•
²	•
²	•
B_{m-5}	Blue value for P_{n-5}
B_{m-4}	Green value for P_{n-4}
B_{m-3}	Blue value for P_{n-3}
B_{m-2}	Green value for P_{n-2}
B_{m-1}	Blue value for P_{n-1}
B_m	Green value for P_n

Odd Lines	
Byte	**Data**
B_0	Green value for P_0
B_1	Red value for P_1
B_2	Green value for P_2
B_3	Red value for P_3
B_4	Green value for P_4
B_5	Red value for P_5
²	•
²	•
²	•
B_{m-5}	Green value for P_{n-5}
B_{m-4}	Red value for P_{n-4}
B_{m-3}	Green value for P_{n-3}
B_{m-2}	Red value for P_{n-2}
B_{m-1}	Green value for P_{n-1}
B_m	Red value for P_n

With the camera set for Bayer BG 8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

9.3.4 Bayer GB 16 Format (Equivalent to DCAM Raw 16)

When a color camera is set for the Bayer GB 16 pixel data format, it outputs 16 bits of data per pixel with 12 bits effective. The 12 bits of effective pixel data fill from the least significant bit. The four unused most significant bits are filled with zeros.

With the Bayer GB 16 the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 12 effective bits of red data. For each pixel covered with a green lens, you get 12 effective bits of green data. And for each pixel covered with a blue lens, you get 12 effective bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "GB" in the name Bayer GB 16 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel zero will be green, pixel one will be blue, pixel two will be green, pixel three will be blue, etc. For odd lines in the images, pixel zero will be red, pixel one will be green, pixel two will be red, pixel three will be green, etc.

For more information about the Bayer filter, see Section 9.3.1 on page 115.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer GB 16 output. Note that the data is placed in the image buffer in **little endian format**.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

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

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Low byte of green value for P_0
B_1	High byte of green value for P_0
B_2	Low byte of blue value for P_1
B_3	High byte of blue value for P_1
B_4	Low byte of green value for P_2
B_5	High byte of green value for P_2
B_6	Low byte of blue value for P_3
B_7	High byte of blue value for P_3
•	•
•	•
•	•
B_{m-7}	Low byte of green value for P_{n-3}
B_{m-6}	High byte of green value for P_{n-3}

Odd Lines	
Byte	**Data**
B_0	Low byte of red value for P_0
B_1	High byte of red value for P_0
B_2	Low byte of green value for P_1
B_3	High byte of green value for P_1
B_4	Low byte of red value for P_2
B_5	High byte of red value for P_2
B_6	Low byte of green value for P_3
B_7	High byte of green value for P_3
•	•
•	•
•	•
B_{m-7}	Low byte of red value for P_{n-3}
B_{m-6}	High byte of red value for P_{n-3}

B_{m-5}	Low byte of blue value for P_{n-2}
B_{m-4}	High byte of blue value for P_{n-2}
B_{m-3}	Low byte of green value for P_{n-1}
B_{m-2}	High byte of green value for P_{n-1}
B_{m-1}	Low byte of blue value for P_n
B_m	High byte of blue value for P_n

B_{m-5}	Low byte of green value for P_{n-2}
B_{m-4}	High byte of green value for P_{n-2}
B_{m-3}	Low byte of red value for P_{n-1}
B_{m-2}	High byte of red value for P_{n-1}
B_{m-1}	Low byte of green value for P_n
B_m	High byte of green value for P_n

When the camera is set for Bayer GB 16, the pixel data output is 16 bit data of the “unsigned short (little endian)” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below. Note that for 16 bit data, you might expect a value range from 0x0000 to 0xFFFF. However, with the camera set for Bayer GB 16 only 12 bits of the 16 bits transmitted are effective. Therefore, the highest data value you will see is 0x0FFF indicating a signal level of 4095.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

Note

When a camera that is set for Bayer GB 16 has only 12 bits effective, the leader of transmitted frames will indicate Bayer GB 12 as the pixel format.

9.3.5 Bayer BG 16 Format (Equivalent to DCAM Raw 16)

When a color camera is set for the Bayer BG 16 pixel data format, it outputs 16 bits of data per pixel with 12 bits effective. The 12 bits of effective pixel data fill from the least significant bit. The four unused most significant bits are filled with zeros.

With the Bayer BG 16 the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 12 effective bits of red data. For each pixel covered with a green lens, you get 12 effective bits of green data. And for each pixel covered with a blue lens, you get 12 effective bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "BG" in the name Bayer BG 16 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel one will be blue, pixel two will be green, pixel three will be blue, pixel four will be green, etc. For odd lines in the images, pixel one will be green, pixel two will be red, pixel three will be green, pixel four will be red, etc.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer BG 16 output. Note that the data is placed in the image buffer in **little endian format**.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

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

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Low byte of blue value for P_0
B_1	High byte of blue value for P_0
B_2	Low byte of green value for P_1
B_3	High byte of green value for P_1
B_4	Low byte of blue value for P_2
B_5	High byte of blue value for P_2
B_6	Low byte of green value for P_3
B_7	High byte of green value for P_3
•	•
•	•
•	•
B_{m-7}	Low byte of blue value for P_{n-3}
B_{m-6}	High byte of blue value for P_{n-3}
B_{m-5}	Low byte of green value for P_{n-2}
B_{m-4}	High byte of green value for P_{n-2}

Odd Lines	
Byte	**Data**
B_0	Low byte of green value for P_0
B_1	High byte of green value for P_0
B_2	Low byte of red value for P_1
B_3	High byte of red value for P_1
B_4	Low byte of green value for P_2
B_5	High byte of green value for P_2
B_6	Low byte of red value for P_3
B_7	High byte of red value for P_3
•	•
•	•
•	•
B_{m-7}	Low byte of green value for P_{n-3}
B_{m-6}	High byte of green value for P_{n-3}
B_{m-5}	Low byte of red value for P_{n-2}
B_{m-4}	High byte of red value for P_{n-2}

B_{m-3}	Low byte of blue value for P_{n-1}
B_{m-2}	High byte of blue value for P_{n-1}
B_{m-1}	Low byte of green value for P_n
B_m	High byte of green value for P_n

B_{m-3}	Low byte of green value for P_{n-1}
B_{m-2}	High byte of green value for P_{n-1}
B_{m-1}	Low byte of red value for P_n
B_m	High byte of red value for P_n

When the camera is set for Bayer BG 16, the pixel data output is 16 bit data of the “unsigned short (little endian)” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below. Note that for 16 bit data, you might expect a value range from 0x0000 to 0xFFFF. However, with the camera set for Bayer BG 16 only 12 bits of the 16 bits transmitted are effective. Therefore, the highest data value you will see is 0x0FFF indicating a signal level of 4095.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

Note

When a camera that is set for Bayer BG 16 has only 12 bits effective, the leader of transmitted frames will indicate Bayer BG 12 as the pixel format.

9.3.6 Bayer GB 12 Packed Format

When a color camera is set for the Bayer GB 12 Packed pixel dataformat, it outputs 12 bits of data per pixel. Every three bytes transmitted by the camera contain data for two pixels.

With the Bayer GB 12 Packed coding, the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens in the sensor's Bayer filter, you get 12 bits of red data. For each pixel covered with a green lens in the filter, you get 12 bits of green data. And for each pixel covered with a blue lens in the filter, you get 12 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

For more information about the Bayer filter, see Section 9.3.1 on page 115.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer GB 12 Packed output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

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

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

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

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

When a color camera is set for Bayer GB 12 Packed, the pixel data output is 12 bit data of the “unsigned” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

9.3.7 Bayer BG 12 Packed Format

When a color camera is set for the Bayer BG 12 Packed pixel dataformat, it outputs 12 bits of data per pixel. Every three bytes transmitted by the camera contain data for two pixels.

With the Bayer BG 12 Packed coding, the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens in the sensor's Bayer filter, you get 12 bits of red data. For each pixel covered with a green lens in the filter, you get 12 bits of green data. And for each pixel covered with a blue lens in the filter, you get 12 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer BG12 Packed output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

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

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

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

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

When a color camera is set for Bayer BG 12 Packed, the pixel data output is 12 bit data of the “unsigned” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

9.3.8 YUV 4:2:2 Packed Format (Equivalent to DCAM YUV 4:2:2)

When a color camera is set for the YUV 422 Packed pixel data format, each pixel in the captured image goes through a two step conversion process as it exits the sensor and passes through the camera's electronics. This process yields Y, U, and V color information for each pixel.

In the first step of the process, an interpolation algorithm is performed to get full RGB data for each pixel. This is required because color cameras use a Bayer filter on the sensor and each individual pixel gathers information for only one color.

For more information on the Bayer filter, see Section 9.3.1 on page 115.

The second step of the process is to convert the RGB information to the YUV color model. The conversion algorithm uses the following formulas:

$Y = 0.30\,R + 0.59\,G + 0.11\,B$

$U = -0.17\,R - 0.33\,G + 0.50\,B$

$V = 0.50\,R - 0.41\,G - 0.09\,B$

Once the conversion to a YUV color model is complete, the pixel data is transmitted to the host PC.

Note

The values for U and for V normally range from -128 to +127. Because the camera transfers U values and V values with unsigned integers, 128 is added to each U value and to each V value before the values are transferred from the camera. This process allows the values to be transferred on a scale that ranges from 0 to 255.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for YUV 4:2:2 Packed output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	**Data**
B_0	U value for P_0
B_1	Y value for P_0
B_2	V Value for P_0
B_3	Y value for P_1
B_4	U value for P_2
B_5	Y value for P_2
B_6	V Value for P_2
B_7	Y value for P_3
B_8	U value for P_4
B_9	Y value for P_4
B_{10}	V Value for P_4
B_{11}	Y value for P_5
•	•
•	•
•	•
B_{m-7}	U value for P_{n-3}
B_{m-6}	Y value for P_{n-3}
B_{m-5}	V Value for P_{n-3}
B_{m-4}	Y value for P_{n-2}
B_{m-3}	U value for P_{n-1}
B_{m-2}	Y value for P_{n-1}
B_{m-1}	V Value for P_{n-1}
B_m	Y value for P_n

When the camera is set for YUV 4:2:2 Packed output, the pixel data output for the Y component is 8 bit data of the “unsigned char” type. The range of data values for the Y component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

The pixel data output for the U component or the V component is 8 bit data of the “straight binary” type. The range of data values for a U or a V component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	127
0xFE	126
•	•
•	•
•	•
0x81	1
0x80	0
0x7F	-1
•	•
•	•
•	•
0x01	-127
0x00	-128

The signal level of a U component or a V component can range from -128 to +127 (decimal). Notice that the data values have been arranged to represent the full signal level range.

Note

The interpolation and conversion algorithms are applied to the averaged pixel values when the averaging feature is used.

9.3.9 YUV 4:2:2 (YUYV) Packed Format

On color cameras, the YUV 4:2:2 (YUYV) packed pixel data format is similar to the YUV 4:2:2 pixel format described in the previous section. The only difference is the order of the bytes transmitted to the host PC. With the YUV 4:2:2 format, the bytes are ordered as specified in the DCAM standard issued by the 1394 Trade Association. With the YUV 4:2:2 (YUYV) format, the bytes are ordered to emulate the ordering normally associated with analog frame grabbers and Windows® frame buffers.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for YUV 4:2:2 (YUYV) output.

With this format, the Y component is transmitted for each pixel, but the U and V components are only transmitted for every second pixel.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	**Data**
B_0	Y value for P_0
B_1	U value for P_0
B_2	Y value for P_1
B_3	V value for P_0
B_4	Y value for P_2
B_5	U value for P_2
B_6	Y value for P_3
B_7	V value for P_2
B_8	Y value for P_4
B_9	U value for P_4
B_{10}	Y value for P_5
B_{11}	V value for P_4
•	•
•	•
•	•
B_{m-7}	Y value for P_{n-3}
B_{m-6}	U value for P_{n-3}
B_{m-5}	Y value for P_{n-2}
B_{m-4}	V value for P_{n-3}
B_{m-3}	Y value for P_{n-1}
B_{m-2}	U value for P_{n-1}
B_{m-1}	Y value for P_n
B_m	V value for P_{n-1}

When a color camera is set for YUV 4:2:2 (YUYV) output, the pixel data output for the Y component is 8 bit data of the “unsigned char” type. The range of data values for the Y component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

The pixel data output for the U component or the V component is 8 bit data of the “straight binary” type. The range of data values for a U or a V component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	127
0xFE	126
•	•
•	•
•	•
0x81	1
0x80	0
0x7F	-1
•	•
•	•
•	•
0x01	-127
0x00	-128

The signal level of a U component or a V component can range from -128 to +127 (decimal). Notice that the data values have been arranged to represent the full signal level range.

Note

The interpolation and conversion algorithms are applied to the averaged pixel values when the averaging feature is used.

9.3.10 Mono 8 Format (Equivalent to DCAM Mono 8)

When a color camera is set for the Mono 8 pixel data format, the pixel values in each captured image are first interpolated and converted to the YUV color model as described for the YUV 4:2:2 Packed format. The camera then transmits the 8 bit Y value for each pixel to the host PC. In the YUV color model, the Y component for each pixel represents a brightness value. This brightness value can be considered as equivalent to the value that would be sent from a pixel in a monochrome camera. So in essence, when a color camera is set for Mono 8, it outputs an 8 bit monochrome image. (This type of output is sometimes referred to as "Y Mono 8".)

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when a color camera is set for Mono 8 output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	**Data**
B_0	Y value for P_0
B_1	Y value for P_1
B_2	Y value for P_2
B_3	Y value for P_3
B_4	Y value for P_4
B_5	Y value for P_5
B_6	Y value for P_6
B_7	Y value for P_7
•	•
•	•
•	•
B_{m-3}	Y value for P_{n-3}
B_{m-2}	Y value for P_{n-2}
B_{m-1}	Y value for P_{n-1}
B_m	Y value for P_n

With the camera set for Mono 8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

Note

The interpolation and conversion algorithms are applied to the averaged pixel values when the averaging feature is used.

Pixel Transmission Sequence

For each captured image, pixel data is transmitted from the camera in the following sequence:

Row $_0$ Col $_0$,	Row $_0$ Col $_1$,	Row $_0$ Col $_2$	.. ..	Row $_0$ Col $_{m-2}$,	Row $_0$ Col $_{m-1}$,	Row $_0$ Col $_m$
Row $_1$ Col $_0$,	Row $_1$ Col $_1$,	Row $_1$ Col $_2$	.. ..	Row $_1$ Col $_{m-2}$,	Row $_1$ Col $_{m-1}$,	Row $_1$ Col $_m$
Row $_2$ Col $_0$,	Row $_2$ Col $_1$,	Row $_2$ Col $_2$	.. ..	Row $_2$ Col $_{m-2}$,	Row $_2$ Col $_{m-1}$,	Row $_2$ Col $_m$
:	:	:		:	:	:
:	:	:		:	:	:
Row $_{n-2}$ Col $_0$,	Row $_{n-2}$ Col $_1$,	Row $_{n-2}$ Col $_2$	.. ..	Row $_{n-2}$ Col $_{m-2}$,	Row $_{n-2}$ Col $_{m-1}$,	Row $_{n-2}$ Col $_m$
Row $_{n-1}$ Col $_0$,	Row $_{n-1}$ Col $_1$,	Row $_{n-1}$ Col $_2$	.. ..	Row $_{n-1}$ Col $_{m-2}$,	Row $_{n-1}$ Col $_{m-1}$,	Row $_{n-1}$ Col $_m$
Row $_n$ Col $_0$,	Row $_n$ Col $_1$,	Row $_n$ Col $_2$	.. ..	Row $_n$ Col $_{m-2}$,	Row $_n$ Col $_{m-1}$,	Row $_n$ Col $_m$

Where Row $_0$ Col $_0$ is the upper left corner of the sensor

The columns are numbered 0 through m from the left side to the right side of the sensor

The rows are numbered 0 through n from the top to the bottom of the sensor

The sequence assumes that the camera is set for full resolution.

10 I/O Control

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

For more detailed information about the physical and electrical characteristics of the input and output lines, see Section 7.7 on page 70.

10.1 Configuring Input Lines

10.1.1 Assigning an Input Line to Receive a Hardware Trigger Signal

You can assign one of the camera’s input lines to receive a external hardware trigger (ExTrig) signal. The incoming ExTrig signal can then be used to control image acquisition.

Section 8.3.2 on page 87 explains how to configure the camera to react to a hardware trigger signal and how to assign an input line to receive the hardware trigger signal.

Note

By default, physical input line 1 is assigned to receive the ExTrig signal. You can assign only one line to receive the ExTrig input signal.

10.1.2 Using an Unassigned Input Line to Receive a User Input Signal

You can use an unassigned input line to receive your own, user-generated input signal. The electrical characteristics of your input signal must meet the requirements shown in the Physical Interface section of this manual.

You can use the Line Status or Line Status All parameters to monitor the state of the input line that is receiving the user-defined signal.

Note

The line assigned to receive the ExTrig input signal can’t be used to receive a user-designed input signal.

For more information about using the Line Status and Line Status All parameters, see Section 10.3.1 on page 147 and Section 10.3.2 on page 147.

10.2 Configuring Output Lines

10.2.1 Assigning a Camera Output Signal to a Physical Output Line

You can use the camera's output signal assignment capability to assign one of the camera's standard output signals as the source signal for a physical output line. The camera has a variety of standard output signals available including:

- Exposure Active
- Trigger Ready
- Timer 1, Timer 2, Timer 3, Timer 4

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

To assign an output signal to an output line or to designate the line as user settable:

- Use the Line Selector to select Output Line 1, Output Line 2, Output Line 3, or Output Line 4.
- Set the value of the Line Source Parameter to one of the available output signals or to user settable. This will set the source signal for the selected line.

Note

By default, the Exposure Active signal is assigned to Output Line 1 and the Trigger Ready Signal is assigned to Output Line 2.

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

```
Camera.LineSelector.SetValue( LineSelector_Out1 );
Camera.LineSource.SetValue( LineSource_ExposureActive );
Camera.LineSelector.SetValue( LineSelector_Out2 );
Camera.LineSource.SetValue( LineSource_TriggerReady );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

For more information about setting the state of user settable output signals, see Section 10.2.2 on page 140.

For more information about working with the timer output signals, see Section 10.2.4 on page 142

For more information about the exposure active signal, see Section 8.7 on page 98.

For more information about the trigger ready signal, see Section 8.6 on page 96.

10.2.2 Setting the State of User Settable Output Lines

As mentioned in the previous section, you can designate one or more of the user output lines as "user settable". Once you have designated an output line as user settable, you can use camera parameters to set the state of the line.

Setting the State of a Single User Settable Output Line

To set the state of a single user settable output line:

- Use the User Output Selector to select the output line you want to set. For example, if you have designated output line 3 as user settable, you would select user settable output 3.
- Set the value of the User Output Value parameter to true (high) or false (low). This will set the state of the selected line.

You can set the Output Selector and the User Output Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to designate output line 3 as user settable and setting the state of the output line:

```
Camera.LineSelector.SetValue( LineSelector_Out3 );
Camera.LineSource.SetValue( LineSource_UserOutput );
Camera.UserOutputSelector.SetValue( UserOutputSelector_UserOutput3 );
Camera.UserOutputValue.SetValue( true );
bool currentUserOutput3State = Camera.UserOutputValue.GetValue( );
```

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

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

Setting the State of Multiple User Settable Output Lines

The User Output Value All parameter is a 32 bit value. As shown in Figure 40, the lowest four bits of the parameter value will set the state of the user settable outputs. If a bit is 0, it will set the state of the associated output to low. If a bit is high, it will set the state of the associated port to high.

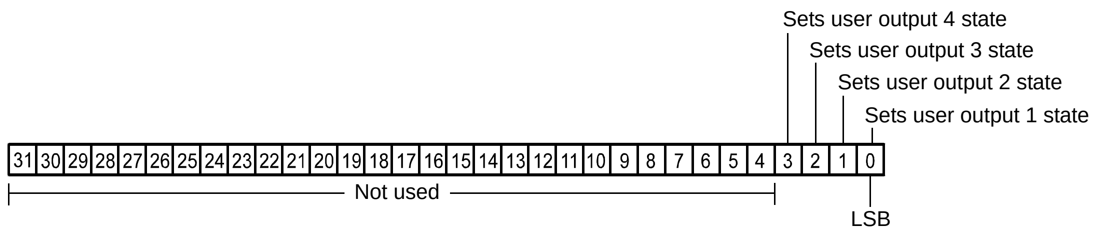


Fig. 40: User Output Value All Parameter Bits

To set the state of multiple user settable output lines:

- Use the User Output Value All parameter to set the state of multiple user settable outputs.

You can set the User Output Value All parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter:

```
Camera.UserOutputValueAll.SetValue( 0x3 );
int64_t currentOutputState = Camera.UserOutputValueAll.GetValue( );
```

Note

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

10.2.3 Setting an Output Line for Invert

You can set each individual output line to invert or not to invert the outgoing signal. To set the invert function on an output line:

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

10.2.4 Working with Timers

The camera has four timer output signals available: Timer 1, Timer 2, Timer 3, and Timer 4. As shown in Figure 41, each timer works as follows:

- A trigger source event occurs that starts the timer.
- A delay period begins to expire.
- When the delay expires, the timer signal goes high and a duration period begins to expire.
- When the duration period expires, the timer signal goes low.

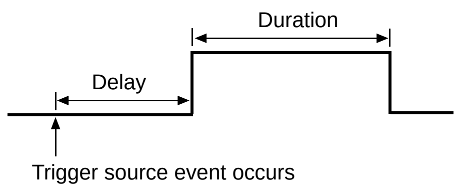


Fig. 41: Timer Signal

Currently, the only trigger source event available to start the timer is "exposure active". In other words, you can use exposure start to trigger the start of a timer.

Timer 1 can only be assigned to output line 1. Timer 2 can only be assigned to output line 2. Timer 3 can only be assigned to output line 3. Timer 4 can only be assigned to output line 4.

If you require the timer signal to be high when the timer is triggered and to go low when the delay expires, simply set the output line to invert.

10.2.4.1 Setting the Trigger Source for a Timer

To set the trigger source for a timer:

- Use the Timer Selector to select timer 1 or timer 2.
- Set the value of the Timer Trigger Source parameter to exposure active. This will set the selected timer to use the start of exposure to begin the timer.

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

10.2.4.2 Setting a Timer Delay Time

There are two ways to set the delay time for a timer: by setting "raw" values or by setting an "absolute value". You can use whichever method you prefer to set the delay time.

Setting the Delay with Raw Values

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

Delay Time = (Timer Delay Raw Parameter Value) x (Timer Delay Time Base)

By default, the Timer Delay Time Base is fixed at 1 µs. Typically, the delay time is adjusted by setting the Timer Delay Raw parameter value.

The Timer Delay Raw parameter value can range from 0 to 4095. So if the value is set to 100, for example, the timer delay will be 100 x 1 µs or 100 µs.

To set the delay for a timer:

- Use the Timer Selector to select a timer.
- Set the value of the Timer Delay Raw parameter.

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDelayRaw.SetValue( 100 );
```

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

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

Changing the Delay Time Base

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

The Timer Delay Time Base Abs parameter value sets the delay time base in µs. The default is 1 µs and it can be changed in 1 µs increments.

Note that there is only one timer delay time base and it is used by all four of the available timers.

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

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

Setting the Delay with an Absolute Value

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

To set the delay for a timer using an absolute value:

- Use the Timer Selector to select a timer.
- Set the value of the Timer Delay Abs parameter.

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDelayAbs.SetValue( 100 );
```

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

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

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

10.2.4.3 Setting a Timer Duration Time

There are two ways to set the duration time for a timer: by setting "raw" values or by setting an "absolute value". You can use whichever method you prefer to set the duration time.

Setting the Duration with Raw Values

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

Duration Time = (Timer Duration Raw Parameter Value) x (Timer Duration Time Base)

By default, the Timer Duration Time Base is fixed at 1 µs. Typically, the duration time is adjusted by setting only the Timer Duration Raw parameter value.

The Timer Duration Raw parameter value can range from 1 to 4095. So if the value is set to 100, for example, the timer duration will be 100 x 1 µs or 100 µs.

To set the duration for a timer:

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

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDurationRaw.SetValue( 100 );
```

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

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

Changing the Duration Time Base

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

The Timer Duration Time Base Abs parameter value sets the duration time base in µs. The default is 1 µs and it can be changed in 1 µs increments.

Note that there is only one timer duration time base and it is used by all four of the available timers.

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

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

Setting the Duration with an Absolute Value

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

To set the duration for a timer using an absolute value:

- Use the Timer Selector to select a timer.
- Set the value of the Timer Duration Abs parameter.

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDurationAbs.SetValue( 100 );
```

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

If you read the current value of the Timer Duration Abs parameter, the value will indicate the product of the Timer Duration Raw parameter and the Timer Duration Time Base. In other words, the Timer Duration Abs parameter will indicate the current duration time setting.

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

10.3 Checking the State of the I/O Lines

10.3.1 Checking the State of a Single Output Line

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

- Use the Line Selector parameter to select an output line.
- Read the value of the Line Status parameter to determine the current state of the selected line. A value of true means the line's state is currently high and a value of false means the line's state is currently low.

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

```
// Select output line 2 and read the state
Camera.LineSelector.SetValue( LineSelector_Out2 );
bool outputLine2State = Camera.LineStatus.GetValue( );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

10.3.2 Checking the State of All Lines

You can determine the current state of all input and output lines with a single operation. To check the state of all lines:

- Read the value of the Line Status All parameter.

You can read the Line Status All parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to read the parameter value:

```
int64_t lineState = Camera.LineStatusAll.GetValue( );
```

The Line Status All parameter is a 32 bit value. As shown in Figure 42, certain bits in the value are associated with each line and the bits will indicate the state of the lines. If a bit is 0, it indicates that the state of the associated line is currently low. If a bit is 1, it indicates that the state of the associated line is current high.

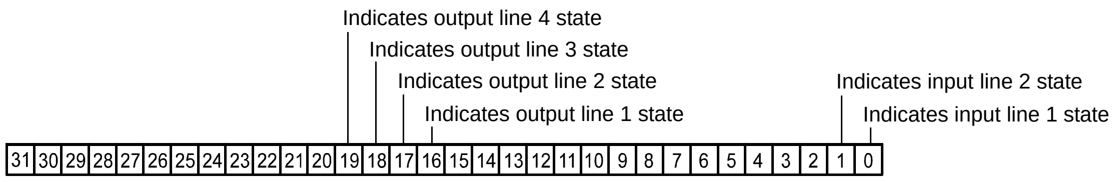

Indicates output line 4 state
Indicates output line 3 state
Indicates output line 2 state
Indicates output line 1 state
Indicates input line 2 state
Indicates input line 1 state
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 42: Line Status All Parameter Bits

11 Features

This section 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.

11.1 Gain

The camera's gain is adjustable. As shown in Figure 43, increasing the gain increases the slope of the response curve for the camera. This results in a higher gray value output from the camera for a given amount of output from the imaging sensor. Decreasing the gain decreases the slope of the response curve and results in a lower gray value for a given amount of sensor output.

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

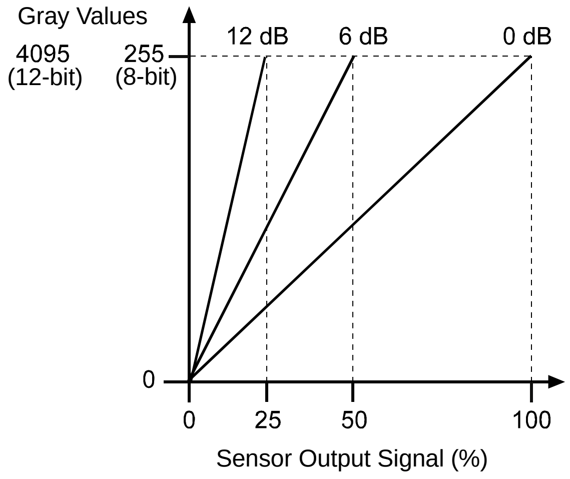


Fig. 43: Gain in dB

As mentioned in the "Functional Description" section of this manual, for readout purposes, the sensor used in the camera is divided into two halves. As a result of this design, there are three gain adjustments available: Gain Raw All, Gain Raw Tap 1, and Gain Raw Tap 2.

Gain Raw All is a global adjustment, i.e., its setting affects both halves of the sensor.

Gain Raw Tap 1 sets an additional amount of gain for the right half of the sensor. The total gain for the right half of the sensor will be the sum of the Gain Raw All value plus the Gain Raw Tap 1 value.

Gain Raw Tap 2 sets an additional amount of gain for the left half of the sensor. The total gain for the left half of the sensor will be the sum of the Gain Raw All value plus the Gain Raw Tap 2 value.

For each camera model, the minimum and maximum allowed Gain Raw and Gain Total settings are shown in the tables below:

		Gain Raw All		Gain Raw Tap 1		Gain Raw Tap 2	
Camera Model	**Min Setting**	**Max Setting (8 bit depth)**	**Max Setting (16 bit depth)**	**Max Setting (8 bit depth)**	**Max Setting (16 bit depth)**	**Max Setting (8 bit depth)**	**Max Setting (16 bit depth)**
piA640-210	0	500	400	500	400	500	400
piA1000-48	0	500	400	500	400	500	400
piA1600-35	0	500	400	500	400	500	400
piA1900-32	0	500	400	500	400	500	400
piA2400-12	0	500	400	500	400	500	400
piA2400-17	0	500	400	500	400	500	400

Table 15: Minimum and Maximum Allowed Gain Raw Settings

		Gain Raw All + Gain Raw Tap 1		Gain Raw All + Gain Raw Tap 2	
Camera Model	**Min Setting**	**Max Setting (8 bit depth)**	**Max Setting (16 bit depth)**	**Max Setting (8 bit depth)**	**Max Setting (16 bit depth)**
piA640-210	0	500	400	500	400
piA1000-48	0	500	400	500	400
piA1600-35	0	500	400	500	400
piA1900-32	0	500	400	500	400
piA2400-12	0	500	400	500	400
piA2400-17	0	500	400	500	400

Table 16: Minimum and Maximum Allowed Total Gain Settings

If, for example, the piA640-210 gm/gc camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed):

- The Gain Raw All value can be set in a range from 0 to 500.
- The Gain Raw Tap 1 value can be set in a range from 0 to 500.
- The Gain Raw Tap 2 value can be set in a range from 0 to 500.
- The sum of the Gain Raw All setting plus the Gain Raw Tap 1 setting must be between 0 and 500 (inclusive).
- The sum of the Gain Raw All setting plus the Gain Raw Tap 2 setting must be between 0 and 500 (inclusive).

If, for example, the piA640-210 gm/gc the camera is set for a pixel data format that yields an effective pixel depth of 12 bits per pixel (Mono 16, Mono 12 Packed):

- The Gain Raw All value can be set in a range from 0 to 400.
- The Gain Raw Tap 1 value can be set in a range from 0 to 400.
- The Gain Raw Tap 2 value can be set in a range from 0 to 400.
- The sum of the Gain Raw All setting plus the Gain Raw Tap 1 setting must be between 0 and 400 (inclusive).
- The sum of the Gain Raw All setting plus the Gain Raw Tap 2 setting must be between 0 and 400 (inclusive).

For normal operation, we recommend that you set the value of Gain Raw Tap 1 and Gain Raw Tap 2 to zero and that you simply use Gain Raw All to set the gain. Typically, the tap gains are only used if you want to adjust the gain balance between the left half and the right half of the sensor.

If you know the current settings for Gain Raw All, Gain Raw Tap 1, and Gain Raw Tap 2, you can use the formulas below to calculate the dB of gain that will result from the settings.

Gain on the Right Sensor Half = (0.0359 x Gain Raw All Setting) + (0.0359 x Gain Raw Tap 1 Setting)

Gain on the Left Sensor Half = (0.0359 x Gain Raw All Setting) + (0.0359 x Gain Raw Tap 2 Setting)

For example, assume that you have set the Gain Raw All to 450, the Gain Raw Tap 1 to 0, and the Gain Raw Tap 2 to 0. Then:

Gain on the Right Sensor Half = (0.0359 x 450) + (0.0359 x 0)

Gain on the Right Sensor Half = 16.2 dB

Gain on the Left Sensor Half = (0.0359 x 450) + (0.0359 x 0)

Gain on the Left Sensor Half = 16.2 dB

Setting the Gain

ⓘ	**Note** Gain can not only be manually set (see below), but can also be automatically adjusted. The Gain Auto function is the "automatic" counterpart of the gain feature and carries out a Gain Raw All adjustment automatically. For more information about auto fuctions, see Section 11.12.1 on page 180. For more information about the Gain Auto function, see Section 11.12.2 on page 187.

To set the Gain Raw All parameter value:

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

To set the Gain Raw Tap 1 parameter value:

- Set the Gain Selector to Tap 1.
- Set the Gain Raw parameter to your desired value.

To set the Gain Raw Tap 2 parameter value:

- Set the Gain Selector to Tap 2.
- Set the Gain Raw parameter to your desired value.

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

```
// Set Gain Raw All
Camera.GainSelector.SetValue( GainSelector_All );
Camera.GainRaw.SetValue( 100 );

//Set Gain Raw Tap 1
Camera.GainSelector.SetValue( GainSelector_Tap1 );
Camera.GainRaw.SetValue( 0 );

//Set Gain Raw Tap 2
Camera.GainSelector.SetValue( GainSelector_Tap2 );
Camera.GainRaw.SetValue( 0 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.2 Black Level

Adjusting the camera's black level will result in an offset to the pixel values output from the camera.

As mentioned in the "Functional Description" section of this manual, for readout purposes, the sensor used in the camera is divided into two halves. As a result of this design, there are three black level adjustments available: Black Level Raw All, Black Level Raw Tap 1, and Black Level Raw Tap 2.

Black Level Raw All is a global adjustment, i.e., its setting affects both halves of the sensor. The Black Level Raw All value can be set in a range from 0 to 1023.

Black Level Raw Tap 1 sets an additional amount of black level adjustment for the right half of the sensor. The Black Level Raw Tap 1 value can be set in a range from 0 to 1023. The total black level for the right half of the sensor will be the sum of the Black Level Raw All value plus the Black Level Raw Tap 1 value.

Black Level Raw Tap 2 sets an additional amount of black level adjustment for the left half of the sensor. The Black Level Raw Tap 2 value can be set in a range from 0 to 1023. The total black level for the left half of the sensor will be the sum of the Black Level Raw All value plus the Black Level Raw Tap 2 value.

If the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed), an increase of 64 in a black level setting will result in a positive offset of 1 in the pixel values output from the camera. And a decrease of 64 in a black level setting result in a negative offset of 1 in the pixel values output from the camera.

If the camera is set for a pixel data format that yields an effective pixel depth of 12 bits per pixel (Mono 16, Mono 12 Packed), an increase of 4 in a black level setting will result in a positive offset of 1 in the pixel values output from the camera. A decrease of 4 in a black level setting will result in a negative offset of 1 in the pixel values output from the camera.

For normal operation, we recommend that you set the value of Black Level Raw Tap 1 and Black Level Raw Tap 2 to zero and that you simply use Black Level Raw All to set the black level. Typically, the tap black level settings are only used if you want to adjust the black level balance between the left half and the right half of the sensor.

Note

The sum of the Black Level Raw All setting plus the Black Level Raw Tap 1 setting must be less than or equal to 1023.

The sum of the Black Level Raw All setting plus the Black Level Raw Tap 2 setting must also be less than or equal to 1023.

Setting the Black Level

To set the Black Level Raw All value:

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

To set the Black Level Raw Tap 1 value:

- Set the Black Level Selector to Tap 1.
- Set the Black Level Raw parameter to your desired value.

To set the Black Level Raw Tap 2 value:

- Set the Black Level Selector to Tap 2.
- Set the Black Level Raw parameter to your desired value.

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

```
// Set Black Level Raw All
Camera.BlackLevelSelector.SetValue ( BlackLevelSelector_All );
Camera.BlackLevelRaw.SetValue( 64 );

//Set Black Level Raw Tap 1
Camera.BlackLevelSelector.SetValue ( BlackLevelSelector_Tap1 );
Camera.BlackLevelRaw.SetValue( 0 );

//Set Black Level Raw Tap 2
Camera.BlackLevelSelector.SetValue ( BlackLevelSelector_Tap2 );
Camera.BlackLevelRaw.SetValue( 0 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.3 White Balance (on Color Models)

White balance capability has been implemented on color models of the camera. White balancing can be used to adjust the color balance of the images transmitted from the camera.

Setting the White Balance

Note

White balance can not only be manually set (see below), but can also be automatically adjusted. The Balance White Auto function is the "automatic" counterpart of the white balance feature and adjusts the white balance automatically.

For more information about auto fuctions, see Section 11.12.1 on page 180.

For more information about the Balance White Auto function, see Section 11.12.5 on page 192.

With the white balancing scheme used on these cameras, the red intensity, green intensity, and blue intensity can each be adjusted. For each color, a Balance Ratio parameter is used to set the intensity of the color. If the Balance Ratio parameter for a color is set to a value of 1, the intensity of the color will be unaffected by the white balance mechanism. If the ratio is set to a value lower than 1, the intensity of the color will be reduced. If the ratio is set to a value greater than 1, the intensity of the color will be increased. The increase or decrease in intensity is proportional. For example, if the balance ratio for a color is set to 1.2, the intensity of that color will be increased by 20%.

The balance ratio value can range from 0.00 to 3.98. But you should be aware that if you set the balance ratio for a color to a value lower than 1, this will not only decrease the intensity of that color relative to the other two colors, but will also decrease the maximum intensity that the color can achieve. For this reason, we don’t normally recommend setting a balance ratio less than 1 unless you want to correct for the strong predominance of one color.

To set the Balance Ratio parameter for a color:

- Set the Balance Ratio Selector to red, green, or blue.
- Set the Balance Ratio Abs parameter to the desired value for the selected color.

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

```
Camera.BalanceRatioSelector.SetValue( BalanceRatioSelector_Green );
Camera.BalanceRatioAbs.SetValue( 1.20 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

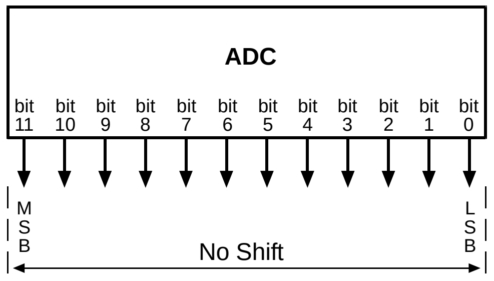
11.4 Digital Shift

The digital shift feature lets you change the group of bits that is output from each ADC in the camera. Using the digital shift feature will effectively multiply the output of the camera by 2 times, 4 times, 8 times, or 16 times. The next two sections describe how the digital shift feature works when the camera is set for a 12 bit pixel format and when it is set for a 8 bit pixel format. There is also a section describing precautions that you must observe when using the digital shift feature and a section that describes enabling and setting the digital shift feature.

11.4.1 Digital Shift with 12 Bit Pixel Formats

No Shift

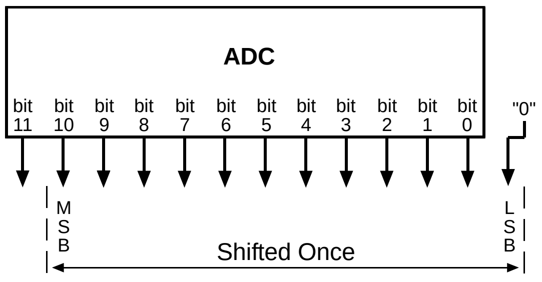
As mentioned in the Functional Description section of this manual, the camera uses 12 bit ADCs to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 12 bit effective depth, by default, the camera transmits the 12 bits that are output from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



Shift by 1

When the camera is set to shift by 1, the output from the camera will include bit 10 through bit 0 from each ADC along with a zero as an LSB.

The result of shifting once is that the output of the camera is effectively multiplied by 2. For example, assume that the camera is set for no shift, that it is viewing a uniform white target, and that under these conditions the reading for the brightest pixel is 100. If you changed the digital shift setting to shift by 1, the reading would increase to 200.

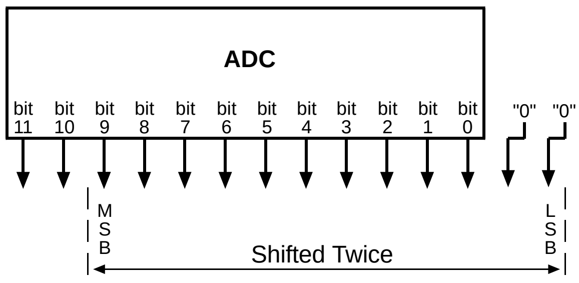


When the camera is set to shift by 1, the least significant bit output from the camera for each pixel value will be 0. This means that no odd gray values can be output and that the gray value scale will only include values of 2, 4, 6, 8, 10, and so on. This absence of some gray values is commonly referred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

If the pixel values being output by the camera's sensor are high enough to set bit 11 to 1, we recommend not using shift by 1. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 1 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 2048.

Shift by 2

When the camera is set to shift by 2, the output from the camera will include bit 9 through bit 0 from each ADC along with 2 zeros as LSBs.



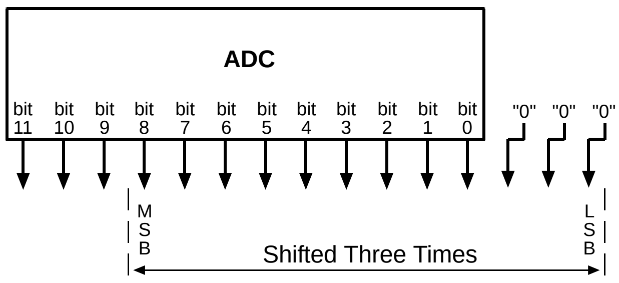
The result of shifting twice is that the output of the camera is effectively multiplied by 4.

When the camera is set to shift by 2, the 2 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 4th gray value, for example, 4, 8, 16, 20, and so on.

If the pixel values being output by the camera's sensor are high enough to set bit 10 or bit 11 to 1, we recommend not using shift by 2. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 2 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 1024.

Shift By 3

When the camera is set to shift by 3, the output from the camera will include bit 8 through bit 0 from each ADC along with 3 zeros as LSBs.



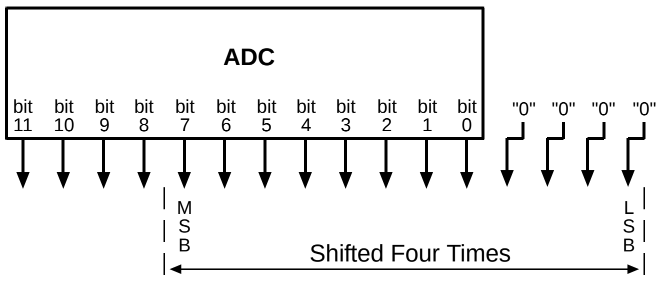
The result of shifting 3 times is that the output of the camera is effectively multiplied by 8.

When the camera is set to shift by 3, the 3 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 8th gray value, for example, 8, 16, 24, 32, and so on.

If the pixel values being output by the camera's sensor are high enough to set bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 3. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 3 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 512.

Shift By 4

When the camera is set to shift by 4, the output from the camera will include bit 7 through bit 0 from each ADC along with 4 zeros as LSBs.



The result of shifting 4 times is that the output of the camera is effectively multiplied by 16.

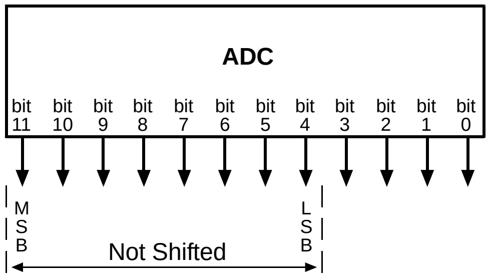
When the camera is set to shift by 4, the 4 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 16th gray value, for example, 16, 32, 48, 64, and so on.

If the pixel values being output by the camera's sensor are high enough to set bit 8, bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 4. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 4 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 256.

11.4.2 Digital Shift with 8 Bit Pixel Formats

No Shift

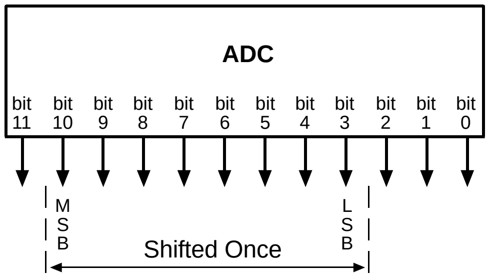
As mentioned in the Functional Description section of this manual, the camera uses 12 bit ADCs to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 8 bit effective depth, by default, the camera drops the 4 least significant bits from each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



Shift by 1

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

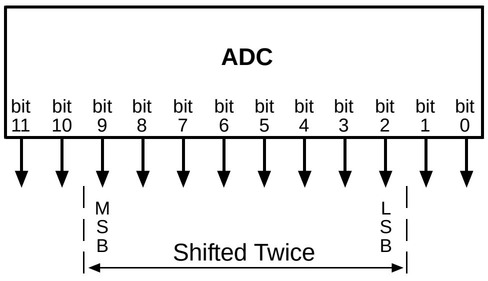
The result of shifting once is that the output of the camera is effectively multiplied by 2. For example, assume that the camera is set for no shift, that it is viewing a uniform white target, and that under these conditions the reading for the brightest pixel is 10. If you changed the digital shift setting to shift by 1, the reading would increase to 20.



If the pixel values being output by the camera's sensor are high enough to set bit 11 to 1, we recommend not using shift by 1. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 1 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 128.

Shift by 2

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

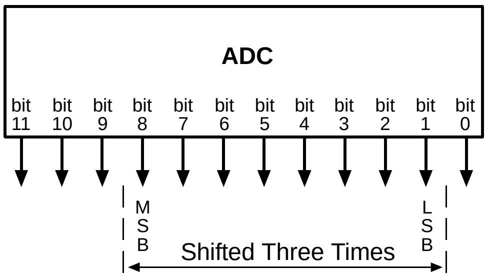


The result of shifting twice is that the output of the camera is effectively multiplied by 4.

If the pixel values being output by the camera's sensor are high enough to set bit 10 or bit 11 to 1, we recommend not using shift by 2. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 2 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 64.

Shift by 3

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

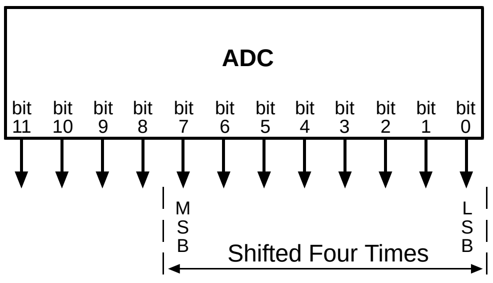


The result of shifting three times is that the output of the camera is effectively multiplied by 8.

If the pixel values being output by the camera's sensor are high enough to set bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 3. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 3 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 32.

Shift by 4

When the camera is set to shift by 4, the output from the camera will include bit 7 through bit 0 from each ADC.



The result of shifting four times is that the output of the camera is effectively multiplied by 16.

If the pixel values being output by the camera's sensor are high enough to set bit 8, bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 4. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 4 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 16.

11.4.3 Precautions When Using Digital Shift

There are several checks and precautions that you must follow before using the digital shift feature. The checks and precautions differ depending on whether the camera will be set for a 12 bit pixel format or for an 8 bit pixel format in your application.

If you will be using a 12 bit pixel format, make this check:

Use the pylon Viewer or the pylon API to set the camera for a 12 bit pixel format and **no digital shift**.

Check the output of the camera under your normal lighting conditions and note the readings for the brightest pixels.

- If any of the readings are above 2048, do not use digital shift.
- If all of the readings are below 2048, you can safely use the shift by 1 setting.
- If all of the readings are below 1024, you can safely use the shift by 1 or 2 settings.
- If all of the readings are below 512, you can safely use the shift by 1, 2, or 3 settings.
- If all of the readings are below 256, you can safely use the shift by 1, 2, 3, or 4 settings.

If you will be using an 8 bit format, make this check:

Use the pylon Viewer or the pylon API to set the camera for a 8 bit pixel format and **no digital shift**.

Check the output of the camera under your normal lighting conditions and note the readings for the brightest pixels.

- If any of the readings are above 128, do not use digital shift.
- If all of the readings are below 128, you can safely use the shift by 1 setting.
- If all of the readings are below 64, you can safely use the shift by 1 or 2 settings.
- If all of the readings are below 32, you can safely use the shift by 1, 2, or 3 settings.
- If all of the readings are below 16, you can safely use the shift by 1, 2, 3, or 4 settings.

11.4.4 Enabling and Setting Digital Shift

You can enable or disable the digital shift feature by setting the value of the Digital Shift parameter. When the parameter is set to zero, digital shift will be disabled. When the parameter is set to 1, 2, 3, or 4, digital shift will be set to shift by 1, shift by 2, shift by 3, or shift by 4 respectively.

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

```
// Disable digital shift
Camera.DigitalShift.SetValue( 0 );

// Enable digital shift by 2
Camera.DigitalShift.SetValue( 2 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.5 Integrated IR Cut Filter (on Color Models)

Color models of the camera that have a C-mount lens adapter are equipped with an IR cut filter as standard equipment. The filter is mounted inside of the lens adapter. Cameras without an IR cut filter are available on request.

Monochrome cameras do not include an IR cut filter in the lens adapter. Monochrome cameras with a C-mount lens adapter can be equipped with a filter on request.

CAUTION

Lens Thread Length is Limited

The location of the IR cut filter limits the length of the threads on any lens you use with the camera. If a lens with a very long thread length is used, the IR cut filter will be damaged or destroyed and the camera will no longer operate.

11.6 Area of Interest (AOI)

The area of interest (AOI) feature lets you specify a portion of the imaging sensor array and after each image is acquired, only the pixel information from the specified portion of the array is transmitted to the host PC.

The area of interest is referenced to the top left corner of the array. The top left corner is designated as column 0 and line 0 as shown in Figure 44.

The location and size of the area of interest is defined by declaring an X offset (coordinate), a width, a Y offset (coordinate), and a height. For example, suppose that you specify the x offset as 10, the width as 16, the y offset as 6, and the height as 10. The area of the array that is bounded by these settings is shown in Figure 44.

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

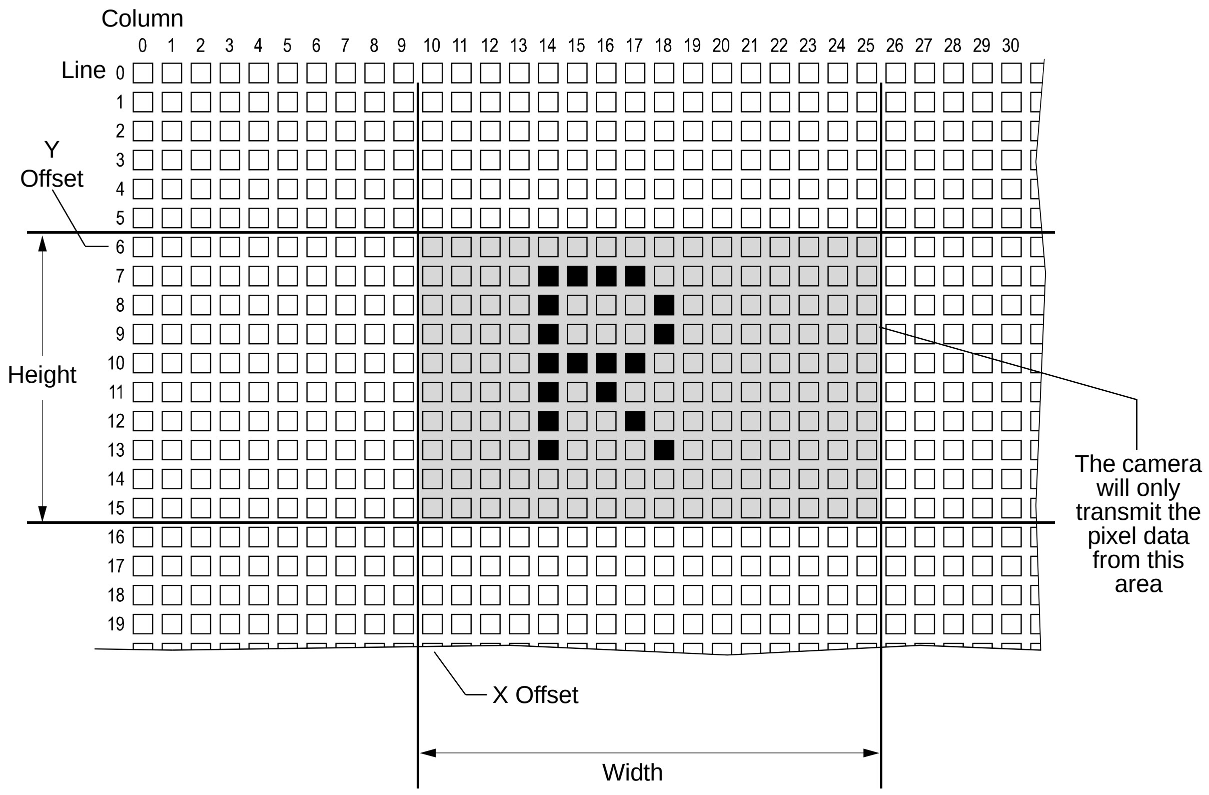


Fig. 44: Area of Interest

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 affects the maximum allowed frame rate, see Section 8.9 on page 102.

Setting the AOI

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

- The value of the X Offset parameter determines the starting column for the area of interest.
- The value of the Y Offset parameter determines the starting line for the area of interest.
- The value of the Width parameter determines the width of the area of interest.
- The value of the Height parameter determines the height of the area of interest.

When you are setting the camera’s area of interest, you must follow these guidelines on all camera models:

- The sum of the X Offset setting plus the Width setting must not exceed the width of the camera’s sensor. For example, on the piA640-210gm, the sum of the X Offset setting plus the Width setting must not exceed 648.
- The sum of the Y Offset setting plus the Height setting must not exceed the height of the camera’s sensor. For example, on the piA640-210gm, the sum of the Y Offset setting plus the Height setting must not exceed 488.

On monochrome cameras:

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

On color cameras:

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

Note

Normally, the X Offset, Y Offset, Width, and Height parameter settings refer to the physical columns and lines in the sensor. But if binning is enabled, these parameters are set in terms of "virtual" columns and lines. For more information, see Section 11.6 on page 163.

You can set the X Offset, Y Offset, Width, and Height parameter values from within your application software by using the pylon API. The following code snippets illustrate using the API to get the maximum allowed settings and the increments for the Width and Height parameters. They also illustrate setting the X Offset, Y Offset, Width, and Height parameter values

```
int64_t widthMax = Camera.Width.GetMax( );
int64_t widhInc = Camera.Width.GetInc();
Camera.Width.SetValue( 200 );
Camera.OffsetX.SetValue( 100 );

int64_t heightMax = Camera.Height.GetMax( );
int64_t heightInc = Camera.Height.GetInc();
Camera.Height.SetValue( 200 );
Camera.OffsetY.SetValue( 100 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.6.1 Changing AOI Parameters "On-the-Fly"

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

11.7 Binning

Note

The binning feature is only available on the monochrome cameras.

Binning increases the camera’s response to light by summing the charges from adjacent pixels into one pixel. Two types of binning are available: vertical binning and horizontal binning.

With vertical binning, adjacent pixels from 2 lines, 3 lines, or a maximum of 4 lines in the imaging sensor array are summed and are reported out of the camera as a single pixel. Figure 45 illustrates vertical binning.

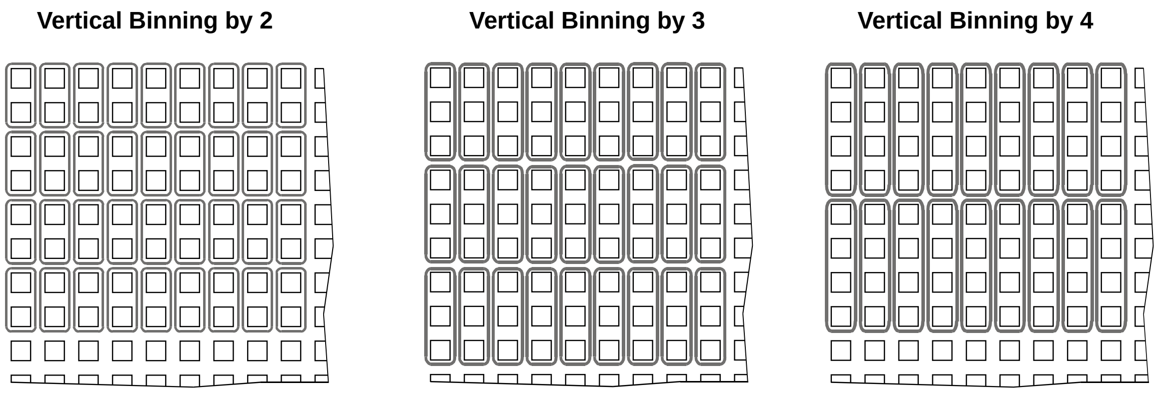


Fig. 45: Vertical Binning

With horizontal binning, adjacent pixels from 2 columns, 3 columns, or a maximum of 4 columns are summed and are reported out of the camera as a single pixel. Figure 46 illustrates horizontal binning.

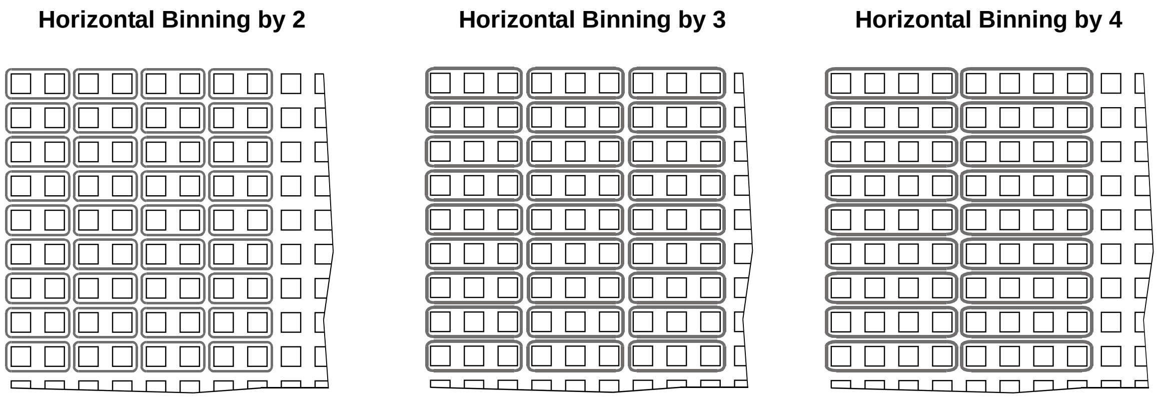


Fig. 46: Horizontal Binning

The availability of binning differs between the camera models:

Camera Model	Vertical Binning	Horizontal Binning
piA640-210gm	by 2, 3, or 4	by 2, 3, or 4
piA1000-48gm	by 2, 3, or 4	by 2, 3, or 4
piA1600-35gm	by 2, 3, or 4	by 2, 3, or 4
piA1900-32gm	by 2	by 2
piA2400-12gm	by 2, 3, or 4	by 2, 3, or 4
piA2400-17gm	by 2, 3, or 4	by 2, 3, or 4

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

Setting Binning

You can enable vertical binning by setting the Binning Vertical parameter. Setting the parameter's value to 2, 3, or 4 enables vertical binning by 2, vertical binning by 3, or vertical binning by 4 respectively. Setting the parameter's value to 1 disables vertical binning.

You can enable horizontal binning by setting the Binning Horizontal parameter. Setting the parameter's value to 2, 3, or 4 enables horizontal binning by 2, horizontal binning by 3, or horizontal binning by 4 respectively. Setting the parameter's value to 1 disables horizontal binning.

You can set the Binning Vertical or the Binning Horizontal parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter values:

```
// Enable vertical binning by 2
Camera.BinningVertical.SetValue( 2 );

// Enable horizontal binning by 4
Camera.BinningHorizontal.SetValue( 4 );

// Disable vertical and horizontal binning
Camera.BinningVertical.SetValue( 1 );
Camera.BinningHorizontal.SetValue( 1 );
```

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

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

11.7.1 Considerations When Using Binning

Increased Response to Light

Using binning can greatly increase the camera's response to light. When binning is enabled, acquired images may look overexposed. If this is the case, you can reduce the lens aperture, reduce the intensity of your illumination, reduce the camera's exposure time setting, or reduce the camera's gain setting.

Reduced Resolution

Using binning effectively reduces the resolution of the camera's imaging sensor. For example, the sensor in the piA640-210gm camera normally has a resolution of 648 (H) x 488 (V). If you set this camera to use horizontal binning by 3 and vertical binning by 3, the effective resolution of the sensor is reduced to 216 (H) by 162 (V). (Note that the 488 pixel vertical dimension of the sensor was not evenly divisible by 3, so we rounded down to the nearest whole number.)

Possible Image Distortion

Objects will only appear undistorted in the image if the numers of binned lines and columns are equal. With all other combinations, the imaged objects will appear distorted. If, for example, vertical binning by 2 is combined with horizontal binning by 4 the widths of the imaged objects will appear shrunk by a factor of 2 compared to the heights.

If you want to preserve the aspect ratios of imaged objects when using binning you must use vertical and horizontal binning where equal numbers of lines and columns are binned, e.g. vertical binning by 3 combined with horizontal binning by 3.

Binning's Effect on AOI Settings

When you have the camera set to use binning, keep in mind that the settings for your area of interest (AOI) will refer to the binned lines and columns in the sensor and not to the physical lines in the sensor as they normally would. Another way to think of this is by using the concept of a "virtual sensor." For example, assume that you are using a piA640-210gm camera set for 3 by 3 binning as described above. In this case, you would act as if you were actually working with a 216 column by 162 line sensor when setting your AOI parameters. The maximum AOI width would be 216 and the maximum AOI height would be 162. When you set the X Offset and the Width for the AOI, you will be setting these values in terms of virtual sensor columns. And when you set the Y Offset and the Height for the AOI, you will be setting these values in terms of virtual sensor lines.

For more informtion about the area of interest (AOI) feature, see Section 11.6 on page 163

Binning’s Effect on the Sensor Readout and Frame Rate Formulas

In several areas of the manual, formulas appear for sensor readout time and for calculating the maximum frame rate. In several of these formulas, you must enter the current height of the area of interest (AOI). If you are not using binning, you would enter the height of the AOI in physical sensor lines. If binning is enabled, however, you must use the concept of a "virtual" sensor as described above and the height of the AOI that you use in the formulas would be in terms of virtual sensor lines.

The affected formulas appear on page 100 and on page 103.

11.8 Reverse X

The reverse X feature is a horizontal mirror image feature. When the reverse X feature is enabled, the pixel values for each line in a captured image will be swapped end-for-end about the line's center. This means that for each line, the value of the first pixel in the line will be swapped with the value of the last pixel, the value of the second pixel in the line will be swapped with the value of the next-to-last pixel, and so on.

Figure 47 shows a normal image on the left and an image captured with reverse X enabled on the right.

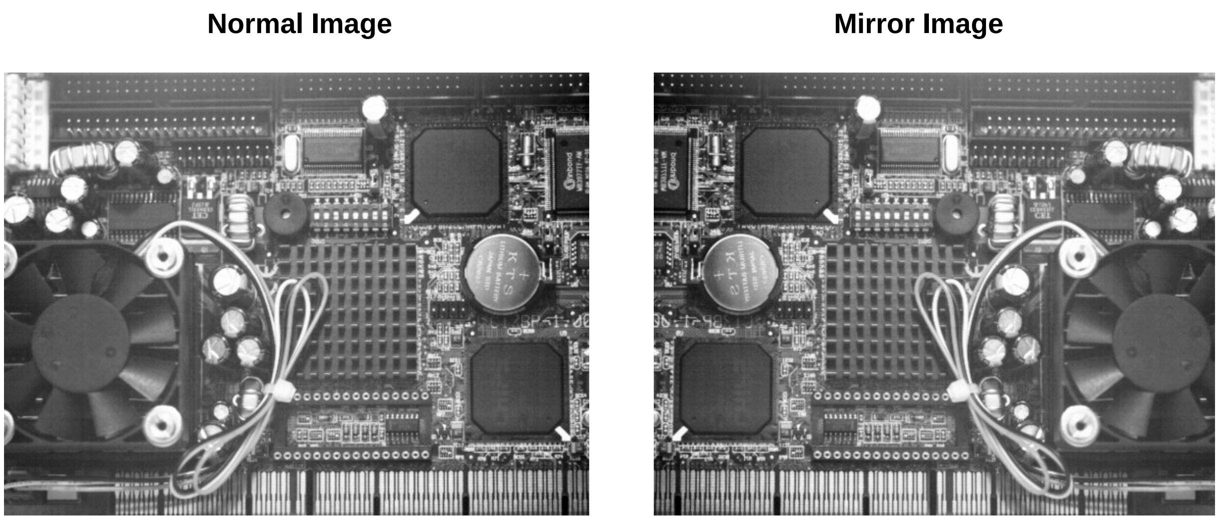


Fig. 47: Reverse X Mirror Imaging

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.

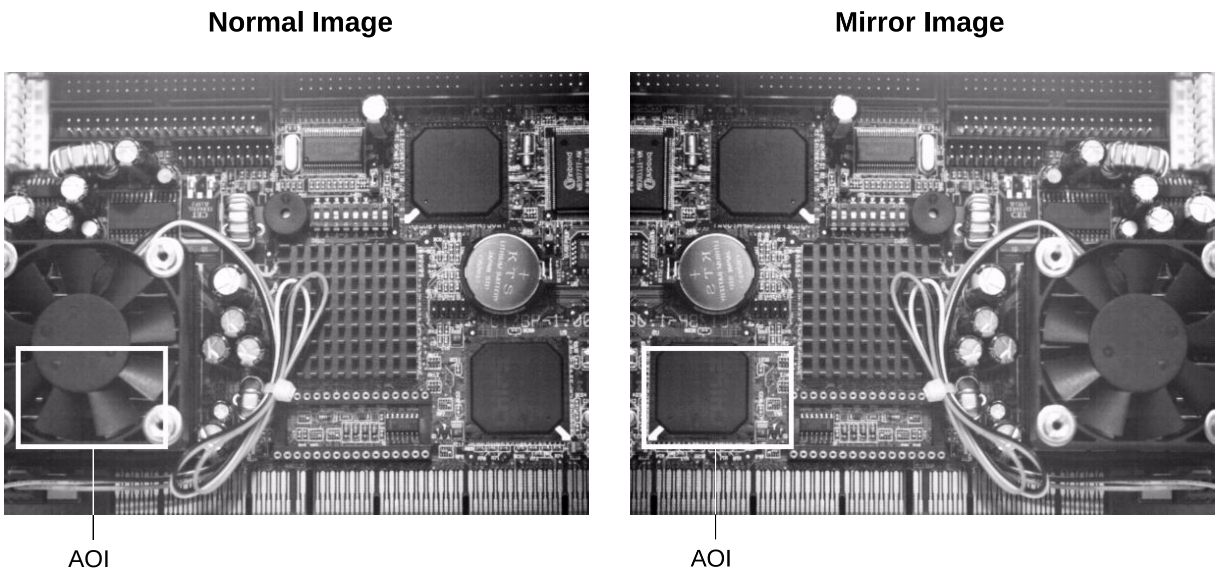


Fig. 48: Using an AOI with Reverse X Mirror Imaging

Note

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

Note

AOIs used for the auto function feature will behave analogous to "standard" AOIs:

- Depending on whether or not the reverse X feature is enabled, an Image AOI will display different images and an Auto Function AOI will refer to different image contents.
- The positions of the AOIs relative to the sensor will not change.

For more information about auto functions, see Section 11.12 on page 180

Setting Reverse X

You can enable or disable the reverse X feature by setting the ReverseX parameter value. You can set the parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
// Enable reverse X
Camera.ReverseX.SetValue(true);
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.9 Averaging

The avaraging feature lets you obtain an image that is the average of a set number of consecutively acquired individual images. You can average up to 256 individual images.

When averaging is active, the pixel values for each pixel will be summed and the total for each pixel will be divided by the number of the individual images acquired. Decimals of the resulting average pixel values will be truncated and the averaged pixel values will be transmitted as integers.

You can use averaging for all modes of image acquisition: You can obtain averaged images when the camera's acquisition mode is set to single frame and to continuous and when the camera acquires images continuously (free-run) or when triggers are used.

Each individual image must be triggered separately. Accordingly, for each averaged image the number of required triggers will be equal to the set number of individual images used for averaging.

When the camera's acquisition mode is set to single frame, a single averaged image will be obtained. The averaged image will be based on the set number of individual images. The number of triggers necessary for each averaged image will be equal to the set number of individual images. For example, if the acquisition mode is set to single frame and the number of individual images used for averaging is set to three, three triggers are needed to obtain the averaged image.

Note

Make sure that for each averaged image, the number of triggers is equal to the set number of individual images used for averaging.

Note

Make sure the object being imaged does not move while the sequence of individual images is acquired. Otherwise, the object will appear blurred in the averaged image.

Note

Although the camera allows changing the settings for all features while a sequence of individual images is acquired, we do not recommend to do so. The new settings would be applied as soon as they are set. Accordingly, the averaged image would be based on individual images acquired with different feature settings and poor quality for the averaged image may result.

We recommend to only change the feature settings while individual images used for averaging are not acquired.

When "end of exposure" event reporting is enabled, an "end of exposure" event will be reported for each image in the sequence of individual images. No "end of exposure" event will be reported specifically for the averaged image.

When a chunk feature is enabled, the data chunk from the last image in the sequence of individual images will be taken for the averaged image.

Output Frame Rate

When averaging is used, the images will be transmitted at an output frame rate which will be lower than the acquisition frame rate. As the number of averaged individual images increases, the output frame rate will decrease.

The output frame rate is described by the following formula:

$$\text{Output Frame Rate} = \frac{\text{Acquisition Frame Rate}}{\text{Number of Averaged Images}}$$

Example

Assume the acquisition frame rate is 248.4 frames per second and 3 images are averaged, then the output frame rate will be 82.2 frames per second.

Note that averaging will allow an increased acquisition frame rate compared to not using averaging, if the frame transmission is the most restricting factor. When averaging is used, Formula 3 in the "Maximum Allowed Acquisition Frame Rate" section is replaced by the following formula:

$$\text{Max. Frames/s} = \frac{\text{Device Current Throughput Parameter Value} \times \text{Number of Averaged Images}}{\text{Payload Size Parameter}}$$

Setting Averaging

You can enable averaging by setting the AveragingNumberOfFrames parameter. Setting the parameter's value to e.g. 3 enables averaging and sets 3 individual images to be averaged. Setting the parameter's value to 1 disables averaging.

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

```
// Enable averaging of 3 images
Camera.AveragingNumberOfFrames.SetValue( 3 );

// Disable averaging
Camera.AveragingNumberOfFrames.SetValue( 1 );
```

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

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

11.10 Luminance Lookup Table

The camera can capture pixel values at a 12 bit depth. When a monochrome camera is set for the Mono 16 or Mono 12 packed pixel format, the camera outputs 12 effective bits. Normally, the 12 effective bits directly represent the 12 bit output from the camera's ADC. The luminance lookup table feature lets you use a custom 12 bit to 12 bit lookup table to map the 12 bit output from the ADC to 12 bit values of your choice.

The lookup table is essentially just a list of 4096 values, however, not every value is the table is actually used. If we number the values in the table from 0 through 4095, the table works like this:

- The number at location 0 in the table represents the 12 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 0.
- The numbers at locations 1 through 7 are not used.
- The number at location 8 in the table represents the 12 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 8.
- The numbers at locations 9 through 15 are not used.
- The number at location 16 in the table represents the 12 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 16.
- The numbers at locations 17 through 23 are not used.
- The number at location 24 in the table represents the 12 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 24.
- And so on.

As you can see, the table does not include a defined 12 bit output value for every pixel value that the sensor can report. So what does the camera do when the sensor reports a pixel value that is between two values that have a defined 12 bit output? In this case, the camera performs a straight line interpolation to determine the value that it should transmit. For example, assume that the sensor reports a pixel value of 12. In this case, the camera would perform a straight line interpolation between the values at location 8 and location 16 in the table. The result of the interpolation would be reported out of the camera as the 12 bit output.

Another thing to keep in mind about the table is that location 4088 is the last location that will have a defined 12 bit value associated with it. (Locations 4089 through 4095 are not used.) If the sensor reports a value above 4088, the camera will not be able to perform an interpolation. In cases where the sensor reports a value above 4088, the camera simply transmits the 12 bit value from location 4088 in the table.

The advantage of the luminance lookup table feature is that it allows a user to customize the response curve of the camera. The graphs below represent the contents of two typical lookup tables. The first graph is for a lookup table where the values are arranged so that the output of the camera increases linearly as the sensor output increases. The second graph is for a lookup table where the values are arranged so that the camera output increases quickly as the sensor output moves from 0 through 2048 and increases gradually as the sensor output moves from 2049 through 4096.

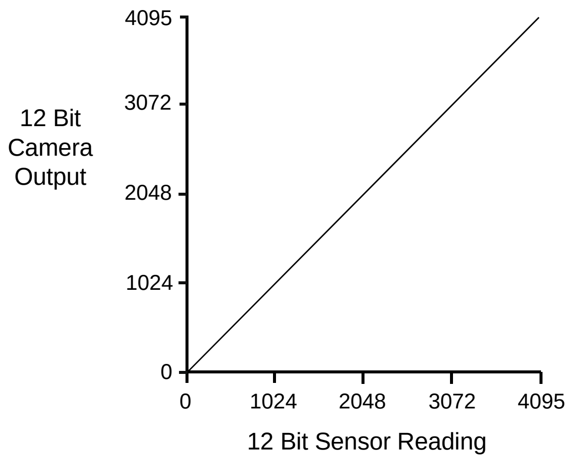


Fig. 49: Lookup Table with Values Mapped in a Linear Fashion

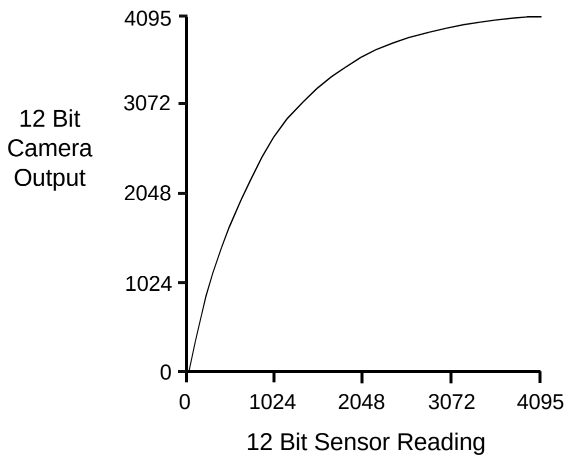


Fig. 50: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

Using the Luminance Lookup Table to Get 8 Bit Output

As mentioned above, when the camera is set for a pixel format where it outputs 12 effective bits, the lookup table is used to perform a 12 bit to 12 bit conversion. But the lookup table can also be used in 12 bit to 8 bit fashion. To use the table in 12 bit to 8 bit fashion, you enter 12 bit values into the table and enable the table as you normally would. But instead of setting the camera for a pixel format that results in 12 bit camera output, you set the camera for a pixel format that results in 8 bit output (such as Mono 8 or YUV 4:2:2 Packed). In this situation, the camera will first use the values in the table to do a 12 bit to 12 bit conversion. It will then truncate the lowest 4 bits of the converted value and will report out the remaining 8 highest bits.

Changing the Values in the Luminance Lookup Table and Enabling the Table

You can change the values in the luminance lookup table (LUT) and enable the use of the lookup table by doing the following:

- Use the LUT Selector to select a lookup table. (Currently there is only one lookup table available, i.e., the "luminance" lookup table described above.)
- Use the LUT Index parameter to select a value in the lookup table. The LUT Index parameter selects the value in the table to change. The index number for the first value in the table is 0, for the second value in the table is 1, for the third value in the table is 2, and so on.
- Use the LUT Value parameter to set the selected value in the lookup table.
- Use the LUT Index parameter and LUT value parameters to set other table values as desired.
- Use the LUT Enable parameter to enable the table.

You can set the LUT Selector, the LUT Index parameter and the LUT Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter values:

```
// Select the lookup table
Camera.LUTSelector.SetValue( LUTSelector_Luminance );

// Write a lookup table to the device.
// The following lookup table causes an inversion of the sensor values
// ( bright -> dark, dark -> bright )
for ( int i = 0; i < 4096; i += 8 )
{
   Camera.LUTIndex.SetValue( i );
   Camera.LUTValue.SetValue( 4095 - i );
}
// Enable the lookup table
Camera.LUTEnable.SetValue( true );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.11 Gamma

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

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

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.

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

A gamma correction factor between 0 and 1 will result in increased overall brightness, and a gamma correction factor 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 Gamma Correction and Setting the Gamma

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

When gamma correction is enabled, the correction factor is determined by the value of the Gamma parameter. The Gamma parameter can be set in a range from 0 to 3.99902. So if the Gamma parameter is set to 1.2, for example, the gamma correction factor will be 1.2.

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

```
// Enable the Gamma feature
Camera.GammaEnable.SetValue( true );

// Set the Gamma value to 1.2
Camera.Gamma.SetValue( 1.2 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.12 Auto Functions

11.12.1 Common Characteristics

Auto functions control image properties and are the "automatic" counterparts of certain features such as the gain feature or the white balance feature, which normally require "manually" setting the related parameter values. Auto functions are particularly useful when an image property must be adjusted quickly to achieve a specific target value and when a specific target value must be kept constant in a series of images.

An Auto Function Area of Interest (Auto Function AOI) lets you designate a specific part of the image as the base for adjusting an image property. Each auto function uses the pixel data from an Auto Function AOI for automatically adjusting a parameter value and, accordingly, for controlling the related image property. Some auto functions use their own individual Auto Function AOI and some auto functions share a single Auto Function AOI.

An auto function automatically adjusts a parameter value until the related image property reaches a target value. Note that the manual setting of the parameter value is not preserved. For example, when the Gain Auto function adjusts the gain parameter value, the manually set gain parameter value is not preserved.

For some auto functions, the target value is fixed. For other auto functions, the target value can be set, as can the limits between which the related parameter value will be automatically adjusted. For example, the gain auto function lets you set an average gray value for the image as a target value and also set a lower and an upper limit for the gain parameter value.

Generally, the different auto functions can operate at the same time. For more information, see the following sections describing the individual auto functions.

ⓘ	A target value for an image property can only be reached if it is in accord with all pertinent camera settings and with the general circumstances used for capturing images. Otherwise, the target value will only be approached. For example, with a short exposure time, insufficient illumination, and a low setting for the upper limit of the gain parameter value, the Gain Auto function may not be able to achieve the current target average gray value setting for the image.

ⓘ	You can use an auto function when binning is enabled (monochrome cameras 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 11.7 on page 166.

11.12.1.1 Modes of Operation

The following auto function modes of operation are available:

- All auto functions provide the "once" mode of operation. When the "once" mode of operation is selected, the parameter values are automatically adjusted until the related image property reaches the target value. After the automatic parameter value adjustment is complete, the auto function will automatically be set to "off" and the new parameter value will be applied to the following images.

  The parameter value can be changed by using the "once" mode of operation again, by using the "continuous" mode of operation, or by manual adjustment.
- 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, unless the camera's microcontroller is kept busy by other tasks.

  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 it is manually adjusted.
- When an auto function is set to "off", the parameter value resulting from the latest automatic adjustment will operate unless it is manually adjusted.

ⓘ	You can enable auto functions and change their settings while the camera is capturing images ("on the fly").

ⓘ	After 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.

ⓘ	If an auto function is set to "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".

11.12.1.2 Auto Function AOI

An Auto Function AOI must be set separately from the AOI used to define the size of captured images (Image AOI). You can specify a portion of the sensor array and only the pixel data from the specified portion will be used for auto function control.

An Auto Function AOI is referenced to the top left corner of the sensor array. The top left corner is designated as column 0 and row 0 as shown in Figure 44.

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 44.

Only the pixel data from within the area defined by your settings will be used by the related auto function.

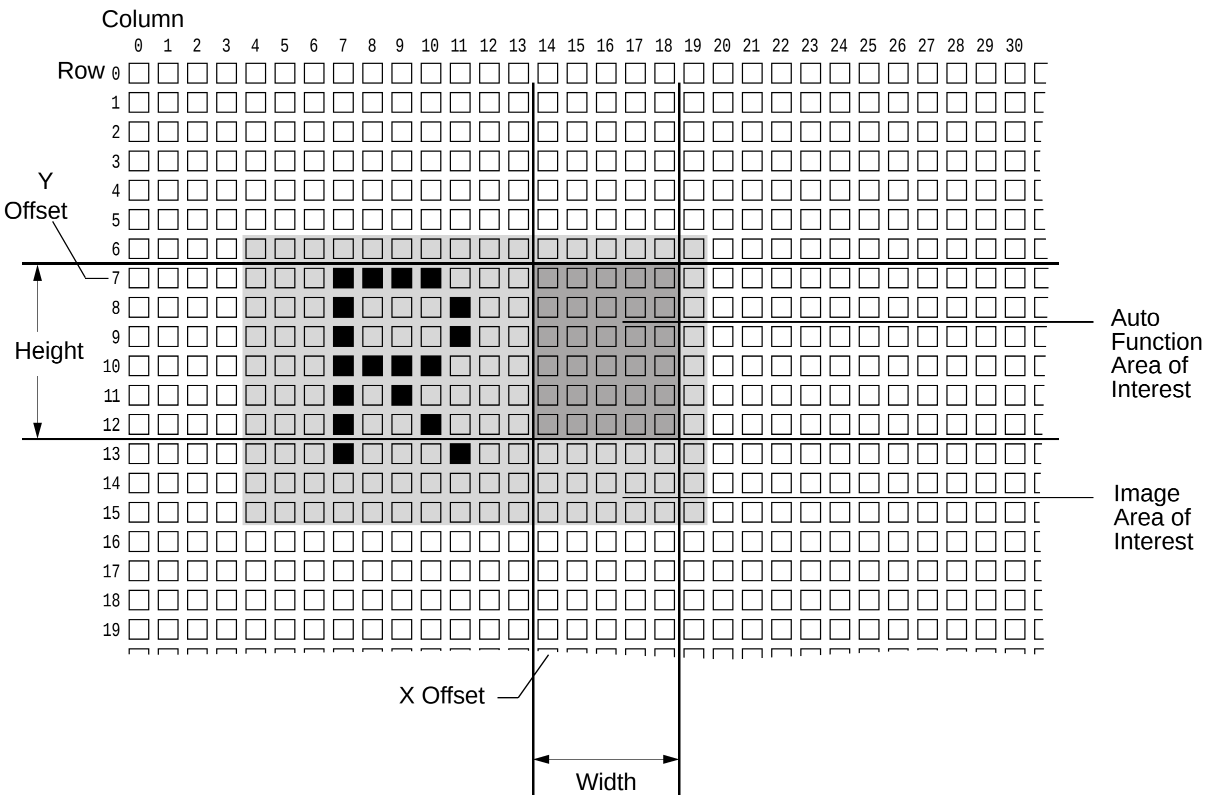


Fig. 51: Auto Function Area of Interest and Image Area of Interest

Relative Positioning of an Auto Function AOI

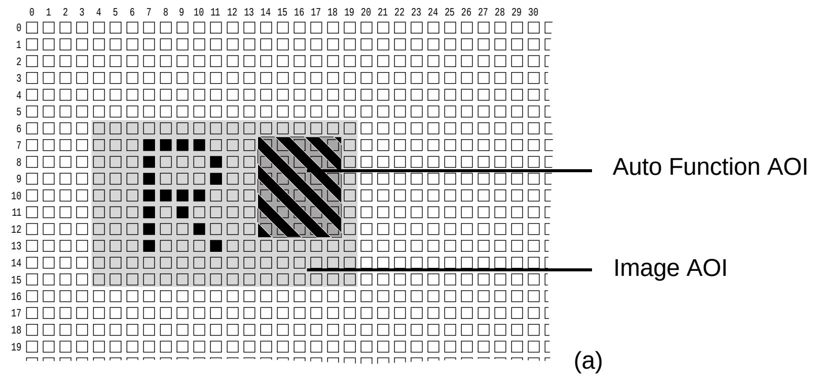
The size and position of an Auto Function AOI can be, but need not be, identical to the size and position of the Image AOI. Note that the overlap between Auto Function AOI and Image AOI determines whether and to what extent the auto function will control the related image property. Only the pixel data from the areas of overlap will be used by the auto function to control the image property of the entire image.

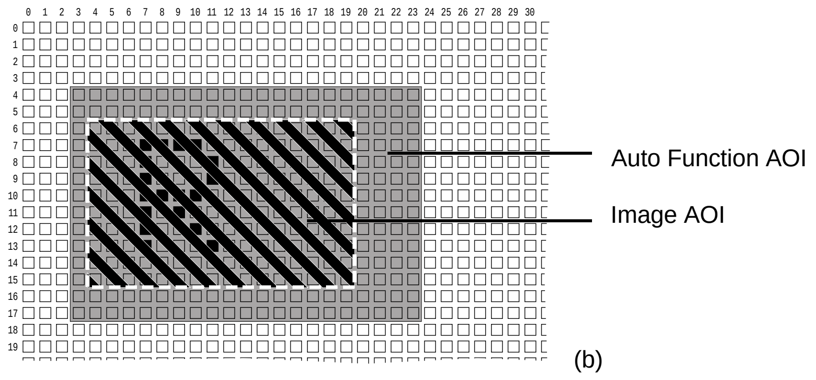
Different degrees of overlap are illustrated in Figure 52. The hatched areas in the figure indicate areas of overlap.

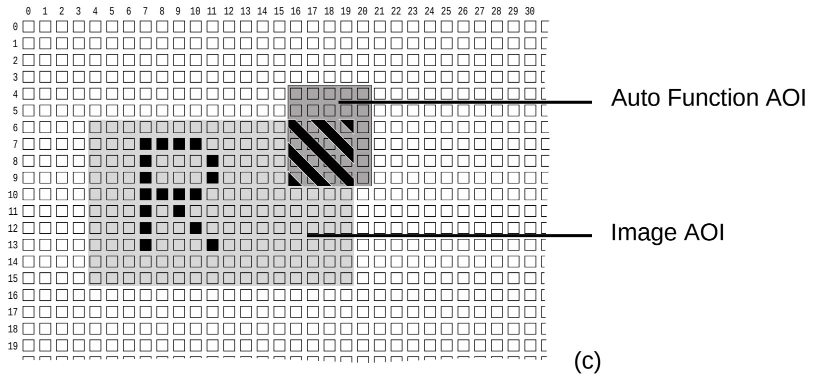
- If the Auto Function AOI is completely included in the Image AOI (see (a) in Figure 52), the pixel data from the Auto Function AOI will be used to control the image property.
- If the Image AOI is completely included in the Auto Function AOI (see (b) in Figure 52), only the pixel data from the Image AOI will be used to control the image property.
- If the Image AOI only partially overlaps the Auto Function AOI (see (c) in Figure 52), 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 52), 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 in the Image AOI, or, depending on your needs, choosing identical positions and sizes for Auto Function AOI and Image AOI.

ⓘ	You can use auto functions when also using the reverse X feature. For information about the behavior and roles of Auto Function AOI and Image AOI when also using the reverse X feature, see the "Reverse X" section.







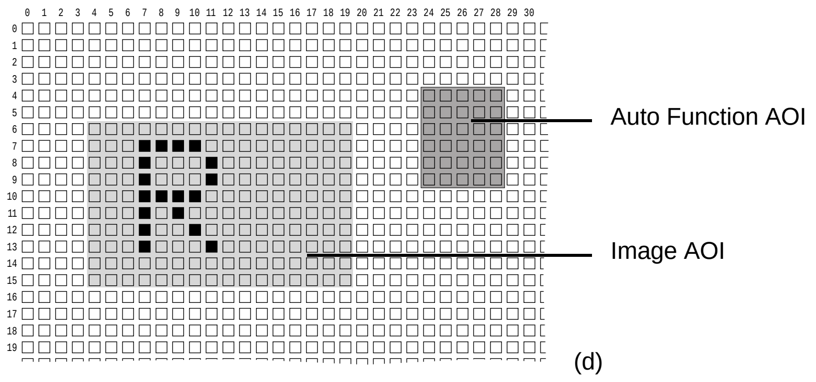


Fig. 52: Various Degrees of Overlap Between the Auto Function AOI and the Image AOI

Setting an Auto Function AOI

Setting an Auto Function AOI is a two-step process: You must first select the Auto Function AOI related to the auto function that you want to use and then set the size and the position of the Auto Function AOI.

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

- The value of the X Offset parameter determines the starting column for the Auto Function AOI.
- The value of the Y Offset parameter determines the starting line for the Auto Function AOI.
- The value of the Width parameter determines the width of the Auto Function AOI.
- The value of the Height parameter determines the height of the Auto Function AOI.

When you are setting an Auto Function AOI, you must follow these guidelines:

- The sum of the X Offset setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the piA640-21gm0, the sum of the X Offset setting plus the Width setting must not exceed 648.
- The sum of the Y Offset setting plus the Height setting must not exceed the height of the camera's sensor. For example, on the piA640-210gm, the sum of the Y Offset setting plus the Height setting must not exceed 488.

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

ⓘ	On color cameras, we strongly recommend setting the X Offset, Y Offset, Width, and Height parameters for an Auto Function AOI in increments of 2 to make the Auto Function AOI match the Bayer filter pattern of the sensor. For example, you should set the X Offset parameter to 0, 2, 4, 6, 8, etc.

ⓘ	Normally, the X Offset, Y Offset, Width, and Height parameter settings for an Auto Function AOI refer to the physical columns and lines in the sensor. But if binning is enabled (monochrome cameras only), 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 11.7.1 on page 168.

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 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 luminance statistics
// Currently AutoFunctionAOISelector_AOI1 is predefined to gather
// luminance statistics
// Set position and size of the auto function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );
```

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.

11.12.1.3 Using an Auto Function

To use an auto function, carry out the following steps:

1. Select the Auto Function AOI that is related to the auto function you want to use.
2. Set the postion and size of the Auto Function AOI.
3. If necessary, set the lower and upper limits for the auto functions’s parameter value.
4. If necessary, set the target value.
5. If necessary, set the auto function profile to define priorities between auto functions.
6. Enable the auto function by setting it to "once" or "continuous".

For more information the individual settings, see the sections below that describe the indvidual auto functions.

11.12.2 Gain Auto

Gain Auto is an auto function and the "automatic" counterpart of the manual gain feature. The gain auto function automatically carries out a Gain Raw (All) adjustment. When the gain auto function is operational, the Gain Raw (All) parameter value is automatically adjusted within set limits, until a target average gray value for the pixel data from Auto Function AOI1 is reached. Automatic adjustments for Gain Raw Tap 1 and Gain Raw Tap 2 are not available.

The gain auto function uses Auto Function AOI1 and can be operated in the "once" and continuous" modes of operation.

If Auto Function AOI1 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI1 will not be used to control the image brightness. Instead, the current manual setting of the Gain Raw parameter value will control the image brightness.

When the gain auto function is used, the exposure auto function can be used at the same time. In this case, however, you must also set the auto function profile feature.

For more information about gain, see Section 11.1 on page 149.

For more information about the auto function profile feature, see Section 11.12.4 on page 191.

To use the gain auto function, perform the following steps:

1. Select Auto Function AOI1.
2. Set the postion and size of Auto Function AOI1.
3. Set the lower and upper limits for the Gain Raw (All) parameter value.
4. Set the target average gray value.
5. If necessary, set the auto function profile.
6. Enable the gain auto function by setting it to "once" or "continuous". You must choose the "continuous" setting when using the auto function profile.

The currently settable limits for the Auto Gain Raw parameter value depend on the current pixel data format, on the current settings for binning, and on whether or not the Gain Raw parameter limits for the manually set gain feature are disabled.

The target average gray value may range from 0 (black) to 255 (white). Note that this range of numbers applies to 8 bit **and** to 16 bit (12 bit effective) output modes. Accordingly, also for 16 bit output modes, black is represented by 0 and white by 255.

You can carry out steps 1 to 6 from within your application software by using the pylon API. The following code snippets illustrate using the API to set the parameter values:

- Selecting and setting Auto Function AOI1
- Setting the limits for the Auto Gain Raw parameter value. The currently accessible minimum and maximum parameter values are chosen as examples
- Setting the target average gray value. A medium gray value is chosen as an example
- Enabling the gain auto function and selecting, for example, the "once" mode of operation

```
 // Select the appropriate auto function AOI for luminance statistics
 // Currently AutoFunctionAOISelector_AOI1 is predefined to gather
 // luminance statistics
 // Set position and size of the auto function AOI
 Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
 Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
 Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
 Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );

 // Select gain for automatic luminance control.
 // Set gain limits for luminance control
 Camera.GainSelector.SetValue( GainSelector_All );
 Camera.AutoGainRawLowerLimit.SetValue( Camera.GainRaw.GetMin() );
 Camera.AutoGainRawUpperLimit.SetValue( Camera.GainRaw.GetMax() );

 // Set target value for luminance control. This is always expressed
 // by an 8 bit value, regardless of the current pixel format
 // i.e. 0 -> black, 255 -> white
 Camera.AutoTargetValue.SetValue( 128 );

 // Set mode of operation for gain auto function
 Camera.GainAuto.SetValue( GainAuto_Once );
```

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

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

For general information about auto functions, see Section 11.12 on page 180.

For information about Auto Function AOIs and how to set them, see Section 11.12.1.2 on page 182.

11.12.3Exposure Auto

Exposure Auto is an auto function and the "automatic" counterpart to manually setting an "absolute" exposure time. The exposure auto function automatically adjusts the Exposure Time Abs parameter value within set limits, until a target average gray value for the pixel data of the related Auto Function AOI is reached.

In contrast to the manually set "absolute" exposure time, the automatically adjusted "absolute" exposure time and the settable limits for parameter value adjustment are not restricted to multiples of the current exposure time base.

The exposure auto function uses Auto Function AOI1 and can be operated in the "once" and continuous" modes of operation.

If Auto Function AOI1 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI1 will not be used to control the image brightness. Instead, the current manual setting of the Exposure Time Abs parameter value will control the image brightness.

The exposure auto function is not available, when trigger width exposure mode is selected.

When the exposure auto function is used, the gain auto function can be used at the same time. In this case, however, you must also set the auto function profile feature.

ⓘ	If the Auto Exposure Time Abs Upper Limit parameter is set to a sufficiently high value the camera’s frame rate may be decreased.

For more information about "absolute" exposure time settings and related limitations, see Section 8.4.2 on page 93.

For more information about exposure modes and how to select them, see Section 8.2.1 on page 80 and Section 8.3.1 on page 85.

For more information about the auto function profile feature, see Section 11.12.4 on page 191.

To use the exposure auto function, carry out the following steps:

1. Make sure trigger width exposure mode is not selected.
2. Select Auto Function AOI1.
3. Set the postion and size of Auto Function AOI1.
4. Set the lower and upper limits for the Exposure Time Abs parameter value.
5. Set the target average gray value.
6. If necessary, set the auto function profile.
7. Enable the exposure auto function by setting it to "once" or "continuous". You must choose the "continuous" setting when using the auto function profile.

The settable limits for the Exposure Time Abs parameter value are limited by the minimum allowed and maximum possible exposure time of the camera model.

The target average gray value may range from 0 (black) to 255 (white). Note that this range of numbers applies to 8 bit **and** to 16 bit (12 bit effective) output modes. Accordingly, also for 16 bit output modes, black is represented by 0 and white by 255.

You can carry out steps 1 to 7 from within your application software by using the pylon API. The following code snippets illustrate using the API to set the parameter values:

- Selecting and setting Auto Function AOI1: See the "Auto Function AOI" section above.
- Setting the limits for the Exposure Time Abs parameter value (the set parameter values serve as examples):
- Setting the target average gray value. A medium gray value is selected as an example:
- Enabling the exposure auto function and selecting, for example, the "continuous" mode of operation:

```
// Select the appropriate auto function AOI for luminance statistics
// Currently AutoFunctionAOISelector_AOI1 is predefined to gather
// luminance statistics
// Set position and size of the auto function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );

// Set exposure time limits for luminance control
Camera.AutoExposureTimeAbsLowerLimit.SetValue( 1000 );
Camera.AutoExposureTimeAbsUpperLimit.SetValue( 1.0E6 );

// Set target value for luminance control. This is always expressed
// by an 8 bit value, regardless of the current pixel format
// i.e. 0 -> black, 255 -> white
Camera.AutoTargetValue.SetValue( 128 );

// Set mode of operation for exposure auto function
Camera.ExposureAuto.SetValue( ExposureAuto_Continuous );
```

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

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

For general information about auto functions, see Section 11.12 on page 180.

For information about Auto Function AOIs and how to set them, see Section 11.12.1.2 on page 182.

For information about minimum allowed and maximum possible exposure time, see Table 10 in Section 8.4 on page 91

11.12.4 Auto Function Profile

If you want to use the gain auto function and the exposure auto function at the same time, you must also set the auto function profile. The auto function profile assigns priorities between related auto functions: The auto function profile specifies whether gain or exposure time shall be kept as low as possible during adjustments until a target average gray value for the pixel data of the related Auto Function AOI is reached.

To use the gain auto function and the exposure auto function at the same time, carry out the following steps:

1. Set the auto function profile to specify whether gain or exposure time shall be minimized during adjustments.
2. Set the gain auto function to the "continuous" mode of operation.
3. Set the exposure auto function to the "continuous" mode of operation.

You can set the auto function profile from within your application software by using the pylon API. The following code snippets illustrate using the API to set the auto function profile. As an example, Gain Auto is set to be minimized during adjustments:

```
// Use GainAuto and ExposureAuto simultaneously
Camera.AutoFunctionProfile.SetValue( AutoFunctionProfile_GainMinimum );
Camera.GainAuto.SetValue( GainAuto_Continuous );
Camera.ExposureAuto.SetValue( ExposureAuto_Continuous );
```

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

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

11.12.5 Balance White Auto

Balance White Auto is an auto function and the "automatic" counterpart of the manual white balance feature. The balance white auto function is only available on color models.

The automatic white balance is a two-step process. First, the Balance Ratio Abs parameter values for red, green, and blue are each set to 1.5. Then, assuming a "gray world" model, the Balance Ratio Abs parameter values are 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 AOI2 and can only be operated in the "once" mode of operation.

If Auto Function AOI2 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI2 will not be used to control the white balance of the image. However, as soon as the Balance White Auto function is set to "once" operation mode, the Balance Ratio Abs parameter values for red, green, and blue are each set to 1.5. These settings will control the white balance of the image.

For information on the white balance feature, see Section 11.3 on page 155.

To use the balance white auto function, carry out the following steps:

1. Select Auto Function AOI2.
2. Set the postion and size of Auto Function AOI2.
3. Enable the balance white auto function by setting it to "once".

You can carry out steps 1 to 3 from within your application software by using the pylon API. The following code snippet illustrates using the API to use the auto function:

- Selecting and setting Auto Function AOI2: See the "Auto Function AOI" section above.
- Enabling the balance white auto function and selecting the "once" mode of operation:

```
 // Set AOI for white balance statistics
 // Currently AutoFunctionAOISelector_AOI2 is predefined to gather
 // white balance statistics
 // Set position and size of the auto function AOI
 Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI2 );
 Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
 Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
 Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );

 // Set mode of operation for balance white auto function
 Camera.BalanceWhiteAuto.SetValue( BalanceWhiteAuto_Once );
```

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

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

For general information about auto functions, see Section 11.12 on page 180.

For information about Auto Function AOIs and how to set them, see Section 11.12.1.2 on page 182.

11.13 Disable Parameter Limits

For each camera parameter, the allowed range of parameter values normally is limited. The factory limits are designed to ensure optimum camera operation and, in particular, good image quality. For special camera uses, however, it may be helpful to set parameter values outside of the factory limits.

The disable parameter limits feature lets you disable the factory parameter limits for certain parameters. When the factory parameter limits are disabled, the parameter values can be set within extended limits. Typically, the range of the extended limits is dictated by the physical restrictions of the camera's electronic devices, such as the absolute limits of the camera's variable gain control.

The values for the extended limits can be seen using the Basler pylon Viewer or from within your application via the pylon API.

Note

Currently, the parameter limits can only be disabled on the Gain feature.

To disable the limits for a parameter:

- Use the Parameter Selector to select the parameter whose limits you wish to disable.
- Set the value of the Remove Limits parameter.

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.14 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.

The following diagram 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.

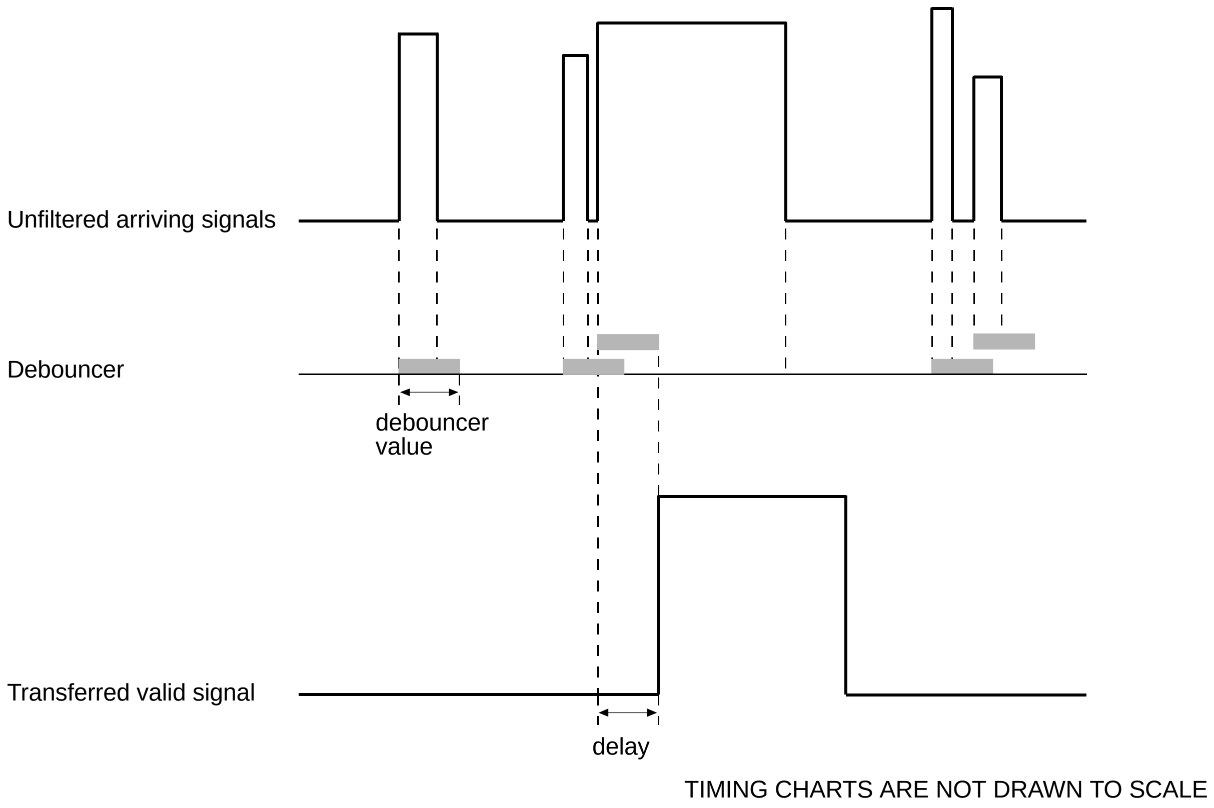

Unfiltered arriving signals
Debouncer
debouncer value
Transferred valid signal
delay
TIMING CHARTS ARE NOT DRAWN TO SCALE

Fig. 53: Filtering of Input Signals by the Debouncer

The debouncer value is determined by the value of the Line Debouncer Time Abs parameter value. The parameter is set in microseconds and can be set in a range from 0 to approximately 1 s.

To set a debouncer:

- Use the Line Selector to select the camera input line for which you want to set the debouncer (input line1 or 2).
- Set the value of the Line Debouncer Time Abs parameter.

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

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

// Set the parameter value to 100 microseconds
Camera.LineDebouncerTimeAbs.SetValue( 100 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.15 Trigger Delay

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

The trigger delay may be specified in the range from 0 to 10000000l µs (equivalent to 10 s). When the delay is set to 0 µs, no delay will be applied.

The trigger delay will not operate when the camera is triggered by your application software and when the camera operates in continuous frame mode (free run).

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

```
// Trigger delay
double TriggerDelay_us = 1000.0   // 1000us == 1ms == 0.001s;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.16 Acquisition Status

When controlling image acquisition with a software trigger you can use the acquisition staus feature to detemine when the camera is ready to be triggered for an image acquisition. Using this feature, you can avoid triggering the camera at a rate that exceeds the maximum allowed with the current camera settings.

Note

It is not possible to monitor the status of the Acquisition Start command. Therefore, you can not use the status of the Acquisition Start command to determine when the camera is ready to be triggered for an image acquisition.

To determine the acquisition status of the camera:

- Use the Acquisition Status Selector to select the Frame Trigger Wait status.
- Read the value of the AcquisitionStatus parameter. If the value is set to "false", the camera is not ready to receive a software trigger, if the value is set to "true", the camera is ready to receive a software trigger.

You can set the Acquisition Status Selector and read the AcquisitionStatus parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set and read the parameter values:

```
// Set the Acquisition Status Selector
Camera.AcquisitionStatusSelector.SetValue(
AcquisitionStatusSelector_FrameTriggerWait );

// Read the acquisition status
bool IsWaitingForFrameTrigger = Camera.AcquisitionStatus.GetValue();
```

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

You can also use the Basler pylon Viewer application to easily set the Acquisition Status Selector.

11.17 Chunk Features

This section provides detailed information about the chunk features available on each camera.

11.17.1 What Are Chunk Features?

In most cases, enabling a camera feature will simply change the behavior of the camera. The Test Image feature is a good example of this type of camera feature. When the Test Image feature is enabled, the camera outputs a test image rather than a captured image. This type of feature is referred to as a "standard" feature.

When certain camera features are enabled, the camera actually develops some sort of information about each image that it acquires. In these cases, the information is added to each image as a trailing data "chunk" when the image is transferred to the host PC. Examples of this type of camera feature are the Frame Counter feature and the Time Stamp feature. When the Frame Counter feature is enabled, for example, after an image is captured, the camera checks a counter that tracks the number of images acquired and develops a frame counter stamp for the image. And if the Time Stamp feature is enabled, the camera creates a time stamp for the image. The frame counter stamp and the time stamp would be added as "chunks" of trailing data to each image as the image is transferred from the camera. The features that add chunks to the acquired images are referred to as “chunk” features.

Before you can use any of the features that add chunks to the image, you must make the chunk mode active. Making the chunk mode active is described in the next section.

11.17.2 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, and the CRC Checksum chunk features available to be enabled.
- It automatically enables the Extended Image Data chunk feature.

To make the chunk mode active:

- Set the Chunk Mode Active parameter to true.

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

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

Note that making the chunk mode inactive switches all chunk features off.

Also note that when you enable ChunkModeActive, the PayloadType for the camera changes from "Pylon::PayloadType_Image" to "Pylon::PayloadType_ChunkData".

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

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

Once the chunk mode is active and the Extended Image Data feature has been enabled, the camera will automatically add an "extended image data" chunk to each acquired image. The extended image data chunk appended to each acquired image contains some basic information about the image. The information contained in the chunk includes:

- The X Offset, Y Offset, Width, and Height for the AOI
- The Pixel Format of the image
- The Minimum Dynamic Range and the Maximum Dynamic Range

To retrieve data from the extended image data chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the extended image data by doing the following:

- Read the value of the Chunk Offset X parameter.
- Read the value of the Chunk Offset Y parameter.
- Read the value of the Chunk Width parameter.
- Read the value of the Chunk Height parameter.
- Read the value of the Chunk Pixel Format parameter.
- Read the value of the Chunk Dynamic Range Min.
- Read the value of the Chunk Dynamic Range Max.

The following code snippet illustrates using the pylon API to run the parser and retrieve the extended image data:

```
// retrieve date from the extended image data chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
  Result.GetPayloadSize() );
int64_t offsetX = Camera.ChunkOffsetX.GetValue();
int64_t offsetY = Camera.ChunkOffsetY.GetValue();
int64_t width = Camera.ChunkWidth.GetValue();
int64_t height = Camera.ChunkHeight.GetValue();
int64_t dynamicRangeMin = Camera.ChunkDynamicRangeMin.GetValue();
int64_t dynamicRangeMax = Camera.ChunkDynamicRangeMax.GetValue();
ChunkPixelFormatEnums pixelFormat = Camera.ChunkPixelFormat.GetValue();
```

For more information about using the chunk parser, see the sample code that is included with the Basler pylon Software Development Kit (SDK).

11.17.3 Frame Counter

The Frame Counter feature numbers images sequentially as they are acquired. When the feature is enabled, a chunk is added to each image containing the value of the counter.

The frame counter is a 32 bit value. The counter starts at 0 and increments by 1 for each acquired image. The counter counts up to 4294967295 unless it is reset before (see below). After having reached the maximum value the counter will continue counting, starting at 0.

Be aware that if the camera is acquiring images continuously and continuous capture is stopped, several numbers in the counting sequence may be skipped. This happens due to the internal image buffering scheme used in the camera.

Note

The chunk mode must be active before you can enable the frame counter feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the frame counter chunk:

- Use the Chunk Selector to select the Frame Counter chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the frame counter chunk is enabled, the camera will add a frame counter chunk to each acquired image.

To retrieve data from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the frame counter information by doing the following:

- Read the value of the Chunk Frame Counter parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the frame counter chunk, run the parser, and retrieve the frame counter chunk data:

```
// make chunk mode active and enable Frame Counter chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_Framecounter );
Camera.ChunkEnable.SetValue( true );

// retrieve date from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
    Result.GetPayloadSize() );
int64_t frameCounter = Camera.ChunkFramecounter.GetValue();
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

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 image 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 1, I/O input 2 or software, and you can disable the reset. By default, the frame counter reset is disabled.

To use the frame counter reset:

- Configure the frame counter reset by setting the counter selector to Counter2 and setting the counter event source to FrameStart.
- Set the counter reset source to Line1, Line2, Software or to Off.
- Execute the command if using software as the counter reset source.

You can set the frame counter reset parameter values from within your application software by using the pylon API. The following code snippets illustrate using the API to configure and set the frame counter reset and to execute a reset via software.

```
// configure reset of frame counter
Camera.CounterSelector.SetValue( CounterSelector_Counter2 );
Camera.CounterEventSource.SetValue( CounterEventSource_FrameStart );

// select reset by signal on input line 1
Camera.CounterResetSource.SetValue( CounterResetSource_Line1 );

// select reset by signal on input line 2
```

```
Camera.CounterResetSource.SetValue( CounterResetSource_Line2 );

// select reset by software
Camera.CounterResetSource.SetValue( CounterResetSource_Software );
// execute reset by software
Camera.CounterReset.Execute();

// disable reset
Camera.CounterResetSource.SetValue( CounterResetSource_Off );
```

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

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

11.17.4Time Stamp

The Time Stamp feature adds a chunk to each acquired image containing a time stamp that was generated when image acquisition was triggered.

The time stamp is a 64 bit value. The time stamp is based on a counter that counts the number of "time stamp clock ticks" generated by the camera. The unit for each tick is 8 ns (as specified by the Gev Timestamp Tick Frequency). The counter starts at camera reset or at power off/on.

Note

The chunk mode must be active before you can enable the time stamp feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the time stamp chunk:

- Use the Chunk Selector to select the Time Stamp chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the time stamp chunk is enabled, the camera will add a time stamp chunk to each acquired image.

To retrieve data from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser that is included in the pylon API. Once the chunk parser has been used, you can retrieve the time stamp information by doing the following:

- Read the value of the Chunk Time Stamp parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the time stamp chunk, run the parser, and retrieve the frame counter chunk data:

```
// make chunk mode active and enable Time Stamp chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_Timestamp );
Camera.ChunkEnable.SetValue( true );

// retrieve data from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
  Result.GetPayloadSize() );
int64_t timeStamp = Camera.ChunkTimestamp.GetValue();
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.17.5Trigger Input Counter

The Trigger Input Counter feature numbers external image acquisition triggers sequentially as they are received. When the feature is enabled, a chunk is added to each image containing the related value of the trigger input counter.

The trigger input counter is a 32 bit value. On the first counting cycle, the counter starts at 1 and increments by 1 for each received trigger. The counter counts up to 4294967295 unless it is reset before (see below). After having reached the maximum value the counter will continue counting, starting at 0.

Be aware that if the camera is oparating in continuous frame mode (free run) the trigger input counter will not be available.

Note

The chunk mode must be active before you can enable the trigger input counter feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the trigger input counter chunk:

- Use the Chunk Selector to select the Trigger Input Counter chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the trigger input counter chunk is enabled, the camera will add a trigger input counter chunk to each acquired image.

To retrieve data from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the trigger input counter information by doing the following:

- Read the value of the Chunk Trigger Input Counter parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the trigger input counter chunk, run the parser, and retrieve the trigger input counter chunk data:

```
// make chunk mode active and enable Trigger Input Counter chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_Triggerinputcounter );
Camera.ChunkEnable.SetValue( true );
```

```
// retrieve data from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
  Result.GetPayloadSize() );
int64_t triggerinputCounter = Camera.ChunkTriggerinputcounter.GetValue();
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

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 image acquisitions.

Trigger Input Counter Reset

Whenever the camera is powered off, the trigger input counter will reset to 0. During operation, you can reset the trigger input counter via I/O input 1, I/O input 2 or software, and you can disable the reset. By default, the trigger input counter reset is disabled.

To use the trigger input counter reset:

- Configure the trigger input counter reset by setting the counter selector to Counter1 and setting the counter event source to FrameTrigger.
- Set the counter reset source to Line1, Line2, Software or to Off.
- Execute the command if using software as the counter reset source.

You can set the trigger input counter reset parameter values from within your application software by using the pylon API. The following code snippets illustrate using the API to configure and set the trigger input counter reset and to execute a reset via software.

```
// configure reset of trigger input counter
Camera.CounterSelector.SetValue( CounterSelector_Counter1 );
Camera.CounterEventSource.SetValue( CounterEventSource_FrameTrigger );

// select reset by signal on input line 1
Camera.CounterResetSource.SetValue( CounterResetSource_Line1 );

// select reset by signal on input line 2
Camera.CounterResetSource.SetValue( CounterResetSource_Line2 );

// select reset by software
Camera.CounterResetSource.SetValue( CounterResetSource_Software );
// execute reset by software
Camera.CounterReset.Execute();

// disable reset
Camera.CounterResetSource.SetValue( CounterResetSource_Off );
```

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

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

11.17.6 Line Status All

The Line Status All feature samples the status of all of the camera's input lines and output lines each time an image acquisition is triggered. It then adds a chunk to each acquired image containing the line status information.

The line status all information is a 32 bit value. As shown in Figure 54, certain bits in the value are associated with each line and the bits will indicate the state of the lines. If a bit is 0, it indicates that the state of the associated line was low at the time of triggering. If a bit is 1, it indicates that the state of the associated line is was high at the time of triggering.

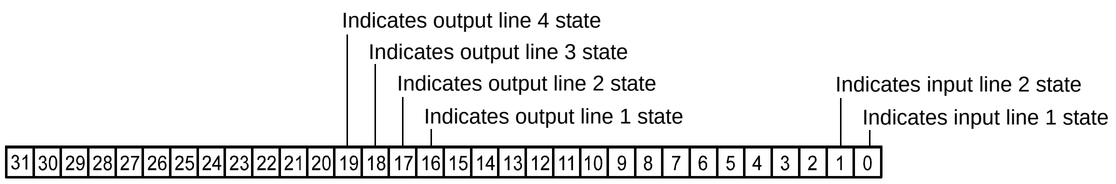

Indicates output line 4 state
Indicates output line 3 state
Indicates output line 2 state
Indicates output line 1 state
Indicates input line 2 state
Indicates input line 1 state
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 54: Line Status All Parameter Bits

Note

The chunk mode must be active before you can enable the line status all feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the line status all chunk:

- Use the Chunk Selector to select the Line Status All chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the line status all chunk is enabled, the camera will add a line status all chunk to each acquired image.

To retrieve data from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the line status all information by doing the following:

- Read the value of the Chunk Line Status All parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the line status all chunk, run the parser, and retrieve the line status all chunk data:

```
// make chunk mode active and enable Line Status All chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_LineStatusAll );
Camera.ChunkEnable.SetValue( true );
```

```
// retrieve data from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
   Result.GetPayloadSize() );
int64_t lineStatusAll = Camera.ChunkLineStatusAll.GetValue();
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.17.7 CRC Checksum

The CRC (Cyclic Redundancy Check) Checksum feature adds a chunk to each acquired image containing a CRC checksum calculated using the Z-modem method. As shown in Figure 6-2, the checksum is calculated using all of the image data and all of the appended chunks except for the checksum itself. The CRC chunk is always the last chunk appended to the image data.

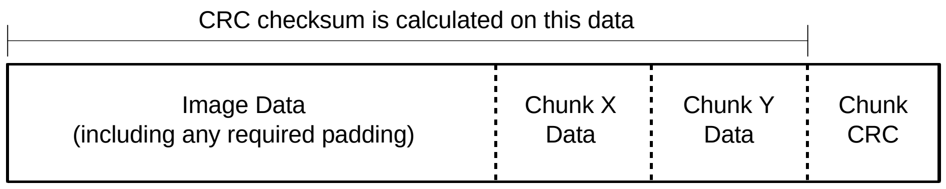

CRC checksum is calculated on this data
Image Data
(including any required padding)
Chunk X Data
Chunk Y Data
Chunk CRC

Fig. 55: CRC Checksum

Note

The chunk mode must be active before you can enable the CRC feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the CRC checksum chunk:

- Use the Chunk Selector to select the CRC chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the CRC chunk is enabled, the camera will add a CRC chunk to each acquired image.

To retrieve CRC information from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the CRC information. Note that the CRC information provided by the chunk parser is not the CRC checksum itself. Rather it is a true/false result. When the image and appended chunks pass through the parser, the parser calculates a CRC checksum based on the received image and chunk information. It then compares the calculated CRC checksum with the CRC checksum contained in the CRC checksum chunk. If the two match, the result will indicate that the image data is OK. If the two do not match, the result will indicate that the image is corrupted.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the time stamp chunk, run the parser, and retrieve the frame counter chunk data:

```
// Make chunk mode active and enable CRC chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_PayloadCRC16 );
Camera.ChunkEnable.SetValue( true );

// Check the CRC checksum of an grabbed image
IChunkParser &ChunkParser =
  *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
  Result.GetPayloadSize() );
if ( ChunkParser.HasCRC() && ! ChunkParser.CheckCRC() )
  cerr << "Image corrupted!" << endl;
```

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

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

11.18 Event Reporting

Event reporting is available on the camera. With event reporting, the camera can generate an "event" and transmit it to the PC whenever a specific situation has occurred.

Currently, the camera can generate and transmit an event for two types of situations:

- An end of an exposure has occurred
- An event overrun has occurred

An Example of Event Reporting

As an example of how event reporting works, assume that "end of exposure" event reporting has been enabled in the camera. Also assume that an end of exposure has just occurred in the camera. In this case:

1. An "end of exposure event" is created. The event contains:

   An *Event Type Identifier*. In this case, the identifier would show that an end of exposure type event has occurred.

   A *Stream Channel Identifier*. Currently this identifier is always 0.

   A *Frame ID*. This number indicates the frame count at the time that the event occurred.

   A *Timestamp*. This is a timestamp indicating when the event occurred. (The time stamp timer starts running at power off/on or at camera reset. The unit for the timer is "ticks" where one tick = 8 ns. The timestamp is a 64 bit value.)
2. The event is placed in an internal queue in the camera.
3. As soon as network transmission time is available, the camera will transmit an event message. If only one event is in the queue, the message will contain the single event. If more than one event is in the queue, the message will contain multiple events.
   a. After the camera sends an event message, it waits for an acknowledgement. If no acknowledgement is received within a specified timeout, the camera will resend the event message. If an acknowledgement is still not received, the timeout and resend mechanism will repeat until a specified maximum number of retrys is reached. If the maximum number of retrys is reached and no acknowledge has been received, the message will be dropped.

      During the time that the camera is waiting for an acknowledgement, no new event messages can be transmitted.

The Event Queue

As mentioned in the example above, the camera has an event queue. The intention of the queue is to handle short term delays in the camera's ability to access the network and send event messages. When event reporting is working "smoothly", a single event will be placed in the queue and this event will be sent to the PC in an event message before the next event is placed in queue. If there is an occasional short term delay in event message transmission, the queue can buffer several events and can send them within a single event message as soon as transmission time is available.

However if you are operating the camera at high frame rates with a small AOI, the camera may be able to generate and queue events faster than they can be transmitted and acknowledged. In this case:

1. The queue will fill and events will be dropped.
2. An event overrun will occur.
3. Assuming that you have event overrun reporting enabled, the camera will generate an "event overrun event" and place it in the queue.
4. As soon as transmission time is available, an event message containing the event overrun event will be transmitted to the PC.

The event overrun event is simply a warning that events are being dropped. The notification contains no specific information about how many or which events have been dropped.

Setting Your System for Event Reporting

To use event reporting, two conditions must be met:

- Event reporting must be enabled in the camera
- A pylon "event grabber" must be created within your application (assuming that you are using the pylon API)

The main purpose of the pylon event grabber is to receive incoming event messages.

Another purpose of the pylon event grabber is to handle event message acknowledgement. The values for the event message timeout and the event message retry count are set via the event grabber.

An event adapter object of the event grabber can be used to parse the information contained within each event message.

You can enable event reporting, create a pylon event grabber, and use the event adapter object from within your application software by using the pylon API. The pylon software development kit includes a "Camera Events" code sample that illustrates the entire process.

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

11.19 Test Images

All cameras include the ability to generate test images. Test images are used to check the camera's basic functionality and its ability to transmit an image to the host PC. Test images can be used for service purposes and for failure diagnostics. For test images, the image is generated internally by the camera's logic and does not use the optics, the imaging sensor, or the ADC. Six test images are available.

The Effect of Camera Settings on Test Images

When any of the test image is active, the camera's analog features such as gain, black level, and exposure time have no effect on the images transmitted by the camera. For test images 1, 2, 3 and 6, the cameras digital features, such as the luminance lookup table, will also have no effect on the transmitted images. But for test images 4 and 5, the cameras digital features will affect the images transmitted by the camera. This makes test images 4 and 5 as good way to check the effect of using a digital feature such as the luminance lookup table.

Enabling a Test Image

The Test Image Selector is used to set the camera to output a test image. You can set the value of the Test Image Selector to one of the test images or to "test image off".

You can set the Test Image Selector from within your application software by using the pylon API. The following code snippets illustrate using the API to set the selector:

```
// set for no test image
Camera.TestImageSelector.SetValue( TestImageSelector_Off );

// set for the first test image
Camera.TestImageSelector.SetValue( TestImageSelector_Testimage1 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

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 56.

The mathematical expression for this test image:

Gray Value = [column number + row number] MOD 256

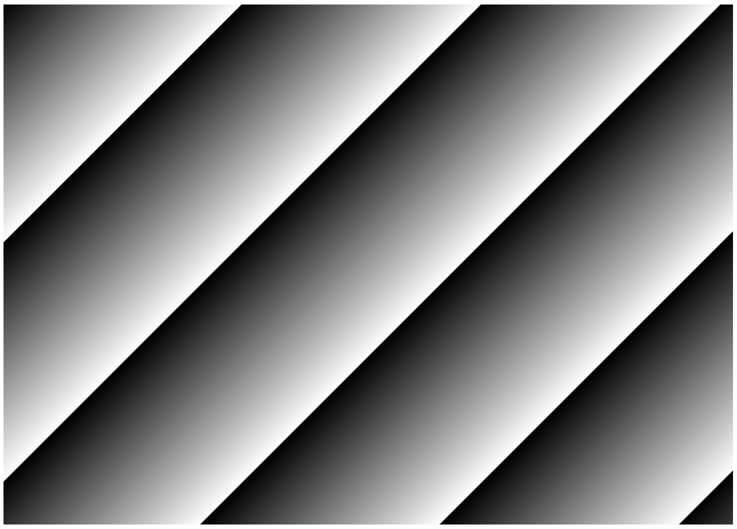


Fig. 56: Test Image One

Test Image 2 - Moving Diagonal Gray Gradient (8 bit)

The 8 bit moving diagonal gray gradient test image is similar to test image 1, but it is not stationary. The image moves by one pixel from right to left whenever a new image acquisition is initiated. The test pattern uses a counter that increments by one for each new image acquisition.

The mathematical expression for this test image is:

Gray Value = [column number + row number + counter] MOD 256

Test Image 3 - Moving Diagonal Gray Gradient (12 bit)

The 12 bit moving diagonal gray gradient test image is similar to test image 2, but it is a 12 bit pattern. The image moves by one pixel from right to left whenever a new image acquisition is initiated. The test pattern uses a counter that increments by one for each new image acquisition.

The mathematical expression for this test image is:

Gray Value = [column number + row number + counter] MOD 4096

Test Image 4 - Moving Diagonal Gray Gradient Feature Test (8 bit)

The basic appearance of test image 4 is similar to test image 2 (the 8 bit moving diagonal gray gradient image). The difference between test image 4 and test image 2 is this: if a camera feature that involves digital processing is enabled, test image 4 **will** show the effects of the feature while test image 2 **will not**. This makes test image 4 useful for checking the effects of digital features such as the luminance lookup table.

Test Image 5 - Moving Diagonal Gray Gradient Feature Test (12 bit)

The basic appearance of test image 5 is similar to test image 3 (the 12 bit moving diagonal gray gradient image). The difference between test image 5 and test image 3 is this: if a camera feature that involves digital processing is enabled, test image 5 **will** show the effects of the feature while test image 3 **will not**. This makes test image 5 useful for checking the effects of digital features such as the luminance lookup table

Test Image 6 - Moving Diagonal Color Gradient

The moving diagonal color gradient test image is available on color cameras only and is designed for use when the camera is set for YUV output. As shown in Figure 57, 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.

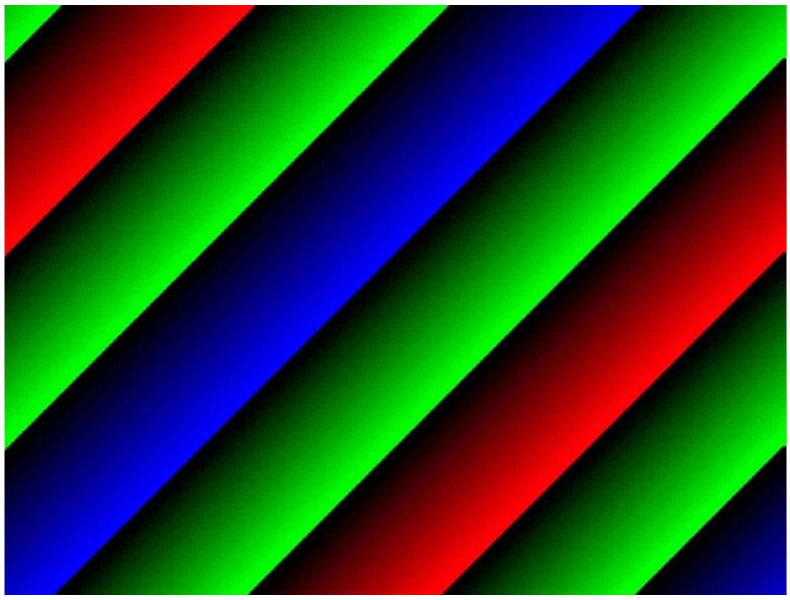


Fig. 57: Test Image Six

11.20 Device Information Parameters

Each camera includes a set of "device information" parameters. These parameters provide some basic information about the camera. The device information parameters include:

- Device Vendor Name (read only) - contains the name of the camera’s vendor. This string will always indicate Basler as the vendor.
- Device Model Name (read only) - contains the model name of the camera, for example, piA640-210gm.
- Device Manufacturer Info (read only) - can contain some information about the camera manufacturer. This string usually indicates "none".
- Device Version (read only) - contains the device version number for the camera.
- Firmware Version (read only) - contains the version of the firmware in the camera.
- Device ID (read only) - contains the serial number of the camera.
- Device User ID (read / write) - is used to assign a user defined name to a device. This name will be displayed in the Basler pylon Viewer and the Basler pylon IP Configuration Tool. The name will also be visible in the "friendly name" field of the device information objects returned by pylon’s device enumeration procedure.
- Device Scan Type (read only) - contains the scan type of the camera, for example, area scan.
- Sensor Width (read only) - contains the physical width of the sensor in pixels.
- Sensor Height (read only) - contains the physical height of the sensor.
- Max Width (read only) - Indicates the camera’s maximum area of interest (AOI) width setting.
- Max Height (read only) - Indicates the camera’s maximum area of interest (AOI) height setting.

You can read the values for all of the device information parameters or set the value of the Device User ID parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to read the parameters or write the Device User ID:

```
// Read the Vendor Name parameter
Pylon::String_t vendorName = Camera.DeviceVendorName.GetValue();

// Read the Model Name parameter
Pylon::String_t modelName = Camera.DeviceModelName.GetValue();

// Read the Manufacturer Info parameter
Pylon::String_t manufacturerInfo = Camera.DeviceManufacturerInfo.GetValue();

// Read the Device Version parameter
Pylon::String_t deviceVersion = Camera.DeviceVersion.GetValue();

// Read the Firmware Version parameter
Pylon::String_t firmwareVersion = Camera.DeviceFirmwareVersion.GetValue();
```

```
// Read the Device ID parameter
Pylon::String_t deviceID = Camera.DeviceFirmwareVersion.GetValue();

// Write and read the Device User ID
Camera.DeviceUserID = "custom name";
Pylon::String_t deviceUserID = Camera.DeviceUserID.GetValue();

// Read the Sensor Width parameter
int64_t sensorWidth = Camera.SensorWidth.GetValue();

// Read the Sensor Height parameter
int64_t sensorHeight = Camera.SensorHeight.GetValue();

// Read the Max Width parameter
int64_t maxWidth = Camera.WidthMax.GetValue();

// Read the Max Height parameter
int64_t maxHeight = Camera.HeightMax.GetValue();
```

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

You can also use the Basler pylon Viewer application to easily read the parameters and to read or write the Device User ID.

You can use the Basler pylon IP Configuration tool to read or write the Device User ID.

For more information about the pylon Viewer, see Section 3.1 on page 29.

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

11.21 Configuration Sets

A configuration set is a group of values that contains all of the parameter settings needed to control the camera. There are three basic types of configuration sets: the active configuration set, the default configuration set, and user configuration sets.

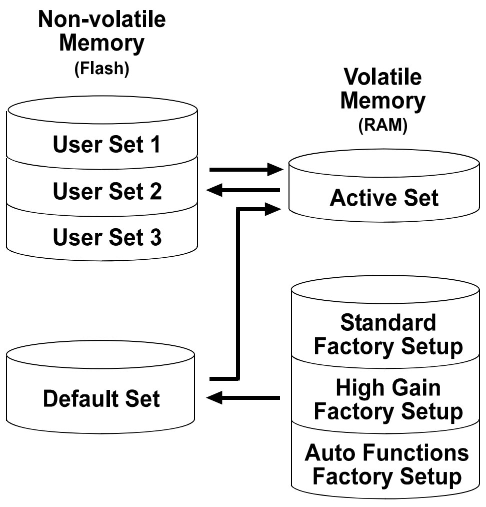


Fig. 58: Configuration Sets

Active Configuration Set

The active configuration set contains the camera's current parameter settings and thus determines the camera's performance, that is, what your image currently looks like. When you change parameter settings using the pylon API or the pylon Viewer, you are making changes to the active configuration set. The active configuration set is located in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The active configuration set is usually called the "active set" for short.

Default Configuration Set

When a camera is manufactured, numerous tests are performed on the camera and three factory optimized setups are determined. The three factory optimized setups are:

- The Standard Factory Setup - is optimized for average conditions and will provide good camera performance in many common applications. In the standard factory setup, the gain is set to a low value, and all auto functions are set to off.
- The High Gain Factory Setup - is similar to the standard factory setup, but the gain is set to + 6 dB.
- The Auto Functions Factory Setup - is similar to the standard factory setup, but the Gain Auto and the Exposure Auto auto functions are both enabled and are set to the continuous mode of operation. During automatic parameter adjustment, gain will be kept to a minimum.

The factory setups are saved in permanent files in the camera's non-volatile memory. They are not lost when the camera is reset or switched off and they cannot be changed.

You can select one of the three factory setups to be the camera's "default configuration set". Instructions for selecting which factory setup will be used as the default set appear below. Note that your selection of which factory setup will serve as the default set will not be lost when the camera is reset or switched off.

The default configuration set can be loaded into the active set. The default configuration set can also be selected as the camera's startup set. Instructions for loading the default set into the active set and for selecting the startup set appear below.

User Configuration Sets

As mentioned above, the active configuration set is stored in the camera’s volatile memory and the settings are lost if the camera is reset or if power is switched off. The camera can save most of the settings from the current active set to a reserved area in the camera’s non-volatile memory. A configuration set saved in the non-volatile memory is not lost when the camera is reset or switched off. There are three reserved areas in the camera’s non-volatile memory available for saving configuration sets. A configuration set saved in a reserved area is commonly referred to as a "user configuration set" or "user set" for short.

The three available user sets are called User Set 1, User Set 2, and User Set 3.

Note

The settings for the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If used, these settings must be set again after each camera reset or restart.

Startup Set

You can select the default configuration set or one of the user configuration sets stored in the camera’s non-volatile memory to be the "startup set." The configuration set that you have selected as the startup set will automatically be loaded into the active set whenever the camera starts up at power on or after a reset. Instructions for selecting the startup set appear below.

11.21.1 Saving User Sets

Saving the current active set into a user set in the camera’s non-volatile memory is a three step process:

- Make changes to the camera’s settings until the camera is operating in a manner that you would like to save.
- Set the User Set Selector to User Set 1, User Set 2, or User Set 3.
- Execute a User Set Save command to save the active set to the selected user set.

Saving an active set to a user set in the camera’s non-volatile memory will overwrite any parameters that were previously saved in that user set.

You can set the User Set Selector and execute the User Set Save command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet1 );
Camera.UserSetSave.Execute( );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on page 29.

11.21.2 Selecting a Factory Setup as the Default Set

When the camera is delivered, the Standard Factory Setup will be selected as the default configuration set. You can, however, select any one of the three factory setups to serve as the default set.

To select which factory setup to serve as the default set:

- Set the Default Set Selector to the Standard Factory Setup, High Gain Factory Setup or Auto Functions Factory Setup.

You can set the Default Set Selector from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector:

If you want to select the Standard Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_Standard);
```

If you want to select the High Gain Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_HighGain);
```

If you want to select the Auto Functions Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_AutoFunctions);
```

Note

Selecting which factory setup will serve as the default set is only allowed when the camera is idle, i.e. when it is not acquiring images continuously or does not have a single image acquisition pending.

Selecting the Standard Factory Setup as the default set and then loading the default set into the active set is a good course of action if you have grossly misadjusted the settings in the camera and you are not sure how to recover. The standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

11.21.3 Loading a Saved Set or the Default Set into the Active Set

If you have saved a configuration set into the camera’s non-volatile memory, you can load the saved set from the camera’s non-volatile memory into the camera’s active set. When you do this, the loaded set overwrites the parameters in the active set. Since the settings in the active set control the current operation of the camera, the settings from the loaded set will now be controlling the camera.

You can also load the default set into the camera’s active set.

To load a saved configuration set or the default set from the camera’s non-volatile memory into the active set:

- Set the User Set Selector to User Set 1, User Set 2, User Set 3 or Default.
- Execute a User Set Load command to load the selected set into the active set.

You can set the User Set Selector and execute the User Set Load command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet2 );
Camera.UserSetLoad.Execute( );
```

Note

Loading a user set or the default set into the active set is only allowed when the camera is idle, i.e. when it is not acquiring images continuously or does not have a single image acquisition pending.

Loading the Default Set with the Standard Factory Setup selected into the active set is a good course of action if you have grossly misadjusted the settings in the camera and you are not sure how to recover. The standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

11.21.4Selecting the Startup Set

You can select the default configuration set (i.e., whichever was selected as the default configuration set, either the Standard Factory Setup, the High Gain Factory Setup, or the Auto Functions Factory Setup) or one of the user configuration sets stored in the camera’s non-volatile memory to be the "startup set". The configuration set that you designate as the startup set will be loaded into the active set whenever the camera starts up at power on or after a reset.

The User Set Default Selector is used to select the startup set:

- Set the User Set Default Selector to User Set 1, User Set 2, User Set 3 or Default.

You can set the User Set Default Selector from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector:

```
Camera.UserSetDefaultSelector.SetValue( UserSetDefaultSelector_Default );
```

12 Troubleshooting and Support

This section outlines the resources available to you if you need help working with your camera.

12.1 Technical 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 on the Basler website at:
www.baslerweb.com/indizes/beitrag_index_en_22089.html

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.

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 on the Basler website at:
www.baslerweb.com/beitraege/beitrag_en_79701.html

12.2 Before Contacting Basler Technical Support

To help you as quickly and efficiently as possible when you have a problem with a Basler camera, it is important that you collect several pieces of information before you contact 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 tool.

If you cannot access the camera, please try to state the following parameter settings:

☐ Image Size (AOI):

☐ Pixel Format:

☐ Packet Size:

☐ Exposure Time:

☐ Frame Rate:

11 Live image/test image

If you are having an image problem, try to generate and save live images that show the problem. Also generate and save test images. Please save the images in BMP format, zip them, and send them to Basler technical support.

Revision History

Doc. ID Number	Date	Changes
AW00015101000	9 Feb 2007	Preliminary version of the document.
AW00015102000	22 Feb 2007	Updated the camera weight and operating temperatur range. This is still a preliminary version.
AW00015103000	24 May 2007	First release covering production cameras.
AW00015104000	8 June 2007	Modified Section 2 for the installation of the Basler pylon software, version 1.0.
AW00015105000	19 July 2007	Integrated the Kodak KAI-2093 sensor. Minor corrections throughout the manual. Added information on IP30 in Section 1.2. Added warning not to remove the serial number in Section 1.9. Updated times in Section 7.7.3. Removed note on scA750-60 output in Sections 9.2.4, 9.2.5, 9.3.8, 9.3.9, and 9.3.10. Modified the Max Gain Raw Tap 1 and Max Gain Raw Tap 2 settings for the piA640-210 and the piA1000-48 in Section 11.1. Added binning information for the piA1600-35gm in Section 11.7.
AW00015106000	20 Sept 2007	Integrated the Sony ICX625 sensor. Minor modifications and corrections throughout the manual.
AW00015107000	17 Oct 2007	Corrected the Bayer filter alignment for the piA2400-12 (in Sections 1.2, 9.2, and 9.3.1; added Sections 9.3.3, 9.3.5, and 9.3.7).
AW00015108000	5 Dec 2007	Changed the camera's family name to "pilot". Modified the exposure start delay in Section 8.8 and the constants for the max. frame rate formulas in Section 8.9 for the piA2400-12.
AW00015109000	21 Dec 2007	Added guidelines for avoiding EMI and ESD problems in Section 2.3.1 on page 40. Removed web link for further information on APIPA in Section 5.3 on page 69. Corrected the voltage ranges relating to logic 0 and logic 1 in Section 7.7.1 on page 70. Added references to Application Notes AW000565xx000 in Section 8.3.1 on page 85 and Section 8.5.1 on page 95. Added binning information for the piA1000-48gm in Section 11.7. Added the Gamma feature in Section 11.11 on page 179. Added the Disable Parameter Limits feature in Section 11.13 on page 194. Added the Debouncer feature in Section 11.14 on page 195. Minor corrections throughout the manual.

Doc. ID Number	Date	Changes
AW00015111000	15 Feb 2008	Added a note on the sensor characteristics of the piA1900-32gm/gc in Section 1.2 on page 2. Included the "Software Licensing Information" section on page 22. Moved the guidelines for avoiding EMI and ESD problems to Section 1.7 on page 23. Included the warning related to code snippets in Section 1.9 on page 25. Transferred to following sections to the "Installation and Setup Guide for Cameras Used with Basler's pylon API": "Software and Hardware Installation", "Network Recommendations", and "Camera and Network Adapter IP Configuration". Added the reference to the "Installation and Setup Guide for Cameras Used with Basler's pylon API" in Section 2 on page 27. Added the "Improve the Network Performance" step in Section 5.2.1 on page 51. Corrected the minimum value for the Timer Delay Raw parameter and indicated the minimum value for the Timer Delay Time Base Abs parameter in Section 10.2.4.2 on page 143. Minor modifications and corrections throughout the manual.
AW00015112000	5 Mar 2008	Modified mechanical drawings in Section 1.5.1.1 on page 14 and Section 1.5.2 on page 17 (dimensions, holes for screw-lock connector). Added information on the input line transition threshold in Table 5 on page 62. Added the maximum exposure times and related settings in Section 8.4.1 on page 92. Minor modifications and corrections throughout the manual.

Doc. ID Number	Date	Changes
AW00015113000	18 Jul 2008	Expanded the voltage information in Section 1.2 on page 2. Updated the distances and related tolerances between the front of the lens mount and the sensor's photosensitive area in Figure 11 on page 15 and Figure 12 on page 16. Added Information about mechanical stress test results in Section 1.5.4 on page 21. Added Information about the lens to which the mechanical stress tests apply in Section 1.5.4 on page 21. Modified the voltage information in Section 1.9 on page 25. Removed voltage information from Table 5 in Section 7.2.1 on page 62. Added Section 7.4.3 on page 67 and notes in Section 7.4.2 on page 65, introducing the PLC cable. Included detailed voltage information in Section 7.5 on page 68, Section 7.7.1.1 on page 70, and Section 7.7.2.1 on page 73. Modified the absolute maximum rating to +30.0 VDC in Section 7.7.1.2 on page 72 and Section 7.7.2.2 on page 73. Added a note relating to the debouncer in Section 8.8 on page 99. Renamed Section 11 on page 149 the Features section and included the contents of the former Chunk Features section. Corrected the minimum value for the white balance ratio in Section 11.3 on page 155. Added Section 11.9 on page 173 introducing the averaging feature. Corrected the name of the Gamma parameter in Section 11.11 on page 179. (The Gamma parameter was incorrectly referred to as the Gamma Raw parameter.) Included the "Auto Functions" section on page 180 and added related information in other parts of the manual. Extended the description of the debouncer in Section 11.14 on page 195. Minor modifications and corrections throughout the manual.
AW00015114000	22 Aug 2008	Updated contact addresses and phone numbers. Official release of the averaging feature and of the auto functions.
AW00015115000	30 Sep 2008	Added information for the new piA2400-17gm/gc models.

Doc. ID Number	Date	Changes
AW00015116000	17 June 2009	Added information (drawings inclusive) about the 90° head housing variant in Section 1 on page 1. The designations of the Kodak sensors were indicated more specifically by adding "M" and "CM" for mono and color sensors, respectively, in Section 1.2 on page 2. Added maximum sensor tilt angles for the piA2400-17gm/gc in Figure 12 on page 16 and Figure 14 on page 19. Indicated the relevance of spectral response curves for the piA2400-17gm/gc in Section 1.3 on page 8 and Section 1.4 on page 11. Section 2 on page 27, and Section 3 on page 29, and have been revised to reflect that the pylon driver package can now be downloaded from the website. Updated the minimum allowed exposure times in Section 8.4 on page 91. Added the digital shift feature in Section 11.4 on page 156. Corrected the indications of x offset and y offset in Figure 44 in Section 11.6 on page 163 and in Figure 51 in Section 11.12.1.2 on page 182. Added the reverse X feature in Section 11.8 on page 170. Removed the statement that auto functions have no effect on frame rate in Section 11.12.1 on page 180 and added a note that frame rate may be affected if exposure auto is used in Section 11.12.3 on page 189. Added a reference to the reverse X feature in Section 11.12.1.2 on page 182. Replaced "Auto Gain Raw" by the correct "Gain Raw (All)" parameter name in Section 11.12.2 on page 187. Replaced "Auto Exposure Time Abs" by the correct "Exposure Time Abs" parameter name in Section 11.12.3 on page 189. Added the auto function profile feature in Section 11.12.4 on page 191 and adjusted Section 11.12.2 on page 187 and Section 11.12.3 on page 189 accordingly.‘ Added the trigger delay feature in Section 11.15 on page 197. Added the acquisition status feature in Section 11.16 on page 198 and added a reference in Section 8.2.3 on page 82. Added descriptions about resetting the frame counter and about relating frame and trigger input counter in Section 11.17.3 on page 202. Corrected the maximum value for the frame counter in Section 11.17.3 on page 202. Added the trigger input counter feature in Section 11.17.5 on page 206. Added the high gain and auto functions factory setups and the standard factory setup (formerly the "default set") in Section 11.21 on page 220. Removed the statement that settings for frame transmission delay and inter packet delay are not saved in the user sets in Section 11.21 on page 220. Added Section 12.1 on page 225 describing how to obtain an RMA number.

Feedback

Your feedback will help us improve our documentation. Please click the link below to access an online feedback form. Your input is greatly appreciated.

http://www.baslerweb.com/umfrage/survey.html

Index