

Basler ace



USER'S MANUAL FOR CAMERA LINK CAMERAS

Document Number: AW000985

Version: 05 Language: 000 (English)

Release Date: 24 March 2015

For customers in the USA

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

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

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

For customers in Canada

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

Pour utilisateurs au Canada

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

Life Support Applications

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

Warranty Information

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

Do not remove the camera's serial number label

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

Do not open the camera housing

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

Prevent ingress or insertion of foreign substances into the camera housing

Prevent liquid, flammable, or metallic substances from entering the camera housing. If operated with any foreign substances inside, the camera may fail or cause a fire.

Avoid electromagnetic fields

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

Transport in original packaging

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

Clean with care

Avoid cleaning the camera's sensor, if possible. If you must clean it, follow the guidelines in the notice on [page 16](#). This notice also provides information on cleaning the housing.

Read the manual

Read the manual carefully before using the camera!

Contacting Basler Support Worldwide

Europe, Middle East, Africa

Basler AG
An der Strusbek 60–62
22926 Ahrensburg
Germany

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

support.europe@baslerweb.com

The Americas

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

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

support.usa@baslerweb.com

Asia-Pacific

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

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1	Specifications, Requirements, and Precautions	1
1.1	Models	1
1.2	General Specifications	2
1.3	Spectral Response	5
1.3.1	Monochrome Cameras	5
1.3.2	Color Cameras	7
1.4	Mechanical Specifications	8
1.4.1	Camera Dimensions and Mounting Points	8
1.4.2	Maximum Allowed Lens Thread Length	9
1.5	Software Licensing Information	10
1.5.1	LZ4 Licensing	10
1.6	Avoiding EMI and ESD Problems	11
1.7	Environmental Requirements	12
1.7.1	Temperature and Humidity	12
1.7.2	Heat Dissipation	12
1.8	Precautions	13
2	Installation	17
3	Tools for Changing Camera Parameters	18
3.1	Basler pylon Camera Software Suite	18
3.1.1	pylon Viewer	18
3.1.2	Basler pylon IP Configurator	19
3.1.3	pylon SDK	19
3.2	The Basler Binary Protocol Library	20
4	Camera Functional Description	21
4.1	Overview	21
5	Physical Interface	23
5.1	General Description of the Camera Connections	23
5.2	Camera Connector Types, Pin Assignments, and Pin Numbering	24
5.2.1	4-Pin Receptacle	24
5.2.2	26-Pin SDR Connectors	24
5.3	Cabling Requirements	27
5.3.1	Camera Link Cables	27
5.3.2	Power and I/O Cables	27
5.4	Camera Power	29
5.4.1	Supplying Power Over Camera Link	30
5.4.2	Supplying Power Via the 4-Pin M5 Connector	31
5.5	LED Indicator	32
5.6	General Purpose I/O (GPIO)	33
5.6.1	Introduction	33

5.6.2	Operation as an Input	34
5.6.2.1	Voltage Requirements	34
5.6.2.2	Electrical Characteristics	35
5.6.2.3	Response Time	36
5.6.3	Selecting the Input as the Source Signal for a Camera Function	37
5.6.4	Operation as an Output	37
5.6.4.1	Voltage Requirements	37
5.6.4.2	Electrical Characteristics	37
5.6.4.3	Response Time	39
5.6.4.4	Selecting a Source Signal for the Output Line	39
5.7	I/O in the Camera Link Interface	40
5.7.1	Inputs	40
5.7.2	Outputs	41
6	I/O Control	42
6.1	Input Lines	42
6.1.1	Available Input Lines	42
6.1.2	Setting the GPIO Line to Operate as an Input	43
6.1.3	Selecting an Input Line as the Source Signal for a Camera Function	44
6.1.4	Input Line Debouncer	44
6.1.5	Setting an Input Line for Invert	47
6.2	Output Lines	48
6.2.1	Available Output Lines	48
6.2.2	Setting the GPIO Line to Operate as an Output	48
6.2.3	Selecting a Source Signal for an Output Line	49
6.2.4	Setting the State of a User Settable Output Line	51
6.2.5	Setting and Checking the State of All User Settable Output Lines	52
6.2.6	Setting the State of a User Settable Synchronous Output Signal	53
6.2.7	Setting and Checking the State of All User Settable Synchronous Output Signals	55
6.2.8	Setting an Output Line for Invert	57
6.3	Checking the State of the I/O Lines	59
6.3.1	Checking the State of a Single Line	59
6.3.2	Checking the State of All Lines	60
6.4	Checking the Line Logic	61
7	Image Acquisition Control	62
7.1	Overview	62
7.2	The Acquisition Start Trigger	66
7.2.1	Acquisition Start Trigger Mode	66
7.2.1.1	Acquisition Start Trigger Mode = Off	66
7.2.1.2	Acquisition Start Trigger Mode = On	66
7.2.2	Acquisition Frame Count	68
7.2.3	Setting The Acquisition Start Trigger Mode and Related Parameters	68

7.2.4	Using a Software Acquisition Start Trigger Signal	70
7.2.4.1	Introduction	70
7.2.4.2	Setting the Parameters Related to Software Acquisition Start Triggering and Applying a Software Trigger Signal	70
7.2.5	Using a Hardware Acquisition Start Trigger Signal	71
7.2.5.1	Introduction	71
7.2.5.2	Acquisition Start Trigger Delay	72
7.2.5.3	Setting the Parameters Related to Hardware Acquisition Start Triggering and Applying a Hardware Trigger Signal	72
7.3	The Frame Start Trigger	74
7.3.1	Frame Start Trigger Mode	74
7.3.1.1	Frame Start Trigger Mode = Off	74
7.3.1.2	Frame Start Trigger Mode = On	75
7.3.1.3	Setting the Frame Start Trigger Mode and Related Parameters	77
7.3.2	Using a Software Frame Start Trigger Signal	79
7.3.2.1	Introduction	79
7.3.2.2	Setting the Parameters Related to Software Frame Start Triggering and Applying a Software Trigger Signal	80
7.3.3	Using a Hardware Frame Start Trigger Signal	82
7.3.3.1	Introduction	82
7.3.3.2	Exposure Modes	83
7.3.3.3	Frame Start Trigger Delay	86
7.3.3.4	Setting the Parameters Related to Hardware Frame Start Triggering and Applying a Hardware Trigger Signal	86
7.4	Setting the Exposure Time	89
7.5	Overlapping Exposure with Sensor Readout	95
7.6	Acquisition Monitoring Tools	98
7.6.1	Acquisition Status Indicator	98
7.6.2	Trigger Wait Signals	100
7.6.2.1	Acquisition Trigger Wait Signal	100
7.6.2.2	Frame Trigger Wait Signal	103
7.6.3	Exposure Active Signal	108
7.6.4	Frame Cycle Signal	111
7.7	Acquisition Timing Chart	113
7.8	Maximum Allowed Frame Acquisition Rate	115
7.8.1	Increasing the Maximum Allowed Frame Rate	117
7.9	Use Case Descriptions and Diagrams	119
8	Color Creation and Enhancement	127
8.1	Color Creation	127
8.1.1	Bayer Color Filter Alignment	128
8.1.2	Pixel Data Formats Available on Color Cameras	128
8.2	Integrated IR Cut Filter (All Color Models)	129

8.3	Color Enhancement Features	130
8.3.1	White Balance	130
8.3.2	Gamma Correction	131
8.3.3	Matrix Color Transformation	133
8.3.3.1	"Custom" Light Source Setting	135
9	Sensor Bit Depth, Pixel Formats, Tap Geometries, and Clock Speeds . . .	137
9.1	Sensor Bit Depth	137
9.2	Pixel Formats	138
9.3	Camera Link Tap Geometry	141
9.3.1	Overview	141
9.3.2	1X2-1Y Tap Geometry Description	143
9.3.3	1X3-1Y Tap Geometry Description	144
9.3.4	1X4-1Y Tap Geometry Description	145
9.3.5	1X6-1Y Tap Geometry Description	146
9.3.6	1X8-1Y Tap Geometry Description	147
9.3.7	1X10-1Y Tap Geometry Description	148
9.3.8	Setting the Tap Geometry	149
9.4	Camera Link Pixel Clock Speed	150
10	Features	152
10.1	Gain	152
10.2	Black Level	154
10.3	Remove Parameter Limits	155
10.4	Image Area of Interest (AOI)	156
10.4.1	Setting the Image AOI	157
10.5	Stacked Zone Imaging	160
10.5.1	Setting Stacked Zone Imaging	162
10.6	Sequencer	165
10.6.1	Auto Sequence Advance Mode	170
10.6.1.1	Operation	170
10.6.1.2	Configuration	173
10.6.2	Controlled Sequence Advance Mode	175
10.6.2.1	Operation with the "Always Active" Sequence Control Source . . .	175
10.6.2.2	Operation with an Input Line as Sequence Control Source	180
10.6.2.3	Operation with the "Disabled" Sequence Control Source	186
10.6.2.4	Configuration	190
10.6.3	Free Selection Sequence Advance Mode	194
10.6.3.1	Operation	194
10.6.3.2	Configuration	198
10.7	Binning	201
10.7.1	Setting Binning	202
10.7.2	Considerations When Using Binning	203
10.8	Vertical Decimation	206
10.8.1	Considerations When Using Decimation	207

10.9	Mirror Imaging	209
10.9.1	Reverse X	209
10.9.2	Reverse Y	211
10.9.3	Enabling Reverse X and Reverse Y	213
10.10	Luminance Lookup Table	214
10.10.1	Entering LUT Values and Enabling the LUT	217
10.11	Auto Functions	219
10.11.1	Auto Function Operating Modes	220
10.11.2	Auto Function AOIs	221
10.11.2.1	Assignment of an Auto Function to an Auto Function AOI	222
10.11.2.2	Relative Positioning of an Auto Function AOI	224
10.11.2.3	Setting an Auto Function AOI Position and Size	226
10.11.3	Using an Auto Function	228
10.11.4	Gain Auto	229
10.11.5	Exposure Auto	232
10.11.6	Auto Function Profile	235
10.11.7	Balance White Auto	236
10.12	Minimum Output Pulse Width	238
10.13	Error Detection	240
10.13.1	LED Indicator	240
10.13.2	Error Codes	241
10.14	Test Images	243
10.14.1	Test Image Descriptions	244
10.15	Device Information Parameters	246
10.16	User-Defined Values	248
10.17	Configuration Sets	249
10.17.1	Selecting a Factory Setup as the Default Set	251
10.17.2	Saving User Sets	252
10.17.3	Loading a Saved User Set or the Default Set into the Active Set	253
10.17.4	Selecting a "Startup" Set	254
	Revision History	255
	Index	258

1 Specifications, Requirements, and Precautions

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

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

1.1 Models

The current Basler ace Camera Link[®] 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 the sensor size and whether the camera's sensor is mono or color.

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

1.2 General Specifications

Specification	acA2000-340km	acA2000-340kmNIR	acA2000-340kc
Sensor Resolution (H x V pixels)	2048 x 1088	2048 x 1088	2046 x 1086
Sensor Type	CMOSIS CMV2000 Progressive scan CMOS	CMOSIS CMV2000--2E12M Progressive scan CMOS	CMOSIS CMV2000 Progressive scan CMOS
Optical Size	2/3"		
Pixel Size	5.5 µm x 5.5 µm		
Max Frame Rate (at full resolution and 10-bit sensor depth)	340 fps		
Mono/Color	Mono	Mono (NIR)	Color
Sensor Bit Depth	12 bits or 10 bits (selectable)		
Data Output Type	Camera Link base, medium, or full configuration		
Camera Link Connectors	Two 26-pin SDR connectors (also known as Mini Camera Link connectors) One 4-pin M5 receptacle		
Camera Link Clock Speed	32.5 MHz, 48 MHz, 65 MHz, or 82 MHz (selectable)		
Camera Link Tap Geometry	1X2-1Y, 1X3-1Y, 1X4-1Y, 1X6-1Y, 1X8-1Y, 1X10-1Y		
Pixel Data Formats	Mono 8 Mono10 Mono 12		Bayer GB 8 Bayer GB 10 Bayer GB 12
Synchronization	Via external trigger signal, software trigger signal, or free run		
Exposure Control	Trigger width or timed		
Power Requirements	PoCL (Power over Camera Link compliant) +12 VDC (± 10 %), < 1 % ripple, supplied via the camera's 4-pin connector		
Max Power Consumption	3.0 W @ 12 VDC		
I/O Lines	3 input lines (CC1 to CC3) and 1 output line (via Camera Link spare bit) 1 GPIO (can be set to operate as an input or an output)		
Lens Adapter	C-mount		
Size (L x W x H)	43.5 mm x 29.0 mm x 29.0 mm (without lens adapter or connectors) 58.9 mm x 29.0 mm x 29.0 mm (with lens adapter and connectors)		
Weight	< 90 g (typical)		
Conformity	CE (*), FCC, GenICam, RoHS, Camera Link, IP30 * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com		

Table 1: General Specifications (acA2000-340km, acA2000-340kmNIR, acA2000-340kc)

Specification	acA2000-340 km	acA2000-340 kmNIR	acA2000-340 kc
Software	Basler pylon 4 Camera Software Suite: Available for Windows in 32-bit and 64-bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).		

Table 1: General Specifications (acA2000-340km, acA2000-340kmNIR, acA2000-340kc)

Specification	acA2040-180 km	acA2040-180 kmNIR	acA2040-180 kc
Sensor Resolution (H x V pixels)	2048 x 2048		2046 x 2046
Sensor Type	CMOSIS CMV4000 Progressive scan CMOS	CMOSIS CMV4000--2E12M Progressive scan CMOS	CMOSIS CMV4000 Progressive scan CMOS
Optical Size	1"		
Pixel Size	5.5 μm x 5.5 μm		
Max Frame Rate (at full resolution and 10-bit sensor depth)	180 fps		
Mono/Color	Mono	Mono (NIR)	Color
Sensor Bit Depth	12 bits or 10 bits (selectable)		
Data Output Type	Camera Link base, medium, or full configuration		
Camera Link Connectors	Two 26-pin SDR connectors (also known as Mini Camera Link connectors) One 4-pin M5 receptacle		
Camera Link Clock Speed	32.5 MHz, 48 MHz, 65 MHz, 82 MHz (selectable)		
Camera Link Tap Geometry	1X2-1Y, 1X3-1Y, 1X4-1Y, 1X6-1Y, 1X8-1Y, 1X10-1Y		
Pixel Data Formats	Mono 8 Mono10 Mono 12	Bayer GB 8 Bayer GB 10 Bayer GB 12	
Synchronization	Via external trigger signal, software trigger signal, or free run		
Exposure Control	Trigger width or timed		
Power Requirements	PoCL (Power over Camera Link compliant) +12 VDC ($\pm 10\%$), < 1 % ripple, supplied via the camera's 4-pin connector		
Max Power Consumption	3.0 W @ 12 VDC		
I/O Lines	3 input lines (CC1 to CC3) and 1 output line (via Camera Link spare bit) 1 GPIO (can be set to operate as an input or an output)		
Lens Adapter	C-mount		
Size (L x W x H)	43.5 mm x 29.0 mm x 29.0 mm (without lens adapter or connectors) 58.9 mm x 29.0 mm x 29.0 mm (with lens adapter and connectors)		

Table 2: General Specifications (acA2040-180km, acA2040-180kmNIR, acA2040-180kc)


Specification	acA2040-180 km	acA2040-180 kmNIR	acA2040-180 kc
Weight	< 90 g (typical)		
Conformity	CE (*), FCC, GenICam, RoHS, Camera Link, IP30 * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com		
Software	Basler pylon 4 Camera Software Suite: Available for Windows in 32-bit and 64-bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).		

Table 2: General Specifications (acA2040-180km, acA2040-180kmNIR, acA2040-180kc)

1.3 Spectral Response

1.3.1 Monochrome Cameras

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

	<p>The spectral response curves exclude lens characteristics and light source characteristics.</p>
---	--

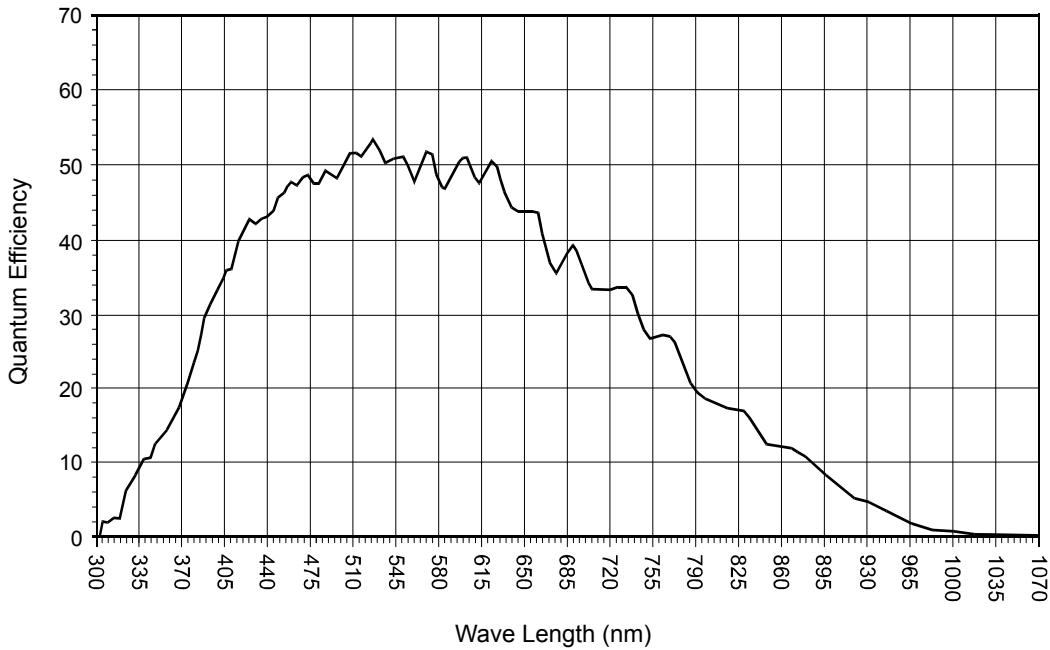


Fig. 1: acA2000-340km, acA2040-180km Spectral Response (From Sensor Data Sheet)

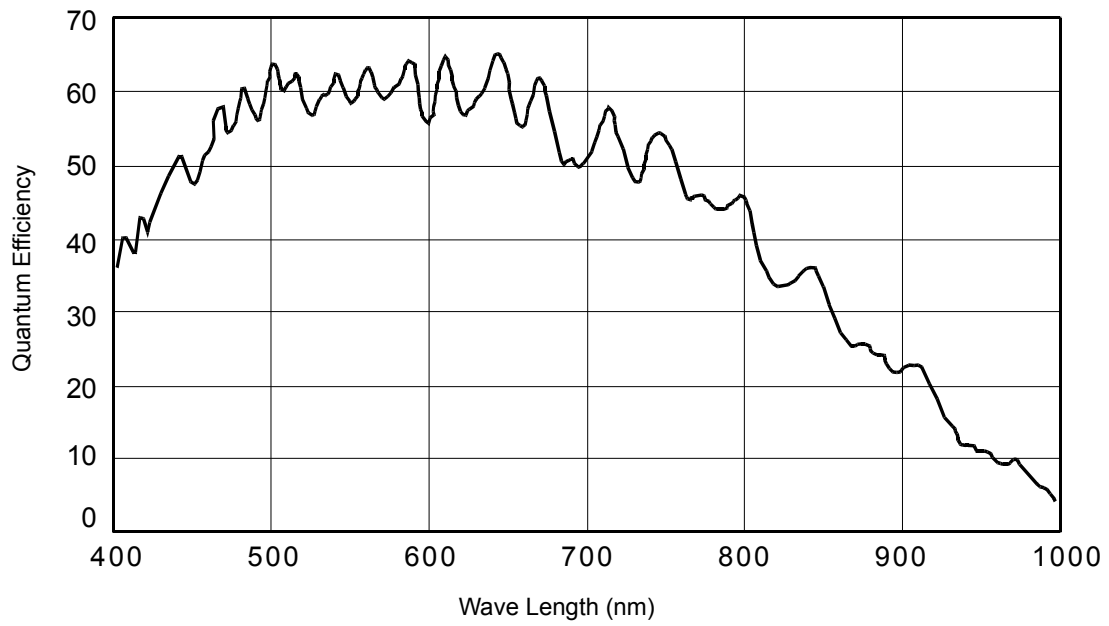



Fig. 2: acA2000-340kmNIR, acA2040-180kmNIR Spectral Response (From Sensor Data Sheet)

1.3.2 Color Cameras

The following graph shows the spectral response for all color cameras.

	<p>The spectral response curve excludes lens characteristics, light source characteristics, and IR cut filter characteristics.</p> <p>To achieve the best performance with 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.</p> <p>A suitable IR cut filter is included in the standard C-mount lens adapter on color models of the camera.</p>
---	---

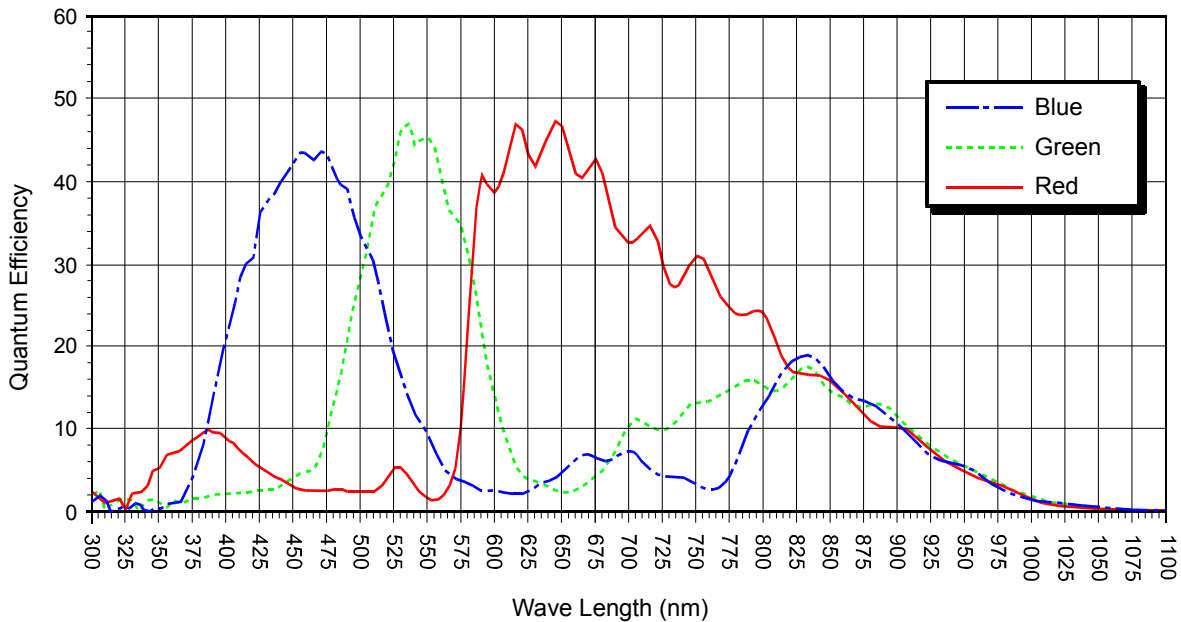


Fig. 3: Color Camera Spectral Response

1.4 Mechanical Specifications

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

1.4.1 Camera Dimensions and Mounting Points

The camera dimensions in millimeters are as shown in Figure 4.

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

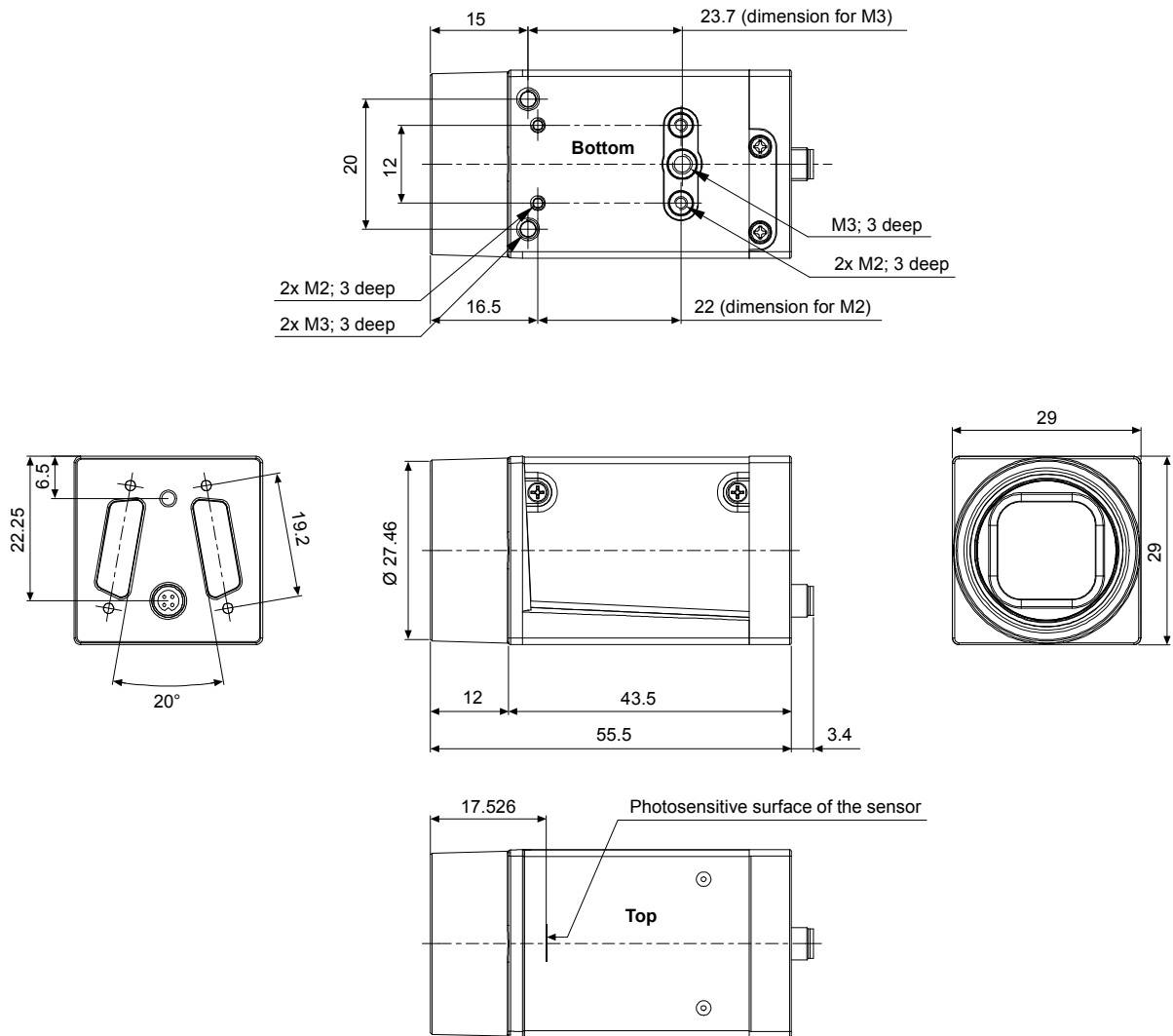


Fig. 4: Mechanical Dimensions (in mm)

1.4.2 Maximum Allowed Lens Thread Length

The C-mount lens mount on all cameras is normally equipped with a plastic filter holder. As shown in Figure 5, the length of the threads on any lens you use with the camera can be a maximum of 9.6 mm, and the lens can intrude into the camera body a maximum of 10.8 mm. If either of these limits is exceeded, the lens mount or the filter holder will be damaged or destroyed and the camera will no longer operate properly.

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

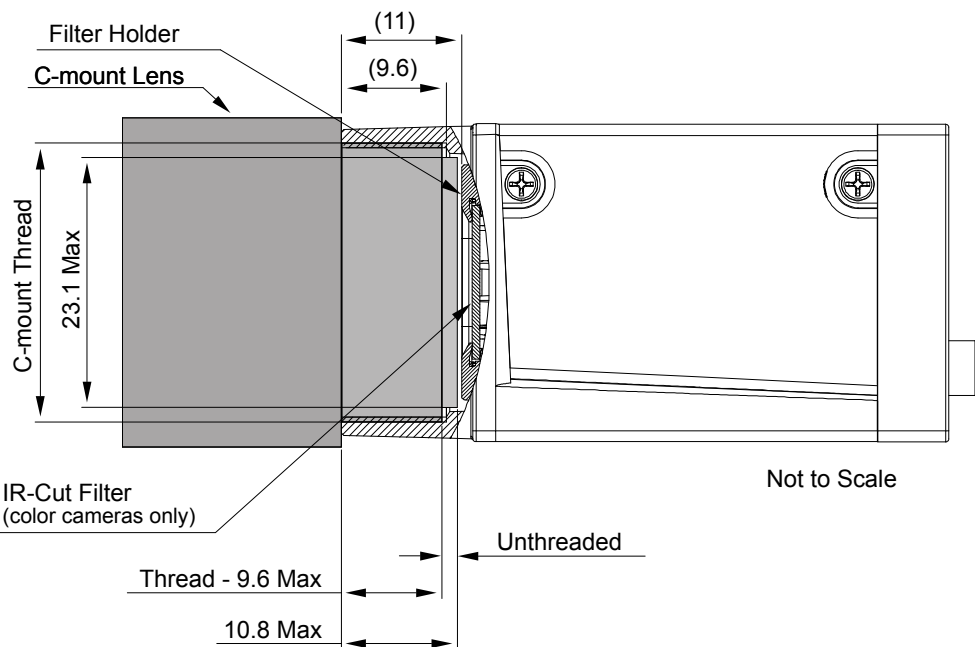


Fig. 5: Maximum Lens Thread Length (dimensions in mm)

1.5 Software Licensing Information

1.5.1 LZ4 Licensing

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

LZ4 - Fast LZ compression algorithm

Copyright (C) 2011-2013, Yann Collet.

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

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

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

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

1.6 Avoiding EMI and ESD Problems

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

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

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



The Basler application note called *Avoiding EMI and ESD in Basler Camera Installations* provides much more detail about avoiding EMI and ESD. You can download this application note from the Basler website: www.baslerweb.com

1.7 Environmental Requirements

1.7.1 Temperature and Humidity

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

1.7.2 Heat Dissipation

You must provide sufficient heat dissipation to maintain the temperature of the camera housing at 50 °C or less. Since each installation is unique, Basler only provides the following general guidelines:

- In all cases, you should monitor the temperature of the camera housing and make sure that the temperature does not exceed 50 °C. Keep in mind that the camera will gradually become warmer during the first hour of operation. After one hour, the housing temperature should stabilize and no longer increase.
- Provide sufficient heat dissipation by e.g. mounting the camera on a substantial, thermally conductive component that can act as a heat sink and by using a fan to provide an air flow over the camera.



To ensure good image quality, Basler recommends not to operate the camera at temperatures above the maximum temperature specified in Section 1.7.1.

1.8 Precautions



DANGER

Electric Shock Hazard

Non-approved power supplies may cause electric shock. Serious injury or death may occur.

- You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to
 - power supplies for I/O line power and
 - power supplies for camera power.
- If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.



WARNING

Fire Hazard

Non-approved power supplies may cause fire and burns.

- You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to
 - power supplies for I/O line power and
 - power supplies for camera power.
- If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.

NOTICE

On all cameras, the lens thread length is limited.

All cameras (mono and color) are equipped with a plastic filter holder located in the lens mount. The location of the filter holder limits the length of the threads on any lens you use with the camera. If a lens with a very long thread length is used, the filter holder or the lens mount will be damaged or destroyed and the camera will no longer operate properly.

For more specific information about the lens thread length, see Section 1.4.2 on [page 9](#).

NOTICE

Voltage outside of the specified range can cause damage.

- If you are supplying power to the camera via Power over Camera Link (PoCL), you must use a PoCL-compliant frame grabber and you must use Camera Link cables that are specifically designed for PoCL as specified in the Camera Link standard. Failure to use a PoCL-compliant frame grabber or the correct cables can result in severe damage to the camera.
- If you are supplying power to the camera via the 4-pin M5 connector, the voltage of the power to the camera must be between +10.8 VDC and +13.2 VDC.
 - If the voltage is greater than +13.2 VDC, severe damage to the camera can result.
 - If the voltage is less than +10.8 VDC, the camera may operate erratically.
- Applying power with the wrong polarity can result in severe damage to the camera.
- The ace CL cameras must only be connected to other limited power sources (LPS) / Safety Extra Low Voltage (SELV) circuits.

For more information about camera power, see Section 5.4 on [page 29](#).

NOTICE

Making or breaking Camera Link connections incorrectly can severely damage the camera.

- If you are supplying power to the camera via the Camera Link connection (PoCL), be sure that the power to the frame grabber is switched off before you connect or disconnect the Camera Link cables.
- If you are supplying power to the camera via the 4-pin M5 connector, switch off the power to the connector before you connect or disconnect the Camera Link cables.

NOTICE

The camera's GPIO line can be set to operate as an input or as an output. Applying incorrect electrical signals to the camera's GPIO line can severely damage the camera.

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

For more information about setting the GPIO line operation, see Section 6.1 on [page 42](#).

For more information about the electrical requirements for the GPIO line, see Section 5.6 on [page 33](#).

NOTICE

An incorrect plug can damage the 4-pin connector.

The plug on the cable that you attach to the camera's 4-pin connector must have 4 female pins. Using a plug designed for a smaller or a larger number of pins can damage the connector.

NOTICE

Inappropriate code may cause unexpected camera behavior.

- The code snippets provided in this manual are included as sample code only. Inappropriate code may cause your camera to function differently than expected and may compromise your application.
- To ensure that the snippets will work properly in your application, you must adjust them to meet your specific needs and must test them thoroughly prior to use.
- The code snippets in this manual are written in C++. Other programming languages can also be used to write code for use with Basler pylon. When writing code, you should use a programming language that is both compatible with pylon and appropriate for your application. For more information about the programming languages that can be used with Basler pylon, see the documentation included with the pylon software.

NOTICE

Avoid dust on the sensor.

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

NOTICE**Cleaning of the sensor and the housing****Sensor**

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

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

Housing

To clean the surface of the camera housing:

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

2 Installation

The information you will need to do a quick, simple installation of the camera is included in the *Installation and Setup Guide for Camera Link Cameras* (AW000996).

More extensive information about the pylon Camera Software Suite can be found in the *Installation and Setup Guide for Cameras Used with the Basler pylon Camera Software Suite* (AW000611).

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



DANGER

Electric Shock Hazard

Non-approved power supplies may cause electric shock. Serious injury or death may occur.

- You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to
 - power supplies for I/O line power and
 - power supplies for camera power.
- If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.



WARNING

Fire Hazard

Non-approved power supplies may cause fire and burns.

- You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to
 - power supplies for I/O line power and
 - power supplies for camera power.
- If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.

3 Tools for Changing Camera Parameters

This chapter provides an overview of the options available for changing the camera's parameters:

- The options available with the Basler pylon Camera Software Suite let you change parameters and control the camera by using a stand-alone GUI (known as the pylon Viewer) or by accessing the camera from within your software application using the pylon SDK.
- You can also control the camera and change parameters via direct access to the camera's register structure.

3.1 Basler pylon Camera Software Suite

The Basler pylon Camera Software Suite is designed for use with all Basler cameras with the following interface types: IEEE 1394a, IEEE 1394b, GigE, USB 3.0. It can also be used with newer Camera Link cameras.

For information about the features included in the pylon Camera Software Suite and about installing pylon software, see the *Installation and Setup Guide for Cameras Used With the Basler pylon Camera Software Suite* (AW000611). You can download the document from the Basler website: www.baslerweb.com

The pylon Camera Software Suite is available for Windows and Linux operating systems and includes several tools, in particular the pylon Viewer and the pylon SDK, that you can use to change the parameters on your camera. The remaining sections in this chapter provide an introduction to these tools.

3.1.1 pylon Viewer

The pylon Viewer is included in Basler's pylon camera software suite. The pylon Viewer is a standalone application that lets you view and change most of the camera's parameter settings via a GUI-based interface. 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.

Note that with Camera Link cameras the viewer can't display the images captured by the camera. Most frame grabbers include software that will display the captured images, so you can use the pylon Viewer to view or change the camera's parameters and the frame grabber software to view the captured images.

3.1.2 Basler pylon IP Configurator

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

For more information about using the IP Configurator, see the *Installation and Setup Guide for Cameras Used With the Basler pylon Camera Software Suite* (AW000611).

3.1.3 pylon SDK

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

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

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

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

3.2 The Basler Binary Protocol Library

Basler ace Camera Link cameras have blocks of mapped memory space known as registers. Reading values from the registers will give you basic information about the camera and tell you the camera's current settings. By writing values to the registers, you can control how the camera's features will operate.

If you use the Basler pylon software described in the previous section, the camera's register structure is hidden. With pylon, a series of function calls allows you to change camera parameter settings without the need to know anything about the register that underlies each parameter.

If desired, you can also change the camera parameter settings and control the camera by directly accessing the camera's register structure. The Basler Binary Protocol Library (BBPL) provides functions that allow you to read data from or write data to the camera's registers. The BBPL is an extension of the `clALLSerial/clSerial` API defined in Chapter 8 of the Camera Link Standard version 2 or higher. The BBPL adds convenience functions to this API that allow you to read from and write to the registers in Basler Camera Link cameras. The read and write requests are transmitted to the camera via a serial link between the camera and the frame grabber; the serial link is part of the standard Camera Link interface.

Sample code showing how to use the BBPL along with supporting documentation can be downloaded from the Basler website: www.baslerweb.com

When using the BBPL to change parameter values, you will need to know the details of the camera's register structure. For details of the register structure, refer to the document called *ace and aviator Register Structure and Access Methods* (AW000997). The document can also be downloaded from the Basler website.

Note that if you are using an earlier Basler Camera Link camera that was originally designed to work with the Basler Binary Protocol II (e.g., the A400k, L400k, L800k, and the sprint), you can now use either the BBPL or the Binary Protocol II to access the camera's registers.

4 Camera Functional Description

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

4.1 Overview

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 Camera Link interface.

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

Accumulated charges are read out of the sensor when exposure ends. The accumulated charges are read out of the sensor sequentially, row-by-row. As shown in Figure 6 on [page 22](#), each row passes through an "analog front end" where the charges are transformed to voltages, gain and black level are adjusted, and an analog to digital conversion is performed. The A to D conversion can be performed at 10-bit or at 12-bit depth.

As shown in Figure 7 on [page 22](#), the digitized values from each row are passed to an LVDS block where they are transmitted to the camera's FPGA. The Image data is then transmitted via the Camera Link interface to the frame grabber in your PC. Several different "tap geometries" are available to determine how the pixel data will be transmitted over the Camera Link interface. The user can select the desired tap geometry. See Section 9.3 on [page 141](#) for more detailed information about tap geometries.

The user can also select from several available pixel clock speeds for the Camera Link interface. See Section 10.1 on [page 152](#) for more information about selecting the pixel clock speed.

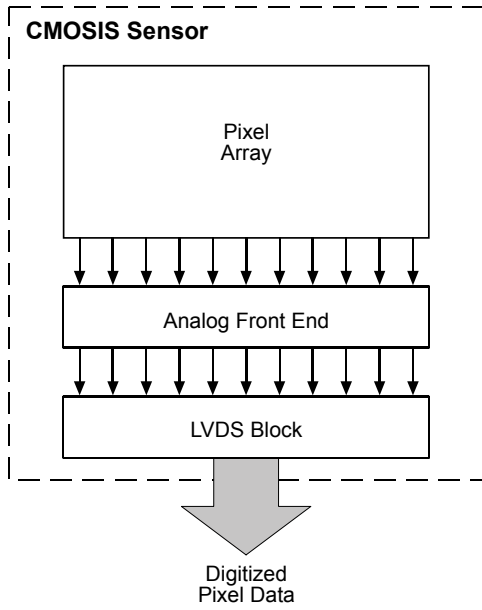


Fig. 6: CMOSIS Sensor Architecture

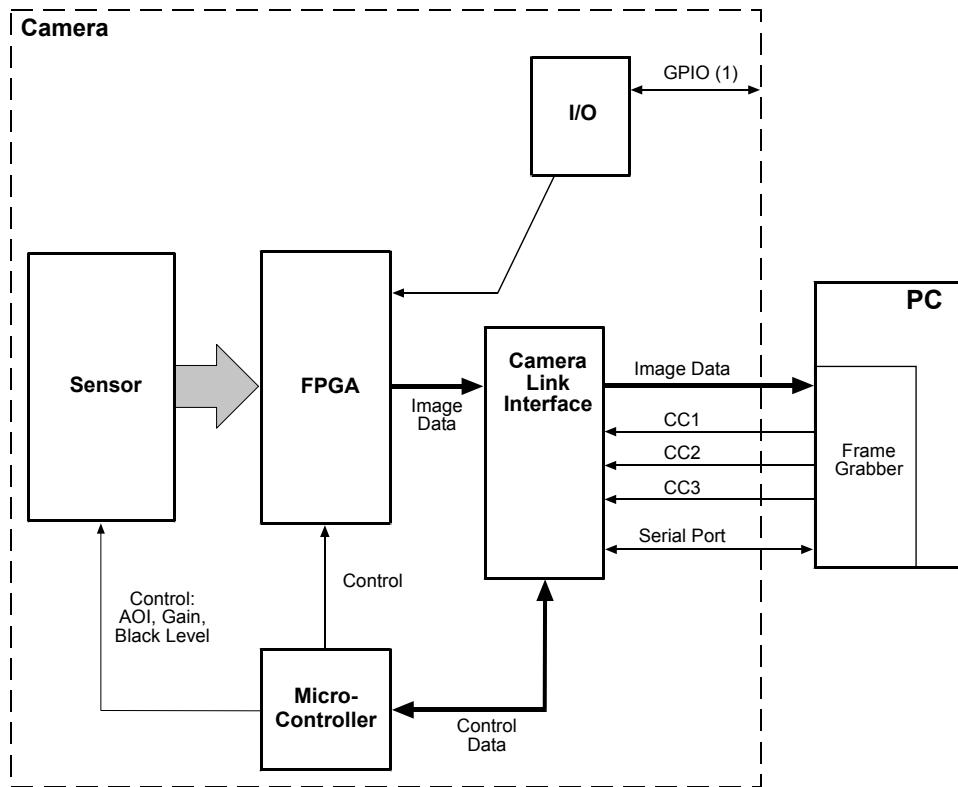


Fig. 7: Camera Block Diagram

5 Physical Interface

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

5.1 General Description of the Camera Connections

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

- Two 26-pin, 0.03" pin spacing, Shrunken Delta Ribbon (SDR) female connectors used to transmit video data, control signals, and configuration commands. (This type of connector is also known as a Mini Camera Link connector.)

These connectors can also be used to supply power to the camera in accordance with the Power over Camera Link (PoCL) specifications in the Camera Link standard.

- One 4-pin M5 receptacle used to connect to the camera's GPIO line.

This connector can also be used to provide power to the camera if PoCL is not used.

The drawing below shows the location of the three connectors.

There is also an LED indicator located on the back of the camera as shown in the drawing.

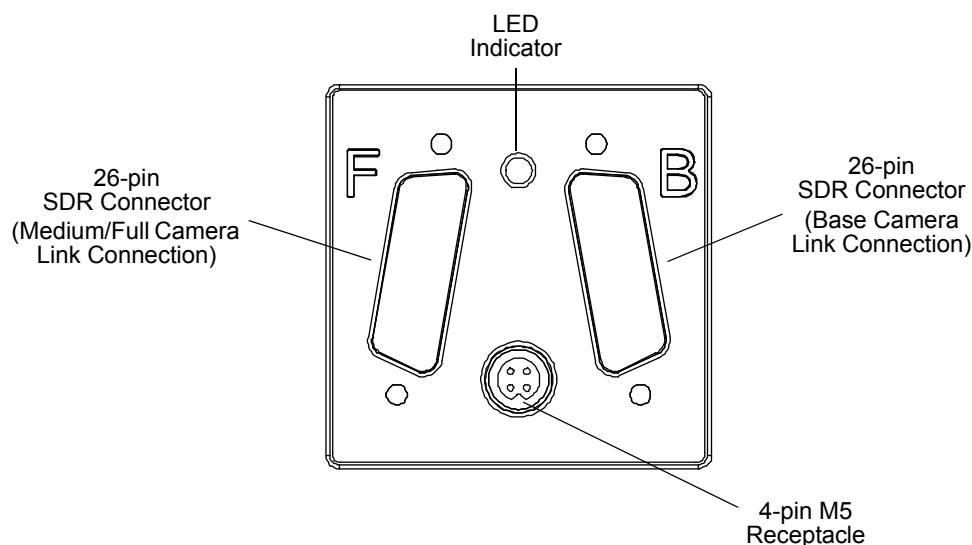


Fig. 8: Camera Connections

5.2 Camera Connector Types, Pin Assignments, and Pin Numbering

5.2.1 4-Pin Receptacle

The 4-pin connector on the camera is a Binder M5 receptacle (part number 09-3111-86-04) or the equivalent.

A recommended mating connector is the Binder M5 plug (part number 79-3108-52-04) or the equivalent. Several other mating connectors in a variety of different form factors are also available from Binder. Suitable I/O cables are available from Basler.

The 4-pin receptacle is used to supply power to the camera and to access the camera's GPIO line. The pin assignments and pin numbering for the receptacle are as shown in Table 3.

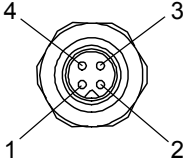

	Pin	Designation
	1	+12 VDC Camera Power (+12 VDC ± 10 %)
	2	General Purpose I/O (GPIO)
	3	Camera Power Ground
	4	GPIO Ground

Table 3: Pin Assignments and Numbering for the 4-pin Receptacle

5.2.2 26-Pin SDR Connectors

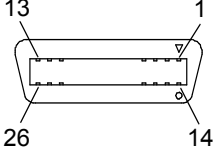
The two 26-pin connectors on the camera are female, 0.03 inch pin spacing, Shrunken Delta Ribbon (SDR) connectors as specified in the Camera Link specification.

The recommended mating connector is also defined in the Camera Link Specification.

	<p>The "SDR" (Shrunken Delta Ribbon) designation is the naming used by the 3M company. Other companies use different names for this type of connector. For example, Honda uses "HDR" as the connector name.</p>
---	---

The 26-pin connectors are used to transmit video data, control signals, and configuration commands. The pin assignments and pin numbering for the base Camera Link SDR connector are as shown in Table 4 and for the medium Camera Link SDR connector they are as shown in Table 5 on [page 26](#).

Pin Number	Signal Name	Direction	Level	Function
1, 26 *	Cam Pow.	In	+12 VDC	Camera power, +12 VDC nominal, < 1 % ripple. For more details about PoCL power requirements see the Camera Link specifications v. 2.0 and above.
13, 14 **	Power Ret.		Return	Camera power return (Gnd)
2	X0-	Output	Camera Link LVDS	Data from transmitter circuit X
15	X0+			
3	X1-	Output	Camera Link LVDS	Data from transmitter circuit X
16	X1+			
4	X2-	Output	Camera Link LVDS	Data from transmitter circuit X
17	X2+			
6	X3-	Output	Camera Link LVDS	Data from transmitter circuit X
19	X3+			
5	XClk-	Output	Camera Link LVDS	Pixel clock from transmitter circuit X
18	XClk+			
7	SerTC+	Input	RS-644 LVDS	Serial communication data receive (SerTC = "serial to camera")
20	SerTC-			
8	SerTFG-	Output	RS-644 LVDS	Serial communication data transmit (SerTFG = "serial to frame grabber")
21	SerTFG+			
9	CC1-	Input	RS-644 LVDS	Configurable
22	CC1+			
10	CC2+	Input	RS-644 LVDS	Configurable
23	CC2-			
11	CC3-	Input	RS-644 LVDS	Configurable
24	CC3+			
12	None	None	None	None
25	None-			



* Pins 1 and 26 are tied together in the camera.

** Pins 13 and 14 are tied together in the camera.

Table 4: Pin Assignments and Numbering for the Base Configuration 26-pin SDR Connector

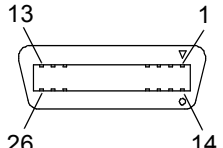
Pin Number	Signal Name	Direction	Level	Function
1, 26 *	Cam Pow.	In		10 kOhm resistor to camera power return (Gnd). Not carrying power.
13, 14 **	Power Ret.		Return	Camera power return (Gnd)
2	Y0-	Output	Camera Link LVDS	Data from transmitter circuit Y
15	Y0+			
3	Y1-	Output	Camera Link LVDS	Data from transmitter circuit Y
16	Y1+			
4	Y2-	Output	Camera Link LVDS	Data from transmitter circuit Y
17	Y2+			
6	Y3-	Output	Camera Link LVDS	Data from transmitter circuit Y
19	Y3+			
5	YClk-	Output	Camera Link LVDS	Pixel clock from transmitter circuit Y
18	YClk+			
7	T+			Connected to T- with 100R; not used
20	T-			Connected to T+ with 100R; not used
8	Z0-	Output	Camera Link LVDS	Data from transmitter circuit Z
21	Z0+			
9	Z1-	Output	Camera Link LVDS	Data from transmitter circuit Z
22	Z1+			
10	Z2-	Output	Camera Link LVDS	Data from transmitter circuit Z
23	Z2+			
12	Z3-	Output	Camera Link LVDS	Data from transmitter circuit Z
25	Z3+			
11	ZClk-	Output	Camera Link LVDS	Pixel clock from transmitter circuit Z
24	ZClk+			
		<p>* Pins 1 and 26 are tied together in the camera.</p> <p>** Pins 13 and 14 are tied together in the camera.</p>		

Table 5: Pin Assignments and Numbering for the Medium/Full Configuration 26-pin SDR Connector

5.3 Cabling Requirements

5.3.1 Camera Link Cables

The Camera Link cables must meet the Mini Camera Link cable specifications specified in Chapter 10 of the Camera Link standard. The maximum length of Camera Link cables is 10 meters. For more information on cable lengths, see the white paper "Easier than ever - What to Consider in Modern PoCL Camera Systems". You can download this white paper from the Basler website: www.baslerweb.com

5.3.2 Power and I/O Cables



You must use a power and I/O cable suitable for the camera and the connected I/O device.

- If the connected I/O device **can apply** signal voltages that meet the input voltage requirements of the camera, and if enhanced protection of the signal transmission against EMI or ESD is not required, use a **standard power and I/O cable**.
- If the connected I/O device **cannot apply** signal voltages that meet the input voltage requirements of the camera (e.g. if the I/O device's signal voltage for the "low" logic state is above 1.4 V) or if additional protection of the signal transmission against EMI or ESD is required, use a **PLC power and I/O cable**.

For more information about input voltage requirements, see Section 5.6.2 on [page 34](#).

For more information about EMI and ESD, see Section 1.6 on [page 11](#).

General Recommendations

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

- must be terminated with a suitable I/O connector. For more information, see Section 5.2 on [page 24](#).
- must be shielded and must be constructed with twisted pair wire. Use of twisted pair wire is essential to ensure that input signals are correctly received. Close proximity to strong magnetic fields should be avoided.
- must have a cross section of at least 0,14 mm², approximately equal to an American wire gauge (AWG) rating of AWG 26.

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

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



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

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

PLC Power and I/O Cables


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.


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

5.4 Camera Power

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

- Via the Camera Link cables. This is commonly known as Power over Camera Link or PoCL.
- Via the 4-pin M5 connector.

	⚠ DANGER
	<p>Electric Shock Hazard</p> <p>Non-approved power supplies may cause electric shock. Serious injury or death may occur.</p> <ul style="list-style-type: none">■ You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to<ul style="list-style-type: none">■ power supplies for I/O line power and■ power supplies for camera power.■ If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.

	⚠ WARNING
	<p>Fire Hazard</p> <p>Non-approved power supplies may cause fire and burns.</p> <ul style="list-style-type: none">■ You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to<ul style="list-style-type: none">■ power supplies for I/O line power and■ power supplies for camera power.■ If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.

5.4.1 Supplying Power Over Camera Link

Power can be supplied to the camera via the Camera Link cables as specified in Chapter 11 of the Camera Link standard. This method of supplying power to the camera is known as Power over Camera Link or PoCL.

The nominal operating voltage for the camera is +12 VDC ($\pm 10\%$) with less than one percent ripple. Power consumption is as shown in the specification tables in Section 1.2 on [page 2](#).

The power for the camera is supplied by connecting it to the frame grabber in your PC. Therefore, if you are planning to use PoCL to power the camera, your frame grabber must be PoCL-compliant.

NOTICE

If you are supplying power to the camera using PoCL:

- Failure to use a PoCL-compliant frame grabber can result in severe damage to the camera.
- Failure to use Camera Link cables that are specifically designed for PoCL (as specified in the Camera Link standard) can result in severe damage to the camera.

NOTICE

Making or breaking Camera Link connections incorrectly can cause severe damage to the camera.

If you are supplying power to the camera via the Camera Link connection (PoCL), make sure that the power to the frame grabber is switched off before you connect or disconnect the Camera Link cables.

5.4.2 Supplying Power Via the 4-Pin M5 Connector

Power can be supplied to the camera via the 4-pin M5 connector on the back of the camera. Nominal operating voltage is +12 VDC ($\pm 10\%$) with less than one percent ripple. Power consumption is as shown in the specification tables in Chapter 1 of this manual.

Close proximity to strong magnetic fields should be avoided.

See Section 5.2.1 on [page 24](#) for a description of the connector pinouts.

See Section 5.3.2 on [page 27](#) for a description of the cable to be used with the 4-pin connector. Suitable cables are available from Basler.

NOTICE

If the voltage of the power to the camera is greater than +13.2 VDC, damage to the camera can result. If the voltage is less than +10.8 VDC, the camera may operate erratically.

Applying power with the wrong polarity can result in severe damage to the camera.

- Always make sure that the voltage of the power to the camera is within the specified range.
- Always make sure that the polarity of the applied voltage is correct.
- The ace CL cameras must only be connected to other limited power sources (LPS) / Safety Extra Low Voltage (SELV) circuits.

NOTICE

An incorrect plug can damage the 4-pin connector.

The plug on the cable that you attach to the camera's 4-pin connector must have 4 female pins. Use of a plug with a different number of pins, can damage the pins in the camera's 4-pin connector.

NOTICE

Making or breaking Camera Link connections incorrectly can severely damage the camera.

If you are supplying power to the camera via the 4-pin M5 connector, switch off the power to the connector before you connect or disconnect the Camera Link cables.

5.5 LED Indicator

The LED indicator on the back of the camera signals whether power is present and also provides some basic error indications for the camera.


For more information, see Section 10.13 on [page 240](#).


5.6 General Purpose I/O (GPIO)

5.6.1 Introduction

The camera has one GPIO line that is accessed via pins 2 and 4 in the 4-pin connector on the back of the camera. The GPIO line can be set to operate as an input to the camera or to operate as a camera output. The GPIO line is designated as line 1.

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

	⚠ DANGER
	<p>Electric Shock Hazard</p> <p>Non-approved power supplies may cause electric shock. Serious injury or death may occur.</p> <ul style="list-style-type: none">■ You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to<ul style="list-style-type: none">■ power supplies for I/O line power and■ power supplies for camera power.■ If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.

	⚠ WARNING
	<p>Fire Hazard</p> <p>Non-approved power supplies may cause fire and burns.</p> <ul style="list-style-type: none">■ You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements. This applies to<ul style="list-style-type: none">■ power supplies for I/O line power and■ power supplies for camera power.■ If you're using a powered hub or powered switch as part of the connection used for data transfer, the powered hub must meet the SELV and LPS requirements.

NOTICE

The camera's GPIO line can be set to operate as an input or as an output. Applying incorrect electrical signals to the camera's GPIO line can severely damage the camera.

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

For more information about setting the GPIO line operation, see Section 6.1 on [page 42](#).

For more information about the 4-pin connector, see Section 5.2.1 on [page 24](#).

5.6.2 Operation as an Input

This section describes the electrical operation of the GPIO line when the line has been set to operate as an input.

5.6.2.1 Voltage Requirements

When the line is set to act as an input and a standard power and I/O cable is used, the following voltage requirements apply to the camera's GPIO line (pin 2 of the 4-pin connector):

Voltage	Significance
+30 VDC	Absolute maximum. The absolute maximum must never be exceeded. Otherwise, the camera may get damaged and the warranty becomes void.
+0 to +20 VDC	Safe operating I/O input voltage range.
+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.

Voltage Requirements with the GPIO Set as an Input



If a PLC power and I/O cable is used, the cable adjusts the input voltage levels (up to 24 V) to the voltage levels required by the camera. For more information, see Section 5.3.2 on [page 27](#).

5.6.2.2 Electrical Characteristics

When the GPIO line is set to operate as an input (as shown in Figure 9), the line is opto-isolated. See the previous section for input voltages and their impacts. The current draw for each input line is between 5 mA and 15 mA.

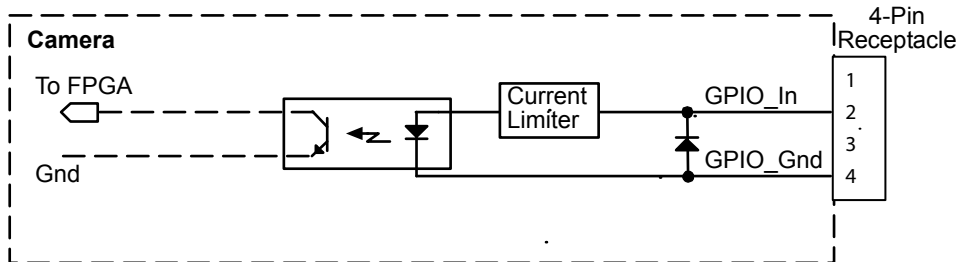


Fig. 9: Schematic with GPIO Set as an Input (Simplified)

Figure 10 shows an example of a typical circuit you can use to input a signal into the camera.

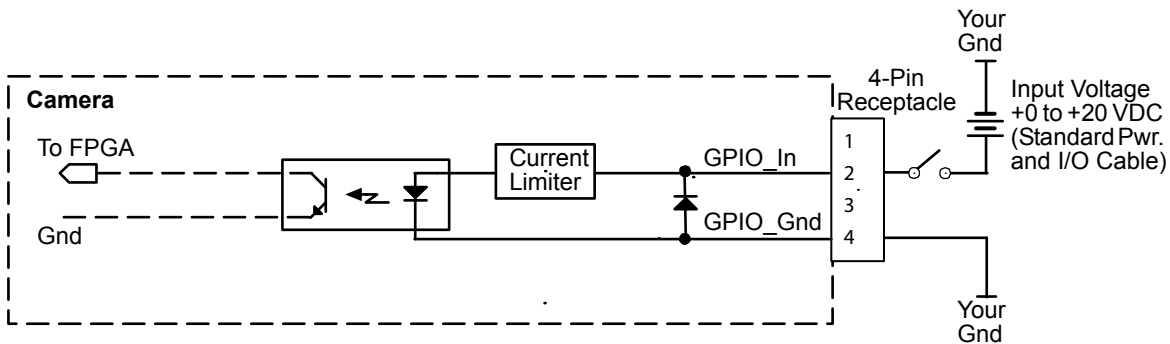


Fig. 10: Typical Application Input Circuit (Simplified)

For more information about GPIO pin assignments and pin numbering, see Section 5.2 on [page 24](#).

For more information about using the GPIO line as an input line, see Section 6.1 on [page 42](#).

5.6.2.3 Response Time



The information in this section assumes that the input circuit on your camera is designed as in the typical voltage input circuit shown in [Section 5.6.2.2](#).

The response times for the GPIO line of the camera when the line is set to operate as an input are as shown in Figure 11.

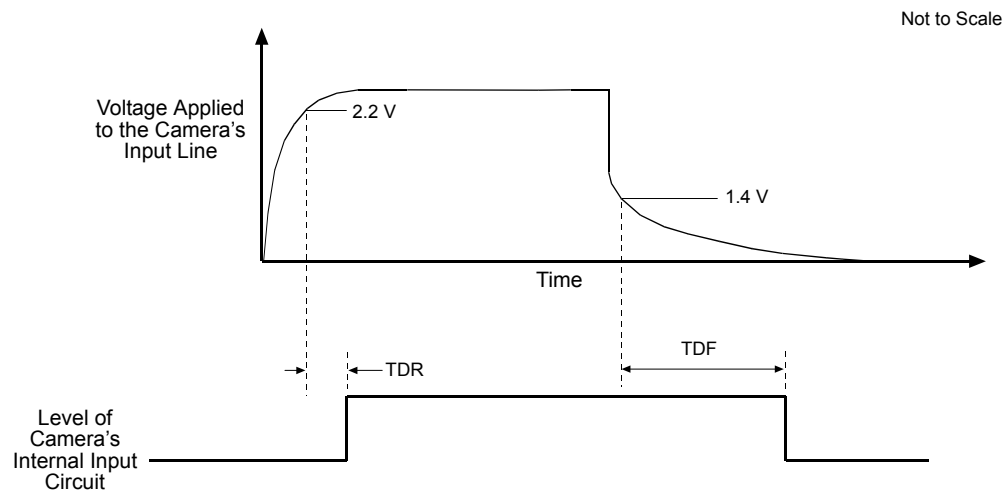


Fig. 11: GPIO Line Input Response Times

Time Delay Rise (TDR) = 1.3 μ s to 1.6 μ s (typical)

Time Delay Fall (TDF) = 40 μ s to 60 μ s (typical)



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

5.6.3 Selecting the Input as the Source Signal for a Camera Function

When the GPIO line is set to operate as an input, you can select the input to act as the source signal for the following camera functions:

- the acquisition start trigger
- the frame start trigger

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

For more information about using the input line as the source signal for a camera function, see Section 6.1.3 on [page 44](#).

5.6.4 Operation as an Output

This section describes the electrical operation of the GPIO line when the line has been set to operate as an output.

5.6.4.1 Voltage Requirements

The following voltage requirements apply to the camera's GPIO line when it is set to operate as an output:

Voltage	Significance
< +3.3 VDC	The output may operate erratically.
+3.3 to +24 VDC	Safe operating I/O output voltage range.

Table 6: Voltage Requirements with the GPIO Set as an Output

5.6.4.2 Electrical Characteristics

See the previous section for safe operating voltages. The maximum current allowed through the output circuit is 50 mA.


If the GPIO line is not set to invert:

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

If the GPIO line is set to invert:

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

For information about inverting an output line signal, see Section 6.2.8 on [page 57](#).



When the GPIO is set to operate as an output, you must use a standard power and I/O cable, **not a PLC power and I/O cable** (see Section 5.3 on [page 27](#)).

Figure 12 shows a typical circuit you can use to monitor the output line with a voltage signal.

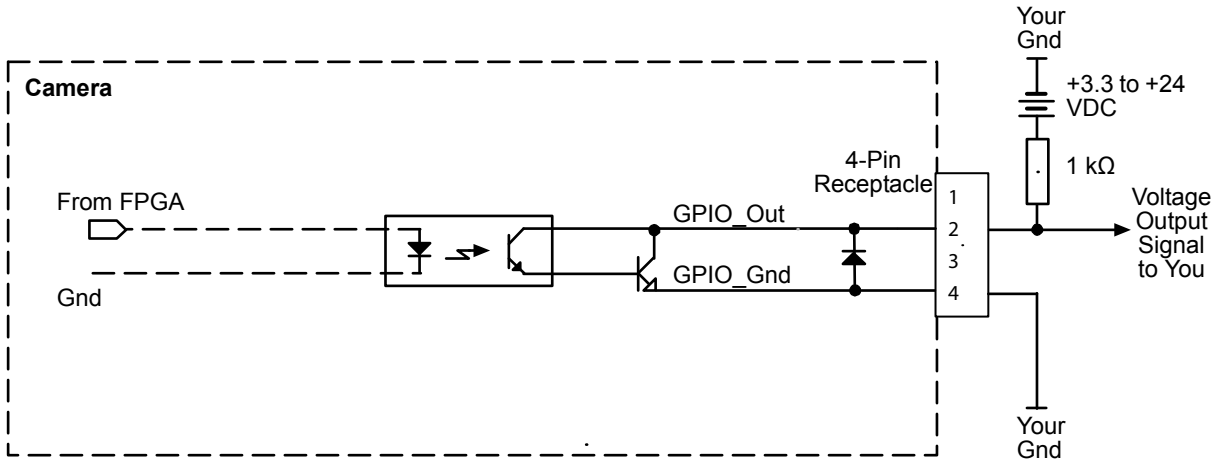


Fig. 12: Typical Voltage Output Circuit (Simplified)

Figure 13 shows a typical circuit you can use to monitor the output line with an LED or an optocoupler. In this example, the voltage for the external circuit is +24 VDC. Current in the circuit is limited by an external resistor.

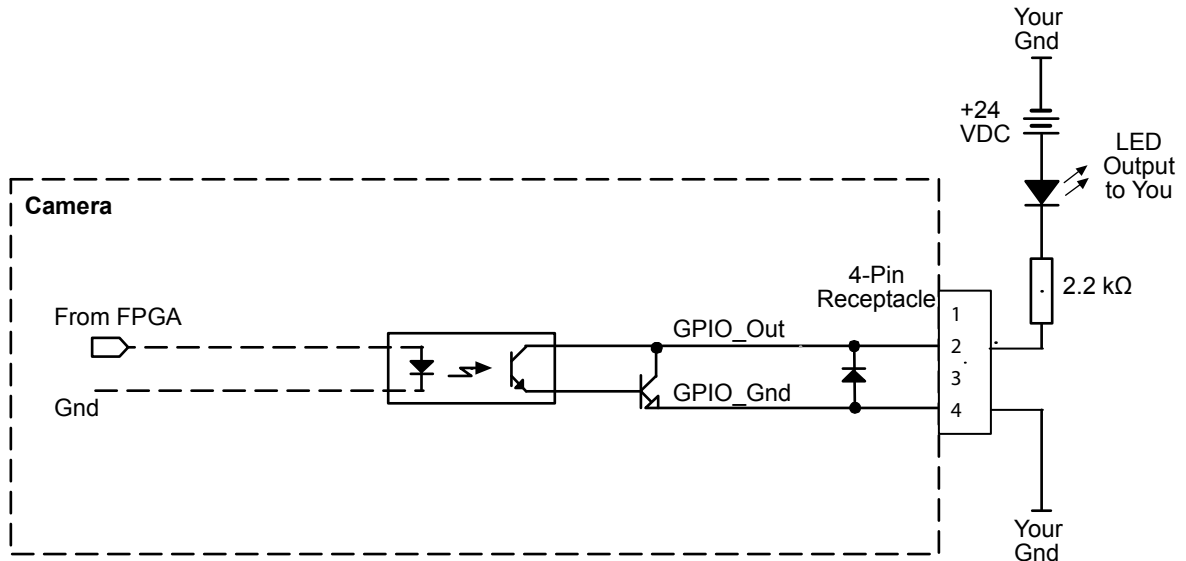


Fig. 13: Typical LED Output Signal at +24 VDC for the External Circuit (Simplified)

For more information about GPIO pin assignments and pin numbering, see Section 5.2 on [page 24](#).

For more information about using the GPIO line as an output line, see Section 6.2 on [page 48](#).

5.6.4.3 Response Time



The information in this section assumes that the output circuit on your camera is designed as in the typical voltage output circuit shown in [Section 5.6.4.2](#).

The response times for the GPIO line of the camera when the line is set to operate as an output are as shown in Figure 14.

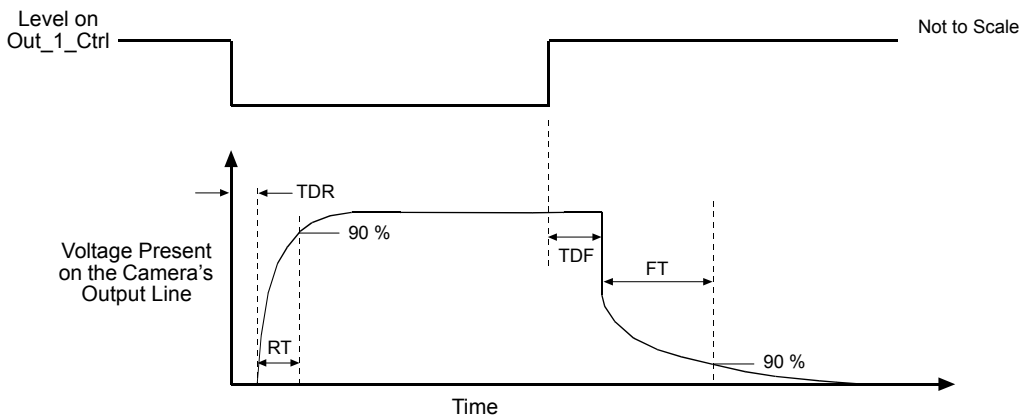


Fig. 14: GPIO Line Output Response Times

Time Delay Rise (TDR) = 40 μs (typical)

Rise Time (RT) = 20 μs to 70 μs (typical)

Time Delay Fall (TDF) = 0.6 μs (typical)

Fall Time (FT) = 0.7 μs to 1.4 μs (typical)



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

5.6.4.4 Selecting a Source Signal for the Output Line

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

For more information about selecting a source signal for the output line, see [Section 6.2.3 on page 49](#).

5.7 I/O in the Camera Link Interface

5.7.1 Inputs

The camera is also equipped with three input lines built into the Camera Link interface. These lines are designated as CC1, CC2, and CC3 as specified in the Camera Link standard. Typically, input signals are applied to these lines by the frame grabber board attached to the camera. The frame grabber board can typically be configured to supply different types of signals to these inputs as required by the camera user.

On the camera side, you can select these inputs to act as the source signal for the following camera functions:

- the acquisition start trigger
- the frame start trigger



The line logic of the CC1, CC2, and CC3 lines is negative. This means that any externally applied signal will be inverted by default. A rising edge will become a falling edge.

As a consequence, if you have set your frame start trigger signal to be "rising edge" and you apply a rising edge signal to a CC line, no image acquisition will take place as a falling edge will be sent to the camera. To counter this, you can invert the line logic using the Line Inverter parameter.

For more information about checking the line logic, see Section 6.4 on [page 61](#).

For more information about inverting the line logic, see Section 6.1.5 on [page 47](#).

For more information about using CC1, CC2, and CC3 on the specific type of frame grabber installed in your system, refer to the documentation for your frame grabber board.

You can also find some general information about how these lines are implemented in the Camera Link interface in the Basler document named *Camera Link Information for Frame Grabber Designers* (AW000990). You can download the document from the Basler website: www.baslerweb.com

For more information about using CC1, CC2, and CC3 as the source signal for a camera function, see Section 6.1 on [page 42](#).

5.7.2 Outputs

As specified in the Camera Link standard, a "CL Spare" data bit is included in the Camera Link interface. On ace Camera Link cameras, the CL Spare data bit can be used as a camera output line.

You can select any one of the camera's standard output signals to act as the source signal for the CL Spare line. The camera has six standard output signals available:

- Exposure Active
- Frame Trigger Wait
- User Output
- Acquisition Trigger Wait
- Frame Cycle
- Sync User Output

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



The CL Spare data bit is not available when the camera is used with either of the following tap geometries:

- 1X8-1Y
- 1X10-1Y

The CL Spare data bit is not directly accessible for the camera user. The data bit must be accessed via the frame grabber attached to the camera. Not all frame grabbers provide users with direct access to this bit. Consult your frame grabber documentation for more information.

For more information about using the CL Spare bit with the frame grabber installed in your system, refer to the documentation for your frame grabber board.

You can also find some general information about how this bit is implemented in the Camera Link interface in the Basler document named *Camera Link Information for Frame Grabber Designers* (AW000990). You can download the document from the Basler website: www.baslerweb.com

For more information about selecting a camera output signal to act as the source signal for the CL Spare bit, see Section 6.2 on [page 48](#).

6 I/O Control

This chapter describes how to set the camera's input and output lines. It also provides information about monitoring the state of the input and output lines.

6.1 Input Lines

6.1.1 Available Input Lines

The camera is equipped with one GPIO line designated as line 1. The GPIO line can be accessed through the four pin connector on the back of the camera as described in Chapter 5 on [page 23](#). The GPIO line can be set to operate as either an input line or an output line. This section describes using the GPIO when it is set to operate as an input line.

The camera is also equipped with three input lines built into the Camera Link interface. These lines are designated as CC1, CC2, and CC3 as specified in the Camera Link standard. Typically, input signals are applied to these lines by the frame grabber board attached to the camera. The frame grabber board can typically be configured to supply different types of signals to these inputs as required by the camera user.

For more information about using CC1, CC2, and CC3 on the specific type of frame grabber installed in your system, refer to the documentation for your frame grabber board.



The line logic of the CC1, CC2, and CC3 lines is negative. This means that any externally applied signal will be inverted by default. A rising edge will become a falling edge.

As a consequence, if you have set your frame start trigger signal to be "rising edge" and you apply a rising edge signal to a CC line, no image acquisition will take place as a falling edge will be sent to the camera. To counter this, you can invert the line logic using the Line Inverter parameter

For more information about checking the line logic, see Section 6.4 on [page 61](#).

For more information about inverting the line logic, see Section 6.1.5 on [page 47](#).

You can also get some general information about how these lines are implemented in the Camera Link interface from the Basler document called *Camera Link Information for Frame Grabber Designers* (AW000990). You can download the document from the Basler website: www.baslerweb.com

6.1.2 Setting the GPIO Line to Operate as an Input

The Line Mode parameter is used to set the camera's GPIO line to operate as an input.

NOTICE

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

1. Before you connect any external circuitry to the GPIO line, we strongly recommend that you set the GPIO line mode so that the line will operate as an input.
2. Once the line mode is set, make sure that you only apply electrical signals to the line that are appropriate for the line's current mode.

For more information about the electrical characteristics of the GPIO line, see Section 5.6 on [page 33](#).

Setting the Line Mode Using Basler pylon

You can set the Line Mode parameter value from within your application software by using the pylon API.

The following code snippet illustrates using the API to set the line mode:

```
// Configure the GPIO line as an input
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Input);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Line Mode Using Direct Register Access

To configure the GPIO line as an input via direct register access:

1. Set the value of the Line Mode Line 1 register to Input.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.1.3 Selecting an Input Line as the Source Signal for a Camera Function

You can set the GPIO line (assuming it is set as an input), CC1, CC2, or CC3 to act as the source signal for one of the following camera functions:

- Acquisition Start Trigger: If an input line is selected as the source signal for the acquisition start trigger, whenever an electrical signal that matches the selected trigger activation signal is applied to the line, the camera will recognize the signal as an acquisition start trigger signal.
- Frame Start Trigger: If an input line is selected as the source signal for the frame start trigger, whenever an electrical signal that matches the selected trigger activation signal is applied to the line, the camera will recognize the signal as a frame start trigger signal.

For detailed information about selecting an input line to act as the source signal for the acquisition start trigger and for details about how the acquisition start trigger operates, see Section 7.2.5 on [page 71](#).

For detailed information about selecting an input line to act as the source signal for the frame start trigger and for details about how the frame start trigger operates, see Section 7.3.3 on [page 82](#).

6.1.4 Input Line Debouncer

When the GPIO line is set to operate as an input, it is equipped with a debouncer feature on the line. CC1, CC2, and CC3 are also equipped with debouncers.

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 creates a delay between the arrival of a valid signal at the camera and its transfer to the camera's internal circuitry. The duration of the delay will be determined by the debouncer value.

Figure 15 illustrates how the debouncer filters out invalid input signals, i.e. signals that are shorter than the debouncer value. The diagram also illustrates how the debouncer delays a valid signal.

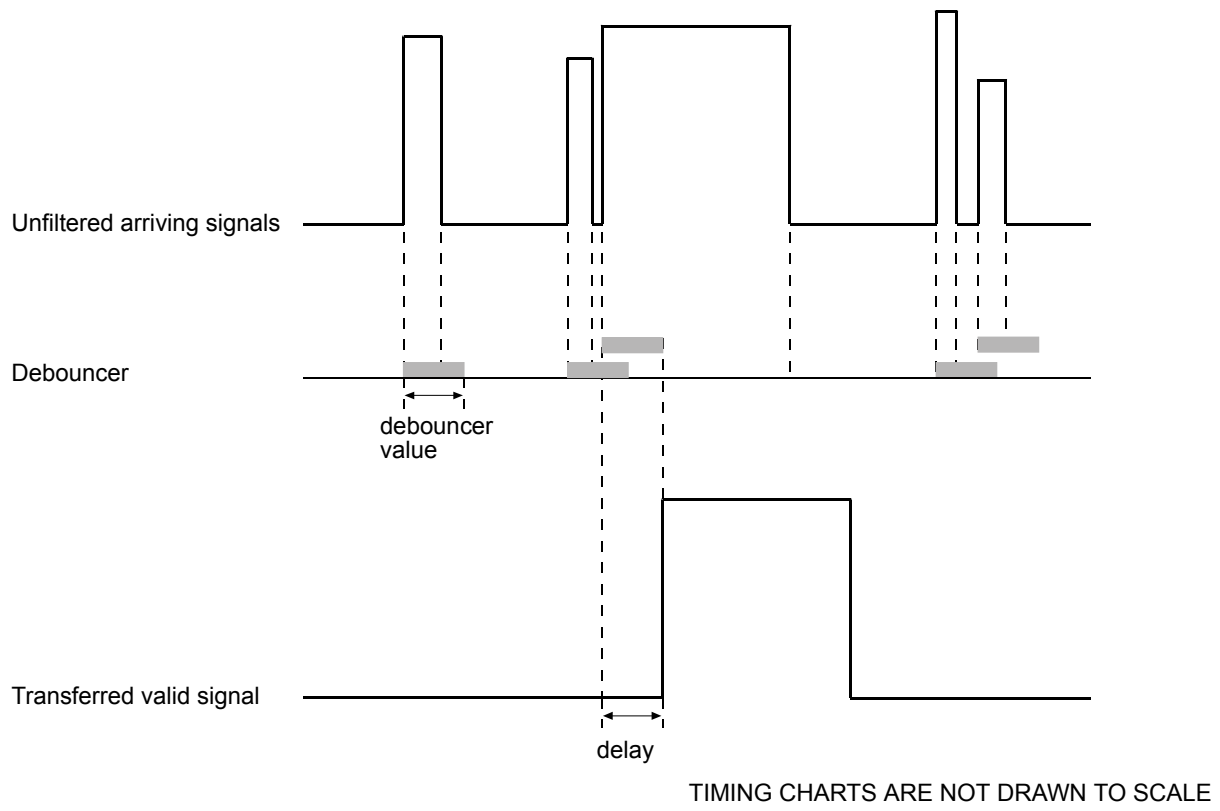


Fig. 15: Filtering of Input Signals by the Debouncer

Setting the Debouncer Using Basler pylon

The debouncer value is determined by the value of the Line Debouncer Time Abs parameter. The parameter is set in microseconds and can be set in a range from 0 to approximately 1 s.

To set the debouncer:

1. Use the Line Selector to select CC1, CC2, CC3, or Line1 (Line1 designates the GPIO line).
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 Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

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

// Set the parameter value to 100 microseconds
Camera.LineDebouncerTimeAbs.SetValue(100);

// Select the CC1 line
Camera.LineSelector.SetValue(LineSelector_CC1);

// Set the parameter value to 150 microseconds
Camera.LineDebouncerTimeAbs.SetValue(150);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Debouncer Using Direct Register Access

To set the value of the input line debouncers via direct register access:

- For the GPIO line, set the value of the Input Debouncer Time Line 1 register as desired (the value represents milliseconds).
- For the CC1 line, set the value of the Input Debouncer Time CC1 register.
- For the CC2 line, set the value of the Input Debouncer Time CC2 register.
- For the CC3 line, set the value of the Input Debouncer Time CC3 register.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.1.5 Setting an Input Line for Invert

You can set CC1, CC2, CC3, and the GPIO line to invert or not to invert the incoming electrical signal.

Setting an Input Line for Invert Using Basler pylon

To set the invert function on an input line:

1. Use the Line Selector to select CC1, CC2, CC3, or Line1 (Line1 designates the GPIO line).
2. Set the value of the Line Inverter parameter to true to enable inversion on the selected line or to false to disable inversion.

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

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

// Select the CC1 line
Camera.LineSelector.SetValue(LineSelector_CC1);
// Disable the inverter on the selected line
Camera.LineInverter.SetValue(false);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting an Input Line for Invert Using Direct Register Access

To set the invert function on an input line via direct register access:

- For the GPIO line, set the value of the Line Inverter Line 1 register to 0 (false) or 1 (true) as desired.
- For the CC1 line, set the value of the Line Inverter CC1 register.
- For the CC2 line, set the value of the Line Inverter CC2 register.
- For the CC3 line, set the value of the Line Inverter CC2 register.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.2 Output Lines

6.2.1 Available Output Lines

The camera is equipped with one GPIO line designated as line 1. The GPIO line can be accessed through the four pin connector on the back of the camera as described in Chapter 5 on [page 23](#). The GPIO line can be set to operate as either an input line or an output line. This section describes using the GPIO line when it is set to operate as an output line.

The camera is also equipped with a CL Spare data bit built into the Camera Link interface as specified in the Camera Link standard. On ace Camera Link cameras, the CL Spare data bit can be used as a camera output line.



The CL Spare data bit is not available when the camera is used with either of the following tap geometries:

- 1X8-1Y
- 1X10-1Y

The CL Spare data bit is not directly accessible for the camera user. The data bit must be accessed via the frame grabber attached to the camera. Not all frame grabbers provide users with direct access to this bit. Consult your frame grabber documentation for more information.

You can get some general information about how this bit is implemented from the Basler document called *Camera Link Information for Frame Grabber Designers* (AW000990). You can download the document from the Basler website: www.baslerweb.com

6.2.2 Setting the GPIO Line to Operate as an Output

The Line Mode parameter is used to set the camera's GPIO line to operate as an output.

NOTICE

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

1. Before you connect any external circuitry to the GPIO line, we strongly recommend that you set the GPIO line mode so that the line will operate as an output.
2. Once the line mode is set, make sure that you only apply electrical signals to the line that are appropriate for the line's current mode.

For more information about the electrical characteristics of the GPIO line, see Section 5.6 on [page 33](#).

Setting the Line Mode Using Basler pylon

You can set the Line Mode parameter value from within your application software by using the pylon API.

The following code snippet illustrates using the API to set the line mode:

```
// Configure the GPIO line as an output
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Output);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Line Mode Using Direct Register Access

To configure the GPIO line as an output via direct register access:

1. Set the value of the Line Mode Line 1 register to Output.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.2.3 Selecting a Source Signal for an Output Line

You can select any one of the camera's standard output signals to act as the source signal for the GPIO line (assuming it is set as an output) or for the CL Spare line. The camera has six standard output signals available:

- Exposure Active
- Frame Trigger Wait
- User Output
- Acquisition Trigger Wait
- Frame Cycle
- Sync User Output

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

For more information about the acquisition trigger and frame trigger wait signals, see Section 7.1 on [page 62](#).

For more information about the exposure active signal, see Section 7.6.1 on [page 98](#).

For more information about the frame cycle signal, see Section 7.6.4 on [page 111](#).

For more information about setting the state of a user settable output line, see Section 6.2.4 on [page 51](#).

For more information about the sync user output signal, see Section 6.2.6 on [page 53](#).

For more information about the electrical characteristics of the GPIO line when it is set as an output, see Section 5.6.4 on [page 37](#).

Selecting a Source Signal Using Basler pylon

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

1. Use the Line Selector to select the line.
2. Set the value of the Line Source parameter to one of the available output signals or to user settable. This will set the source signal for the output line.

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

```
// Select the GPIO line
Camera.LineSelector.SetValue(LineSelector_Line1);
// Set the source signal for the selected line
Camera.LineSource.SetValue(LineSource_ExposureActive);

// Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_CLSpare);
// Set the source signal for the selected line
Camera.LineSource.SetValue(LineSource_UserOutput);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Selecting a Source Signal Using Direct Register Access

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

- For the GPIO line, set the value of the Line Source Line 1 register to Acquisition Trigger Wait, Frame Trigger Wait, Exposure Active, Frame Cycle, User, or Sync User Output as desired.
- For the CL Spare line, set the value of the Line Source Line CL Spare register to Acquisition Trigger Wait, Frame Trigger Wait, Exposure Active, Frame Cycle, User, or Sync User Output as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.2.4 Setting the State of a User Settable Output Line

As mentioned in the previous section, the GPIO line (assuming it is set as an output) and the CL Spare line can be designated as "user settable" which means that you can assign a state (high or low) to them using the User Output Value parameter.

You can use this to control external events or devices, e.g. a light source. Note that this is a non-sequence parameter and therefore its values can't be changed using the sequencer feature. If you want to achieve the same result with the sequencer, you need to set the Sync User Output Value parameter instead (see Section 6.2.6 on [page 53](#)).

For more information about the sequencer feature, see Section 10.6 on [page 165](#).

Setting the State Using Basler pylon

To set the state of a user settable output line:

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

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

```
// Set the GPIO line to be user settable
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineSource.SetValue(LineSource_UserOutput);
// Select the user settable line and set the state
Camera.UserOutputSelector.SetValue(UserOutputSelector_UserOutputLine1);
Camera.UserOutputValue.SetValue(true);

// Set the CL Spare line to be user settable
Camera.LineSelector.SetValue(LineSelector_ClSpare);
Camera.LineSource.SetValue(LineSource_UserOutput);
// Select the user settable line and set the state
Camera.UserOutputSelector.SetValue(UserOutputSelector_UserOutputClSpare);
Camera.UserOutputValue.SetValue(true);
```

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



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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the State Using Direct Register Access

To set the state of a user settable output line via direct register access:

- For the GPIO line, set the value of the User Output Line 1 register to 1 (true) or 0 (false) as required.
- For the CL Spare line, set the value of the User Output CL Spare register to 1 (true) or 0 (false) as required.

For more information about direct register access, see Section 3.2 on [page 20](#).

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

The User Output Value All parameter can be used to set and check the current state of all user settable output lines. The parameter value is expressed as a hexadecimal number by the Basler pylon Viewer and as a 32-bit word in the Basler pylon API. The number is derived from bits 0 and 12 of a 32-bit word and is based on all current binary User Output Value parameter values.

As shown in Figure 16, bit 0 is associated with Line 1 and bit 12 is associated with CL spare. The states of those lines are expressed by the related binary User Output Value parameter values.

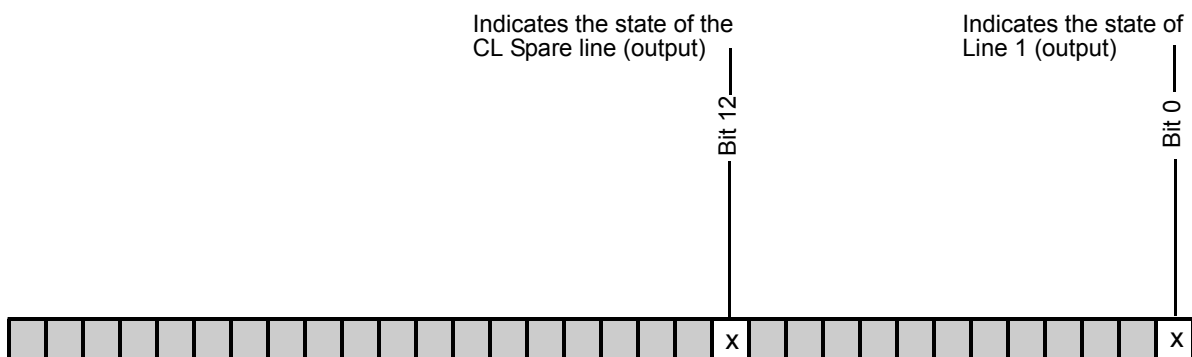


Fig. 16: Bit Field of the User Output Value All Parameter: Bit Numbers and Assignment of Output Lines

To determine all current User Output Value parameter values in a single step, check the hexadecimal number of the User Output Value All parameter value. This contains the current states of all user settable lines.

For example, if a User Output Value All parameter value of 0x1000 is reported while all line inverters are disabled, this can be translated into the following states:

- the User Output Value parameter value for Line 1 is currently 0, indicating that the line state is currently low and
- the User Output Value parameter value for the CL Spare line is currently 1, indicating that the line state is currently high.

Setting and Checking the State Using Basler pylon

To set the state of all user outputs in a single step:

1. Set the value of the User Output Value All parameter to set all user output values.

To check the state of all user outputs in a single step:

1. Read the value of the User Output Value All parameter to determine the current settings of all user output values.

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

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

You can also use the Basler pylon Viewer application to easily set the parameters. For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting and Checking the State Using Direct Register Access

To set the state of all user outputs in a single step:

1. Set the value of the User Output Value All register as required.

To check the state of all user outputs in a single step:

1. Read the value of the User Output Value All register to determine the current settings of all user output values.

6.2.6 Setting the State of a User Settable Synchronous Output Signal

User settable output lines, the GPIO line and the CL Spare line, can be set to supply output signals synchronous to the frame start trigger in a sequence set. This works in a similar fashion to the User Output Value parameter described in Section 6.2.4 on [page 51](#) with the difference that these parameter value can be changed with the sequencer feature.

Using the synchronous output signal, you can control external events and devices, e.g. a light source, when you're using the sequencer feature. To do this, you save the state of an output line, high or low, in a sequence set, so when a frame start trigger is received and the sequencer

advances from one set to the next, the output signal is either high or low. This can be used to turn a lamp on or off to account for different lighting requirements depending on the sequence set for example. Put in other words, the turning on or off of a lamp is synchronized with the frame start trigger of the sequences.

For more information about the sequencer feature, see Section 10.6 on [page 165](#).

Setting the State Using Basler pylon

To set the state of a synchronous output signal:

1. Use the Sync User Output Selector to select the GPIO line or the CL Spare line.
2. Set the value of the Sync User Output Value parameter to true (1) or false (0) as required.

You can set the Sync User Output Selector and the Sync User Output Value parameter from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the state of the synchronous output signal:

```
// Set the GPIO line to provide a synchronous user settable output signal
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineSource.SetValue(LineSource_SyncUserOutput);
// Select the user settable line and set the state of the synchronous user output
signal
camera.SyncUserOutputSelector.SetValue(SyncUserOutputSelector_SyncUserOutputLine
1);
camera.SyncUserOutputValue.SetValue(true);

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

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



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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the State Using Direct Register Access

To set the state of a synchronous user output signal:

- For the GPIO line, set the value of the Sync User Output Line 1 register to 1 (true) or 0 (false) as required.
- For the CL Spare line, set the value of the Sync User Output CL Spare register to 1 (true) or 0 (false) as required.

For more information about direct register access, see Section 3.2 on [page 20](#).

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

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

As shown in Figure 17, bit 0 is associated with Line 1 and bit 12 is associated with CL Spare. The states of those lines are expressed by the related binary Sync User Output Value parameter values.

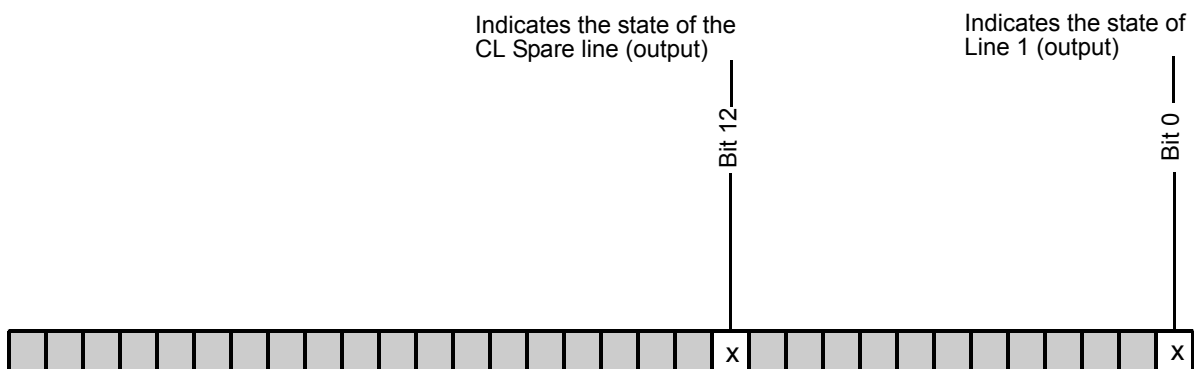


Fig. 17: Bit Field of the Sync User Output Value All Parameter: Bit Numbers and Assignment of Output Lines

To determine all current Sync User Output Value parameter values in a single step, check the hexadecimal number of the Sync User Output Value All parameter value. This contains the current state of all user settable synchronous output signals.

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

- the Sync User Output Value parameter value for Line 1 is currently 1, indicating that the signal state is currently high and
- the Sync User Output Value parameter value for the CL Spare line is currently 0, indicating that the signal line state is currently low.

Setting and Checking the State Using Basler pylon

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

1. Set the value of the Sync User Output Value All parameter to set all synchronous output signals.

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

1. Read the value of the Sync User Output Value All parameter to determine the current settings of all synchronous output signals.

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

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

You can also use the Basler pylon Viewer application to easily set the parameters. For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting and Checking the State Using Direct Register Access

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

1. Set the value of the Sync User Output Value All register as required.

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

1. Read the value of the Sync User Output Value All register to determine the current settings of all synchronous output signal values.

6.2.8 Setting an Output Line for Invert

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

GPIO Line

If the GPIO line is not set to invert:

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

If the GPIO line is set to invert:

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

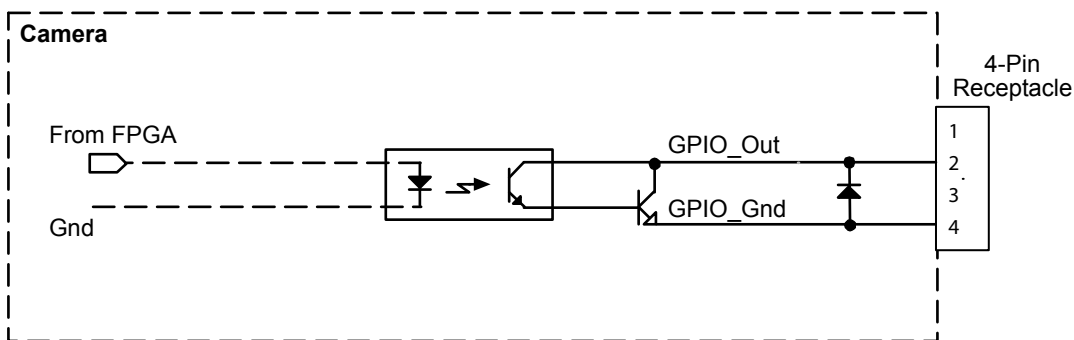


Fig. 18: GPIO Line Schematic (GPIO Line Set as an Output)

CL Spare Line

If the CL Spare line is not set to invert:

- When the source camera signal assigned to the line is high, the CL Spare line will transmit a logical 1. And when the source signal is low, the CL Spare line will transmit a logical 0.

If the CL Spare line is set to invert:

- When the source camera signal assigned to the line is high, the CL Spare line will transmit a logical 0. And when the source signal is low, the CL Spare line will transmit a logical 1.

Setting an Output Line for Invert Using pylon

To set the invert function on an output line:

1. Use the Line Selector to select the GPIO line or the CL Spare line.
2. Set the value of the Line Inverter parameter to true to enable inversion on the selected line or to false to disable inversion.

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

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

// Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_CLSpare);
// Disable the inverter on the selected line
Camera.LineInverter.SetValue(false);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting an Output Line for Invert Using Direct Register Access

To set the invert function on an output line via direct register access:

- For the GPIO line, set the value of the Line Inverter Line 1 register to 0 (false) or 1 (true) as desired.
- For the CL Spare line, set the value of the Line Inverter CL Spare register to 0 (false) or 1 (true) as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.3 Checking the State of the I/O Lines

6.3.1 Checking the State of a Single Line

Checking the State Using Basler pylon

To check the current state of an I/O line:

1. Use the Line Selector parameter to select a line.
2. Read the value of the Line Status parameter to determine the current state of the line.
A value of true means the line's state is currently high and a value of false means the line's state is currently low.

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

```
// Select the GPIO line and read the state
Camera.LineSelector.SetValue(LineSelector_Line1);
bool Line1State = Camera.LineStatus.GetValue();

// Select the CLSpare line and read the state
Camera.LineSelector.SetValue(LineSelector_CLSpare);
bool CLSpareState = Camera.LineStatus.GetValue();

// Select the CC1 line and read the state
Camera.LineSelector.SetValue(LineSelector_CC1);
bool CC1 = Camera.LineStatus.GetValue();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Checking the State Using Direct Register Access

To check the current state of an I/O line:

- For the GPIO line, read the value of the Line Status Line 1 register. The value will indicate 1 (true) or 0 (false).
- For the CL Spare line, read the value of the Line Status CL Spare register.
- For the CC1 line, read the value of the Line Status CC1 register.
- For the CC2 line, read the value of the Line Status CC2 register.
- For the CC3 line, read the value of the Line Status CC3 register.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.3.2 Checking the State of All Lines

Checking the State Using Basler pylon

You can determine the current state of all input and output lines by reading 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 19, certain bits in the value are associated with each I/O line, and each of these bits will indicate the state of the associated line. If a bit is 0, it indicates that the state of the associated line is currently low. If a bit is 1, it indicates that the state of the associated line is currently high.

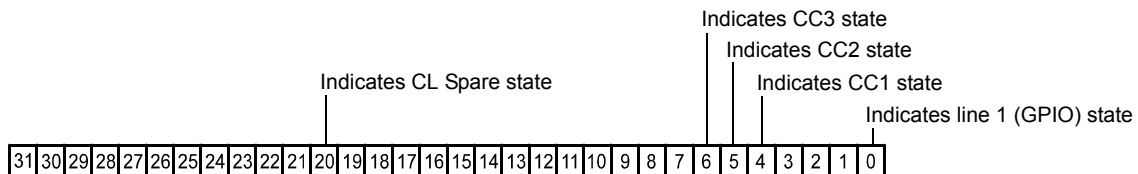


Fig. 19: Line Status All Parameter Bits

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Checking the State Using Direct Register Access

To check the current state of all I/O lines via direct register access, read the value of the Line Status All register. The register holds a 32-bit value that indicates the state of each I/O line. The mapping of the bits in the value to I/O lines is identical to the mapping described above when using Basler pylon to check the state.

For more information about direct register access, see Section 3.2 on [page 20](#).

6.4 Checking the Line Logic

Checking the Line Logic Using Basler pylon

To check the line logic of an I/O line:

1. Use the Line Selector parameter to select a line.
2. Read the value of the Line Logic parameter.

The parameter will indicate whether the logic is positive or negative.

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

```
// Select the GPIO line and read the line logic type
Camera.LineSelector.SetValue(LineSelector_Line1);
LineLogicEnums lineLogicLine1 = Camera.LineLogic.GetValue();

// Select the CLSpare line and read line logic type
Camera.LineSelector.SetValue(LineSelector_CLSpare);
LineLogicEnums lineLogicCLSpare = Camera.LineLogic.GetValue();

// Select the CC1 line and read the line logic type
Camera.LineSelector.SetValue(LineSelector_CC1);
LineLogicEnums lineLogicCC1 = Camera.LineLogic.GetValue();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Checking the Line Logic Using Direct Register Access

To check the line logic of an I/O line:

- For the GPIO line, read the value of the Line Logic Line 1 register. The value will indicate 1 (positive) or 0 (negative).
- For the CL Spare line, read the value of the Line Logic CL Spare register.
- For the CC1 line, read the value of the Line Logic CC1 register.
- For the CC2 line, read the value of the Line Logic CC2 register.
- For the CC3 line, read the value of the Line Logic CC3 register.

For more information about direct register access, see Section 3.2 on [page 20](#).

7 Image Acquisition Control

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

7.1 Overview

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

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

- The acquisition start trigger
- The frame start trigger
- Exposure time control

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

When reading the material in this chapter, it is helpful to refer to Figure 20 on [page 64](#) and to the use case diagrams in Section 7.9 on [page 119](#). These diagrams present the material related to the acquisition start trigger and the frame start trigger in a graphical format.

Acquisition Start Trigger

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

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

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

If the Trigger Mode parameter for the acquisition start trigger is set to on, the initial acquisition status of the camera will be "waiting for acquisition start trigger" (see Figure 20 on [page 64](#)). When the camera is in this acquisition status, it can't react to frame start trigger signals. When an acquisition start trigger signal is applied to the camera, the camera will exit the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status. The camera can then react to frame start trigger signals. The camera will continue to react to frame start trigger signals until the number of frame start trigger signals it has received is equal to an integer parameter setting called the Acquisition Frame Count. At that point, the camera will return to the "waiting for acquisition start trigger" acquisition status and will remain in that status until a new acquisition start trigger signal is applied.

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

Frame Start Trigger

Assuming that an acquisition start trigger signal has just been applied to the camera, the camera will exit from the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status. Applying a frame start trigger signal to the camera at this point will exit the camera from the "waiting for frame start trigger" acquisition status and will begin the process of exposing and reading out a frame (see Figure 20 on [page 64](#)). As soon as the camera is ready to accept another frame start trigger signal, it will return to the "waiting for frame start trigger" acquisition status. A new frame start trigger signal can then be applied to the camera to begin another frame exposure.

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

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

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

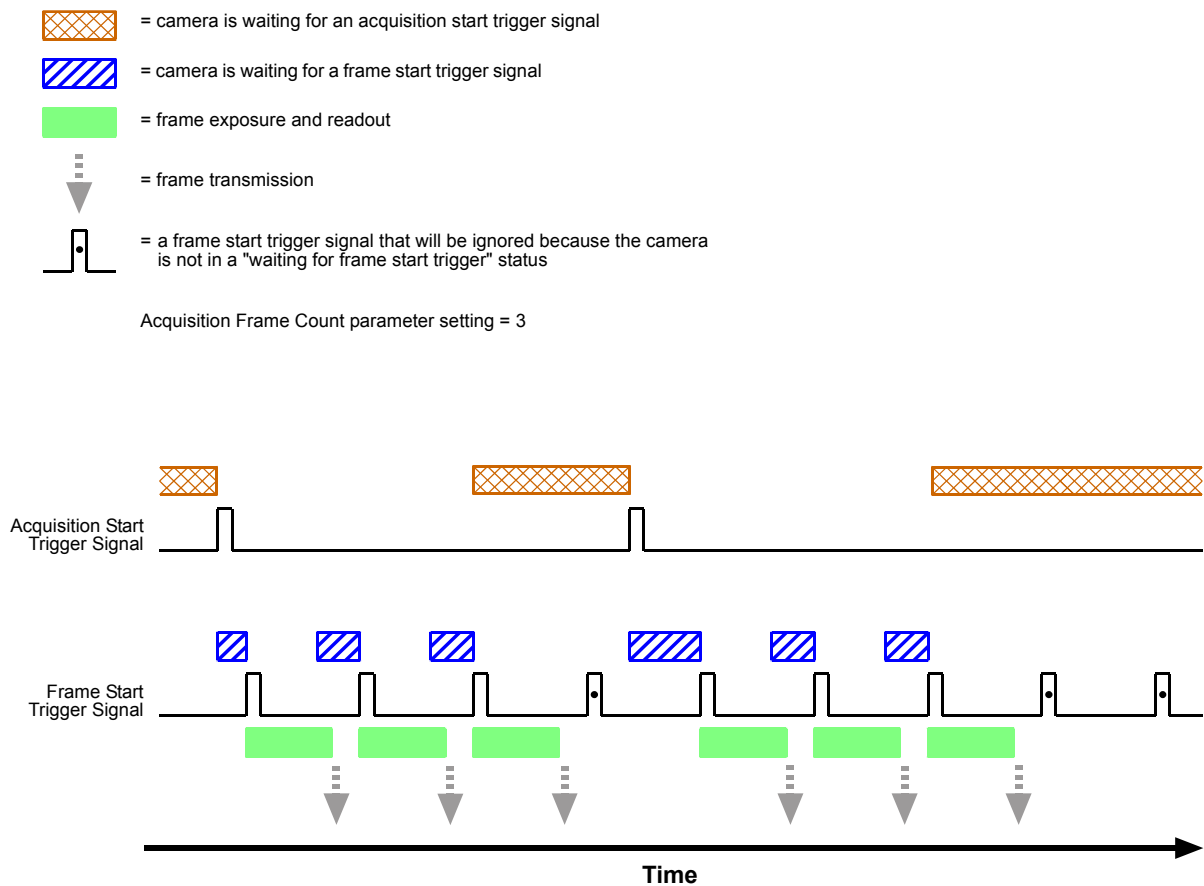


Fig. 20: Acquisition Start and Frame Start Triggering

Applying Trigger Signals

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

To apply trigger signals via software:

1. Select the acquisition start or the frame start trigger.
2. Select software as the source for the trigger signal.

At that point, the Basler pylon API or direct camera register access can be used to apply a software trigger signal to the camera for the selected trigger.

To apply trigger signals via hardware:

1. Select the acquisition start or the frame start trigger.
2. Select the camera's GPIO line or one of the CC lines as the source for the trigger signal.
3. Select a trigger activation signal.

After having made these settings, each time an electrical signal that matches the selected trigger activation signal is applied to the selected input, it will be recognized by the camera.

The Trigger Selector (This Only Applies When You Are Using Basler pylon)

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

Exposure Time Control

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

If the camera is set for software frame start triggering, the camera's Exposure Time parameter will determine the exposure time for each frame.

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



An exposure time offset exists in all CMOSIS V2 sensors. This means that the exposure of the sensor is always longer than specified in the camera's settings. For more information and advice how to counter this effect, see "Exposure Time Offset in CMOSIS V2 Sensors" on [page 89](#).

7.2 The Acquisition Start Trigger

(When reading this section, it is helpful to refer to Figure 20 on [page 64](#).)

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

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

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

7.2.1 Acquisition Start Trigger Mode

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

7.2.1.1 Acquisition Start Trigger Mode = Off

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

7.2.1.2 Acquisition Start Trigger Mode = On

When the Trigger Mode parameter for the acquisition start trigger is set to on, the camera will initially be in a "waiting for acquisition start trigger" acquisition status and can't react to frame start trigger signals. You must apply an acquisition start trigger signal to the camera to exit the camera from the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status. The camera can then react to frame start trigger signals and will continue to do so until the number of frame start trigger signals it has received is equal to the current Acquisition

Frame Count parameter setting. The camera will then return to the "waiting for acquisition start trigger" acquisition status. In order to acquire more frames, you must apply a new acquisition start trigger signal to the camera for it to exit the "waiting for acquisition start trigger" acquisition status.

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

- Software: When the source signal is set to software, you apply an acquisition start trigger signal to the camera by executing a Trigger Software command for the acquisition start trigger on the host PC.
- Line 1: When the source signal is set to line 1, you apply an acquisition start trigger signal to the camera by injecting an externally generated electrical signal (commonly referred to as a hardware trigger signal) into the GPIO line of the camera. This assumes that the GPIO line of the camera has been set to operate as an input line.
- CC1: When the source signal is set to CC1, you apply an acquisition start trigger signal to the camera by injecting an externally generated electrical signal into CC1 in the Camera Link interface.
- CC2: When the source signal is set to CC2, you apply an acquisition start trigger signal to the camera by injecting an externally generated electrical signal into CC2 in the Camera Link interface.
- CC3: When the source signal is set to CC3, you apply an acquisition start trigger signal to the camera by injecting an externally generated electrical signal into CC3 in the Camera Link interface.

If the Trigger Source parameter for the acquisition start trigger is set to line 1, CC1, CC2, or CC3, you must also set the Trigger Activation parameter. The options available for this parameter are:

- Rising Edge: Specifies that a rising edge of the hardware trigger signal will act as the acquisition start trigger.
- Falling Edge: Specifies that a falling edge of the hardware trigger signal will act as the acquisition start trigger.



Typically, a frame grabber is used to supply the electrical frame start trigger signal to CC1, CC2, or CC3.



The CC lines have a negative line logic. For more information, see the note on [page 42](#).

For more detailed information about the GPIO line of the camera, see Section 5.6 on [page 33](#).

For more detailed information about the CC1, CC2, and CC3 inputs in the Camera Link interface, refer to the document called *Camera Link Information for Frame Grabber Designers* (AW000990). You can download the document from the Basler website: www.baslerweb.com

7.2.2 Acquisition Frame Count

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

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

7.2.3 Setting The Acquisition Start Trigger Mode and Related Parameters

Setting the Parameters Using Basler pylon

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

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

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

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

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Configure the GPIO line as an input
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Input);
```

```
// Set the source for the selected trigger to line 1
Camera.TriggerSource.SetValue (TriggerSource_Line1);
// Set the activation mode for the selected trigger to rising edge
Camera.TriggerActivation.SetValue (TriggerActivation_RisingEdge);
// Set the acquisition frame count
Camera.AcquisitionFrameCount.SetValue (5);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

Setting the Parameters Using Direct Register Access

To set the parameters related to the acquisition start trigger:

1. Set the value of the Trigger Mode Acquisition Start register to On.
2. Set the value of the Trigger Source Acquisition Start register to Software, Line 1, CC1, CC2, or CC3.
 - a. If the trigger source is set to line 1, set the GPIO line to operate as an input by setting the value of the LineModeLine1 register to Input.
 - b. If the trigger source is set to line 1, CC1, CC2, or CC3, set the value of the Trigger Activation Acquisition Start register to Rising Edge or Falling Edge.
3. Set the value of the Acquisition Frame Count register as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.2.4 Using a Software Acquisition Start Trigger Signal

7.2.4.1 Introduction

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

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

Section 7.2.4.2 includes more detailed information about applying a software acquisition start trigger to the camera using Basler pylon or via direct register access.

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

Setting the Parameters and Applying the Signal Using Basler pylon

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

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Set the source for the selected trigger
Camera.TriggerSource.SetValue (TriggerSource_Software);
// Set the acquisition frame count
Camera.AcquisitionFrameCount.SetValue(5);
// Execute a trigger software command to apply a software acquisition start trigger
// signal to the camera
Camera.TriggerSoftware.Execute();

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

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Parameters and Applying the Signal Using Direct Register Access

To set the parameters for software acquisition start triggering:

1. Set the value of the Trigger Mode Acquisition Start register to On.
2. Set the value of the Trigger Source Acquisition Start register to Software.
3. Set the value of the Acquisition Frame Count register as desired.
4. Set the value of the Trigger Software Acquisition Start register to 1.

Setting the value of this register to 1 applies a software acquisition start trigger signal to the camera. The register resets to 0 when execution is complete.

For more information about direct register access, see Section 3.2 on [page 20](#).

7.2.5 Using a Hardware Acquisition Start Trigger Signal

7.2.5.1 Introduction

If the Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Source parameter is set to line 1, CC1, CC2, or CC3, an externally generated electrical signal injected into the selected source will act as the acquisition start trigger signal for the camera. This type of trigger signal is generally referred to as a hardware trigger signal or as an external acquisition start trigger signal (ExASTrig).

If the Trigger Source is set to line 1, the GPIO line must be properly set to operate as an input in order to accept the trigger signal.

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

When the Trigger Mode parameter is set to on, the camera will initially be in a "waiting for acquisition start trigger" acquisition status. It can't react to frame start trigger signals when in this acquisition status. When the appropriate ExASTrig signal is applied to the selected source, the camera will exit the "waiting for acquisition start trigger" acquisition status and will enter the "waiting for frame start trigger" acquisition status. It can then react to frame start trigger signals. When the number of frame start trigger signals received by the camera is equal to the current Acquisition Frame Count parameter setting, the camera will return to the "waiting for acquisition start trigger" acquisition status. When a new ExASTrig signal is applied to the selected source, the camera will again exit from the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status.

For more information about setting the camera for hardware acquisition start triggering and selecting the source to receive the ExASTrig signal, see Section 7.2.5.3 on [page 72](#).

For more information about the electrical requirements for the GPIO line when it is set to operate as an input, see Section 5.6.2 on [page 34](#).

For more information about CC1, CC2, and CC3, see Section 6.1 on [page 42](#).

7.2.5.2 Acquisition Start Trigger Delay

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

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

If you are parameterizing the camera with Basler pylon, the Trigger Delay Abs parameter will determine the length of the delay.

If you are parameterizing the camera via direct register access, the Trigger Delay Raw Acquisition Start register will determine the length of the delay.



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

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

Setting the Parameters Using Basler pylon and Applying a Signal

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

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Configure the GPIO line as an input
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Input);
// Set the source for the selected trigger to line 1
Camera.TriggerSource.SetValue(TriggerSource_Line1);
```

```
// Set the activation mode for the selected trigger to rising edge
Camera.TriggerActivation.SetValue(TriggerActivation_RisingEdge);
// Set the acquisition frame count
Camera.AcquisitionFrameCount.SetValue(5);

// Apply a rising edge of the externally generated electrical signal
// (ExASTrig signal) to line 1 on the camera
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

Setting the Parameters Using Direct Register Access and Applying a Signal

To set the parameters for hardware acquisition start triggering:

1. Set the value of the Trigger Mode Acquisition Start register to On.
2. Set the value of the Trigger Source Acquisition Start register to Line 1, CC1, CC2, or CC3.
 - a. If the trigger source is set to line 1, set the GPIO line to operate as an input by setting the value of the LineModeLine1 register to Input.
3. Set the value of the Trigger Activation Acquisition Start register to Rising Edge or Falling Edge.
4. Set the value of the Acquisition Frame Count register as desired.
5. Apply the appropriate externally generated electrical signal (ExASTrig signal) to the selected trigger source.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.3 The Frame Start Trigger

The frame start trigger is used to begin frame acquisition.

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

(For a quick overview of acquisition start triggering and frame start triggering, see Section 7.1 on [page 62](#).)

When reading this section, it is helpful to refer to Figure 20 on [page 64](#) and the use case diagrams that appear in Section 7.9 on [page 119](#).

7.3.1 Frame Start Trigger Mode

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

7.3.1.1 Frame Start Trigger Mode = Off

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

If you are using Basler pylon to parameterize the camera, the rate at which the frame start trigger signals will be generated is determined by the camera's Acquisition Frame Rate Abs parameter:

- If the parameter is not enabled, the camera will generate frame start trigger signals at the maximum rate allowed given the current camera settings.
- If the parameter is enabled and is set to a value less than the maximum allowed frame rate, the camera will generate frame start trigger signals at the rate specified by the parameter setting.
- If the parameter is enabled and is set to a value greater than the maximum allowed frame rate, the camera will generate frame start trigger signals at the maximum allowed frame rate.

If you are using direct register access to parameterize the camera, the rate at which the frame start trigger signals will be generated is determined by the camera's Acquisition Frame Period Raw parameter (Frame Rate = 1/ Frame Period):

- If the parameter is not enabled, the camera will generate frame start trigger signals at the maximum rate allowed given the current camera settings.
- If the parameter is enabled and is set to a value that would result in a frame rate less than the maximum allowed, the camera will generate frame start trigger signals at the rate that results from the parameter setting.
- If the parameter is enabled and is set to a value that would result in a frame rate greater than the maximum allowed given the current camera settings, the camera will generate frame start trigger signals at the maximum allowed frame rate.

Exposure Time Control with the Frame Start Trigger Off

When the Trigger Mode parameter for the frame start trigger is set to off, the exposure time for each frame acquisition is determined by:

- the value of the camera's Exposure Time Abs parameter if you are parameterizing the camera using Basler pylon.
- the value in the camera's Exposure Time Raw register if you are parameterizing the camera using direct register access.

For more information about setting the Exposure Time Abs parameter or the Exposure Time Raw register, see Section 7.4 on [page 89](#).

7.3.1.2 Frame Start Trigger Mode = On

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



Do not trigger frame acquisitions at a rate that exceeds

- the maximum allowed for the current camera settings. If you apply frame start trigger signals to the camera when it isn't ready to receive them, the signals will be ignored.

For more information about determining the maximum allowed frame rate, see Section 7.8 on [page 115](#).

- the host computer's capacity limits for data transfer or storage or both. If you try to acquire more images than the host computer is able to process, frames may be dropped.

For more information about bandwidth optimization, see the *Installation and Setup Guide for Cameras Used With Basler pylon for Windows* (AW000611).

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

- Software: When the source signal is set to software, you apply a frame start trigger signal to the camera by executing a Trigger Software command for the frame start trigger on the host PC.
- Line 1: When the source signal is set to line 1, you apply a frame start trigger signal to the camera by injecting an externally generated electrical signal (commonly referred to as a hardware trigger signal) into the GPIO line of the camera. This assumes that the GPIO line of the camera has been set to operate as an input line.
- CC1: When the source signal is set to CC1, you apply a frame start trigger signal to the camera by injecting an externally generated electrical signal into CC1 in the Camera Link interface.

- CC2: When the source signal is set to CC2, you apply a frame start trigger signal to the camera by injecting an externally generated electrical signal into CC2 in the Camera Link interface.
- CC3: When the source signal is set to CC3, you apply a frame start trigger signal to the camera by injecting an externally generated electrical signal into CC3 in the Camera Link interface.

If the Trigger Source parameter is set to line 1, CC1, CC2, or CC3, you must also set the Trigger Activation parameter. The options available for this parameter are:

- Rising Edge: Specifies that a rising edge of the hardware trigger signal will act as the frame start trigger.
- Falling Edge: Specifies that a falling edge of the hardware trigger signal will act as the frame start trigger.



Typically, a frame grabber is used to supply an electrical frame start signal to CC1, CC2, or CC3.



The CC lines have a negative line logic. For more information, see the note on [page 42](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

For more information about using a software trigger to control frame acquisition start, see Section 7.3.2 on [page 79](#).

For more information about using a hardware trigger to control frame acquisition start, see Section 7.3.3 on [page 82](#).

Exposure Time Control with the Frame Start Trigger On

When the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to software, the exposure time for each frame acquisition is determined by:

- the value of the camera's Exposure Time Abs parameter if you are parameterizing the camera using Basler pylon.
- the value in the camera's Exposure Time Raw register if you are parameterizing the camera using direct register access.

When the Trigger Mode parameter is set to on and the Trigger Source parameter is set to line 1, CC1, CC2, or CC3, the exposure time for each frame acquisition can be controlled using one of the values mentioned in the two bullet points above or it can be controlled by manipulating the hardware trigger signal.

For more information about controlling exposure time when using a software trigger, see Section 7.3.2 on [page 79](#).

For more information about controlling exposure time when using a hardware trigger, see Section 7.3.3 on [page 82](#).

7.3.1.3 Setting the Frame Start Trigger Mode and Related Parameters

Setting the Parameters Using Basler pylon

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

The following code snippet illustrates using the API to set the Trigger Mode parameter for the frame start trigger to on and the Trigger Source parameter to CC1:

```
// Select the frame start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_FrameStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Set the source for the selected trigger
Camera.TriggerSource.SetValue(TriggerSource_CC1);
```

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

```
// Select the frame start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_FrameStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Configure the GPIO line as an input
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Input);
// Set the source for the selected trigger to line 1
Camera.TriggerSource.SetValue(TriggerSource_Line1);
```

The following code snippet illustrates using the API to set the Trigger Mode parameter for the frame start trigger to off and the Acquisition Frame Rate Abs parameter to 60:

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

```
Camera.AcquisitionFrameRateEnable.SetValue(true);  
Camera.AcquisitionFrameRateAbs.SetValue(60.0);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

Setting the Parameters Using Direct Register Access

To set the trigger mode for the frame start trigger to on and to select a trigger source:

1. Set the value of the Trigger Mode Frame Start register to On.
2. Set the value of the Trigger Source Frame Start register to Software, Line 1, CC1, CC2, or CC3.
3. If the trigger source is set to Line 1, set the GPIO line to operate as an input by setting the value of the Line Mode Line 1 register to Input.
4. If the trigger source is set to Line 1, CC1, CC2, or CC3, set the value of the Trigger Activation Frame Start register to Rising Edge or Falling Edge.

To set the trigger mode for the frame start trigger to off, set the exposure time, and set the frame acquisition rate

1. Set the value of the Trigger Mode Frame Start register to Off.
2. Set the value of the Exposure Time Raw register as desired.
A value in a raw register is simply an integer value with no units. To determine what the actual setting will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 μ s.
For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 1000 μ s (1000 x 1 μ s = 1000 μ s).
3. Set the value of the Acquisition Frame Period Enable register to 1 (true).
(This will enable the camera's ability to internally control the frame period.)
4. Set the value of the Acquisition Frame Period Raw register as desired (Frame Rate = 1 /Frame Period).

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.3.2 Using a Software Frame Start Trigger Signal

7.3.2.1 Introduction

If the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to software, you must apply a software frame start trigger signal to the camera to begin each frame acquisition. Assuming that the camera is in a "waiting for frame start trigger" acquisition status, frame exposure will start when the software frame start trigger signal is received by the camera. Figure 21 illustrates frame acquisition with a software frame start trigger signal.

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

When you are using a software trigger signal to start each frame acquisition, the camera's Exposure Mode parameter must be set to timed. The exposure time for each acquired frame will be determined by the value of the camera's Exposure Time Abs parameter if you are parameterizing the camera with Basler pylon or by the Exposure Time Raw parameter if you are parameterizing the camera via direct register access.

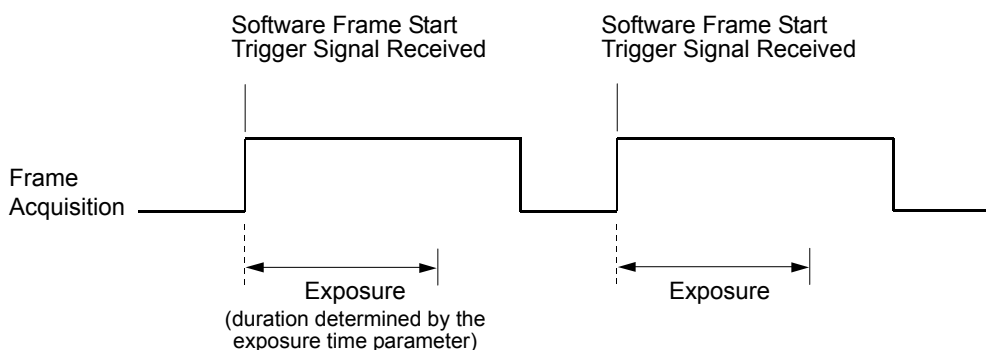


Fig. 21: Frame Acquisition with a Software Frame Start Trigger

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

Section 7.3.2.2 includes more detailed information about applying a software frame start trigger signal to the camera using Basler pylon or via direct register access.

For more information about determining the maximum allowed frame rate with the current camera settings, see Section 7.8 on [page 115](#).

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

Setting the Parameters and Applying the Signal Using Basler pylon

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

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_Off);
// Disable the acquisition frame rate parameter (this will disable the camera's
// internal frame rate control and allow you to control the frame rate with
// software frame start trigger signals)
Camera.AcquisitionFrameRateEnable.SetValue(false);
// Select the frame start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_FrameStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Set the source for the selected trigger
Camera.TriggerSource.SetValue(TriggerSource_Software);
// Set for the timed exposure mode
Camera.ExposureMode.SetValue(ExposureMode_Timed);
// Set the exposure time
Camera.ExposureTimeAbs.SetValue(3000);

while (! finished)
{
    // Execute a trigger software command to apply a frame start
    // trigger signal to the camera
    Camera.TriggerSoftware.Execute();
    // Retrieve acquired frame here
}

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

The following code snippet illustrates using the API to check the acquisition status:

```
// Set the acquisition status selector
Camera.AcquisitionStatusSelector.SetValue(AcquisitionStatusSelector_FrameTrigger
Wait);

// Read the acquisition status
bool IsWaitingForFrameTrigger = Camera.AcquisitionStatus.GetValue();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Parameters and Applying the Signal Using Direct Register Access

To set the parameters for software frame start triggering (acquisition start trigger mode set to off):

1. Set the value of the Trigger Mode Acquisition Start register to Off.
2. Set the value of the Acquisition Frame Period Enable register to 0 (false).
This will disable the camera's ability to internally control the frame period and allow you to control the frame rate with a software trigger signals.
3. Set the value of the Trigger Mode Frame Start register to On.
4. Set the value of the Trigger Source Frame Start register to Software.
5. Set the value of the Exposure Mode register to Timed.
6. Set the value of the Exposure Time Raw parameter as desired.
A value in a raw register is simply an integer value with no units. To determine what the actual setting will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 μ s.
For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 1000 μ s (1000 x 1 μ s = 1000 μ s).
7. Set the value of the Trigger Software Frame Start register to 1.
Setting the value of this register to 1 applies a software frame start trigger to the camera. The register resets to 0 when execution is complete.

To determine the acquisition status of the camera:

1. Read the value of the Status Frame Trigger Wait register.
A value of 0 indicates that the camera is not ready to receive a frame start trigger. A value of 1 indicates that the camera is ready to receive a frame start trigger.

For more information about direct register access, see Section 3.2 on [page 20](#).

7.3.3 Using a Hardware Frame Start Trigger Signal

7.3.3.1 Introduction

If the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to line 1, CC1, CC2, or CC3, an externally generated electrical signal injected into the selected source will act as the frame start trigger signal for the camera. This type of trigger signal is generally referred to as a hardware trigger signal or as an external frame start trigger signal (ExFSTrig signal).

If the Trigger Source is set to line 1, the GPIO line must be properly set to operate as an input in order to accept the trigger signal.

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

Assuming that the camera is in a "waiting for frame start trigger" acquisition status, frame acquisition will start when the hardware frame start trigger signal is received by the camera.

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

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

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

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

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

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

If you have selected CC1, CC2, or CC3 as the trigger source, your frame grabber will typically apply the electrical signal to the selected input via the Camera Link cable. For more information about applying an ExFSTrig signal to CC1, CC2, or CC3, see the documentation for your frame grabber.

If you have selected line 1 as the trigger source, some other kind of external electrical device will be used to apply the electrical signal to the selected input.



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

To avoid overtriggering, you should not attempt to acquire frames at a rate that exceeds the maximum allowed with the current camera settings.

For more information about determining the maximum allowed frame rate with the current camera settings, see Section 7.8 on [page 115](#).

For more information about setting the camera for hardware triggering and selecting the source to receive the ExFSTrig signal, see Section 7.3.3.4 on [page 86](#).

For more information about the electrical requirements for the GPIO line when it is set to operate as an input, see Section 5.6.2 on [page 34](#).

For more information about CC1, CC2, and CC3, see Section 6.1 on [page 42](#).

7.3.3.2 Exposure Modes

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

Timed Exposure Mode

When timed mode is selected, the exposure time for each frame acquisition is determined by:

- the value of the camera's Exposure Time Abs parameter if you are parameterizing the camera with Basler pylon.
- the value of the Exposure Time Raw register if you are parameterizing the camera via direct register access.

If the camera is set for rising edge triggering, the exposure time starts when the ExFSTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExFSTrig signal falls. Figure 22 illustrates timed exposure with the camera set for rising edge triggering.

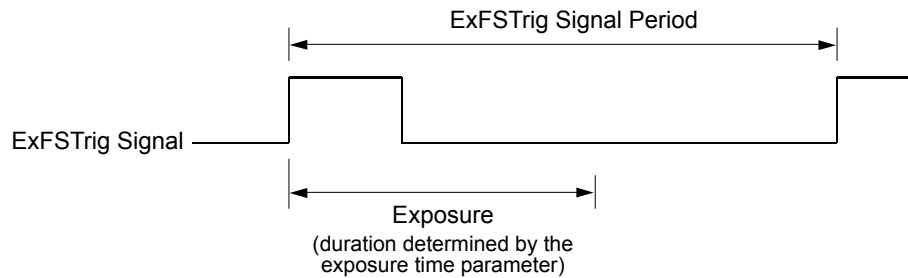


Fig. 22: Timed Exposure with Rising Edge Triggering

Note that if you attempt to trigger a new exposure start while the previous exposure is still in progress, the trigger signal will be ignored, and an overtrigger error will be generated. This situation is illustrated in Figure 23 for rising edge triggering.

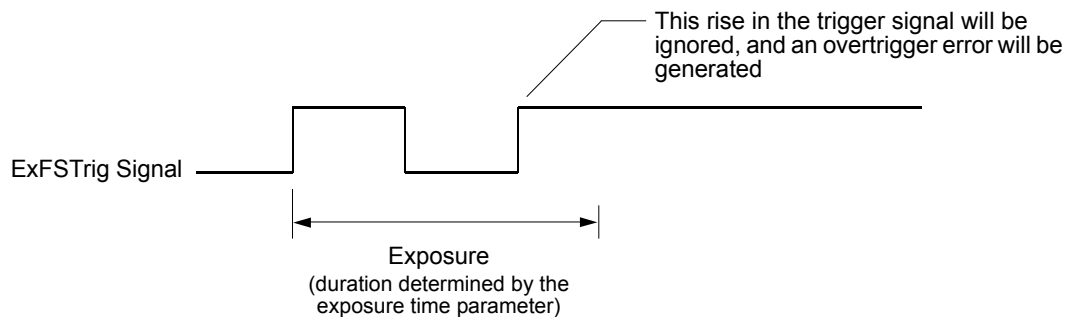


Fig. 23: Overtriggering with Timed Exposure

For more information about setting the exposure time, see Section 7.3.3.4 on [page 86](#).

For more information about the overtrigger error, see Section 10.13 on [page 240](#).

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure for each frame acquisition will be directly controlled by the ExFSTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExFSTrig signal rises and continues until the ExFSTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExFSTrig signal falls and continues until the ExFSTrig signal rises. Figure 24 illustrates trigger width exposure with the camera set for rising edge triggering.

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

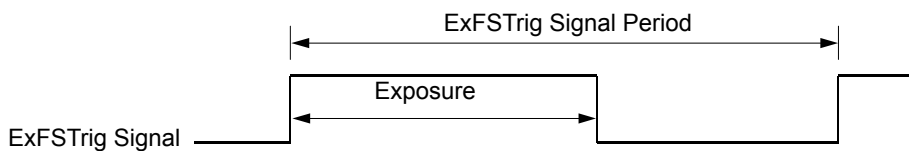


Fig. 24: Trigger Width Exposure with Rising Edge Triggering

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

You should set the Exposure Overlap Time Max Abs parameter value to represent the shortest exposure time you intend to use. For example, assume that you will be using trigger width exposure mode and that you intend to use the ExFSTrig signal to vary the exposure time in a range from 3000 μ s to 5500 μ s. In this case you would set the camera's Exposure Overlap Time Max Abs parameter to 3000 μ s.

For more information about the Frame Trigger Wait signal and the Exposure Overlap Time Max Abs parameter, see Section 7.6.2.2 on [page 103](#).

7.3.3.3 Frame Start Trigger Delay

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

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

If you are parameterizing the camera with Basler pylon, the Trigger Delay Abs parameter will determine the length of the delay.

If you are parameterizing the camera via direct register access, the Trigger Delay Raw Frame Start register will determine the length of the delay.



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

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

Setting the Parameters Using Basler pylon and Applying the Signal

You can set all of the parameters needed to perform hardware frame start triggering from within your application software by using the pylon API.

The following code snippet illustrates using the API to set the parameter values and execute the commands related to hardware frame start triggering with the camera set for the timed exposure mode with rising edge triggering and CC1 as the trigger source. In this example, the trigger mode for the acquisition start trigger will be set to off.



You also have to enable the line inverter as the CC lines have a negative line logic. This means that in order to trigger frame acquisition with a rising edge signal, the line signal needs to be inverted.

For more information about the line logic of the CC lines, see Section 5.7.1 on [page 40](#).

For more information about inverting the line logic, see Section 6.1.5 on [page 47](#).

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_Off);
// Disable the acquisition frame rate parameter (this will disable the camera's
// internal frame rate control and allow you to control the frame rate with
// external frame start trigger signals)
Camera.AcquisitionFrameRateEnable.SetValue(false);
// Select the frame start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_FrameStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Set the source for the selected trigger
Camera.TriggerSource.SetValue(TriggerSource_CC1);
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue(TriggerActivation_RisingEdge);
// Select the CC1 line and enable the line inverter
camera.LineSelector.SetValue(LineSelector_CC1);
camera.LineInverter.SetValue(true);
// Set for the timed exposure mode
Camera.ExposureMode.SetValue(ExposureMode_Timed);
// Set the exposure time
Camera.ExposureTimeAbs.SetValue(3000);

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

The following code snippet illustrates using the API to set the parameter values and execute the commands related to hardware frame start triggering with the camera set for the trigger width exposure mode with rising edge triggering, with line 1 as the trigger source, and with a trigger delay. In this example, the trigger mode for the acquisition start trigger will be set to off:

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_Off);
// Disable the acquisition frame rate parameter (this will disable the camera's
// internal frame rate control and allow you to control the frame rate with
// external frame start signals)
Camera.AcquisitionFrameRateEnable.SetValue(false);
// Select the frame start trigger
Camera.TriggerSelector.SetValue(TriggerSelector_FrameStart);
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue(TriggerMode_On);
// Configure the GPIO line as an input
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Input);
// Set the source for the selected trigger to line 1
Camera.TriggerSource.SetValue(TriggerSource_Line1);
```



```
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue(TriggerActivation_RisingEdge);
// Set for the trigger width exposure mode
Camera.ExposureMode.SetValue(ExposureMode_TriggerWidth);
// Set the trigger delay for one millisecond (1000us == 1ms == 0.001s)
double TriggerDelay_us = 1000.0;
Camera.TriggerDelayAbs.SetValue(TriggerDelay_us);

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

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

Setting the Parameters Using Direct Register Access and Applying the Signal

To set the parameters needed to perform hardware frame start triggering (with the acquisition start trigger mode set to off):

1. Set the value of the Trigger Mode Acquisition Start register to Off.
2. Set the value of the Acquisition Frame Period Enable register to 0 (false).
This will disable the camera's ability to control the frame period internally and allow you to control the frame rate with an external signal.
3. Set the value of the Trigger Mode Frame Start register to On.
4. Set the value of the Trigger Source Frame Start register to receive the external trigger signal on Line 1, CC1, CC2, or CC3.
5. If the trigger source is set to line 1, set the GPIO line to operate as an input by setting the value of the Line Mode Line 1 register to Input.
6. Set the value of the Trigger Activation Frame Start register to Rising Edge or Falling Edge as desired.
7. Set the value of the Exposure Mode register to Timed or to Trigger Width.
If the mode is set to timed, set the value of the Exposure Time Raw register as desired.
A value in a raw register is simply an integer value with no units. To determine what the actual setting will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 μ s.
For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 1000 μ s (1000 x 1 μ s = 1000 μ s).
8. Apply the appropriate externally generated electrical signal (ExFSTrig signal) to the selected trigger source.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.4 Setting the Exposure Time



This section describes how the exposure time can be adjusted "manually", i.e., by setting the value of the Exposure Time parameter.

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

For more information about auto functions in general, see Section 10.11 on [page 219](#).

For more information about the Exposure Auto function in particular, see Section 10.11.5 on [page 232](#).

If you are operating the camera in any one of the following ways, you must use the camera's Exposure Time parameter to set the exposure time:

- the frame start trigger mode is set to off
- the frame start trigger mode is set to on and the trigger source is set to software
- the frame start trigger mode is set to on, the trigger source is set to line 1, CC1, CC2, or CC3, and the exposure mode is set to timed

The minimum allowed exposure time for all camera models is 24 μs .

The maximum allowed exposure time for all camera models is 2500000 μs .

Exposure Time Offset in CMOSIS V2 Sensors

Due to a frequency-related exposure offset in CMOSIS V2 sensors, the exposure always lasts longer than specified in the camera settings. Currently, this offset is not compensated automatically in order to ensure compatibility between older and newer cameras in the field.

The offset is present in all exposure modes (timed and trigger width). The offset values differ depending on the tap geometry, Camera Link clock speed, and bit depth.

If a precise exposure time is critical for you, you have to account for the offset by adjusting the exposure time or the trigger width accordingly.

- When using a timed exposure mode, subtract the offset from the desired exposure time.
- When using the trigger width exposure mode, subtract the offset from the length of the high/low state of the ExFSTrig signal.

For example, if you need an exposure time of 500 μs , and the offset is 18 μs , set the Exposure Time parameter to 482 μs .

Table 7 lists the exposure offsets for different scenarios.

For more information about exposure modes, see Section 7.3.3.2 on [page 83](#).

Camera Model	Bit Depth	Camera Link Clock Speed	Camera Link Taps	Exposure Offset (μs)	
acA2000	12 bit	82 MHz	2	13.9	
			3		
			4		
			6		
			8		
			10		
		65 MHz	2	18.5	
			3	13.9	
			4		
			6		
			8		
			10		
		48 MHz	2		13.9
			3	15.8	
			4	13.9	
			6		
			8		
			10		
		32.5 MHz	2		18.5
			3		13.9
			4	18.5	
			6	13.9	
			8		
			10		

Table 7: Exposure Time Offsets in CMOSIS V2 Sensors

Camera Model	Bit Depth	Camera Link Clock Speed	Camera Link Taps	Exposure Offset (μ s)
acA2000	10 bit	82 MHz	2	13.9
			3	18.5
			4	13.9
			6	18.5
			8	13.9
			10	11.1
		65 MHz	2	18.5
			3	11.6
			4	18.5
			6	11.6
			8	18.5
			10	13.9
		48 MHz	2	11.6
			3	15.8
			4	11.6
			6	15.8
			8	11.6
			10	18.5
		32.5 MHz	2	18.5
			3	11.6
			4	18.5
			6	11.6
			8	18.5
			10	13.9

Table 7: Exposure Time Offsets in CMOSIS V2 Sensors

Camera Model	Bit Depth	Camera Link Clock Speed	Camera Link Taps	Exposure Offset (μ s)	
acA2040	12 bit	82 MHz	2	27.7	
			3		
			4		
			6		
			8		
			10		
		65 MHz	2	37.0	
			3	27.7	
			4		
			6		
			8		
			10		
		48 MHz	2		27.7
			3	31.7	
			4	27.7	
			6		
			8		
			10		
		32.5 MHz	2		37.0
			3		27.7
			4	37.0	
			6	27.7	
			8		
			10		

Table 7: Exposure Time Offsets in CMOSIS V2 Sensors

Camera Model	Bit Depth	Camera Link Clock Speed	Camera Link Taps	Exposure Offset (μ s)
acA2040	10 bit	82 MHz	2	27.7
			3	37.0
			4	27.7
			6	37.0
			8	27.7
			10	22.2
		65 MHz	2	37.0
			3	23.1
			4	37.0
			6	23.1
			8	37.0
			10	27.7
		48 MHz	2	23.1
			3	31.7
			4	23.1
			6	31.7
			8	23.1
			10	37.0
		32.5 MHz	2	37.0
			3	23.1
			4	37.0
			6	23.1
			8	37.0
			10	27.7

Table 7: Exposure Time Offsets in CMOSIS V2 Sensors

Setting the Exposure Time Using Basler pylon

If you are parameterizing the camera with Basler pylon, the exposure time is determined by the setting of the Exposure Time Abs parameter. The Exposure Time Abs parameter sets the exposure time in μs . The exposure time can be set in increments of 1 μs .

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

```
// Set the exposure time to 1000  $\mu\text{s}$ 
Camera.ExposureTimeAbs.SetValue(1000);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Exposure Time Using Direct Register Access

If you are parameterizing the camera via direct register access, the Exposure Time Raw register sets the exposure time.

To set the exposure time:

1. Set the value of the Exposure Time Raw register.

A value in a raw register is simply an integer value with no units. To determine what the actual exposure time will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 μs .

For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 1000 μs (1000 x 1 μs = 1000 μs).

For more information about direct register access, see Section 3.2 on [page 20](#).

7.5 Overlapping Exposure with Sensor Readout

The frame acquisition process on the camera includes two distinct parts. The first part is the exposure of the pixels in the imaging sensor. Once exposure is complete, the second part of the process – readout of the pixel values from the sensor – takes place. In regard to this frame acquisition process, there are two common ways for the camera to operate: with “non-overlapped” exposure and with “overlapped” exposure.

In the non-overlapped mode of operation, each time a frame is acquired the camera completes the entire exposure/readout process before acquisition of the next frame is started. The exposure for a new frame does not overlap the sensor readout for the previous frame. This situation is illustrated in Figure 25 with the camera set for the trigger width exposure mode.

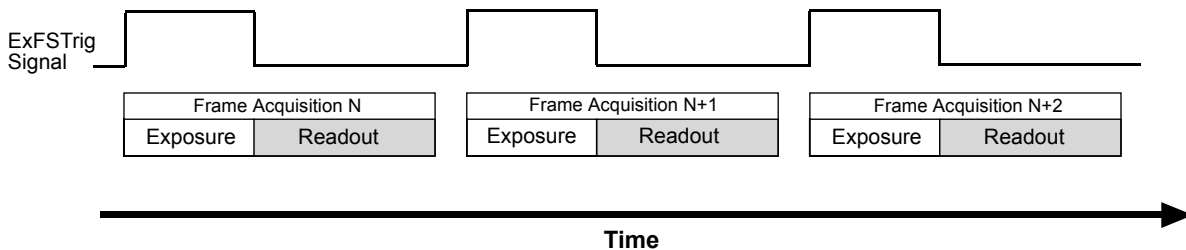


Fig. 25: Non-overlapped Exposure and Readout

In the overlapped mode of operation, the exposure of a new frame begins while the camera is still reading out the sensor data for the previously acquired frame. This situation is illustrated in Figure 26 with the camera set for the trigger width exposure mode.

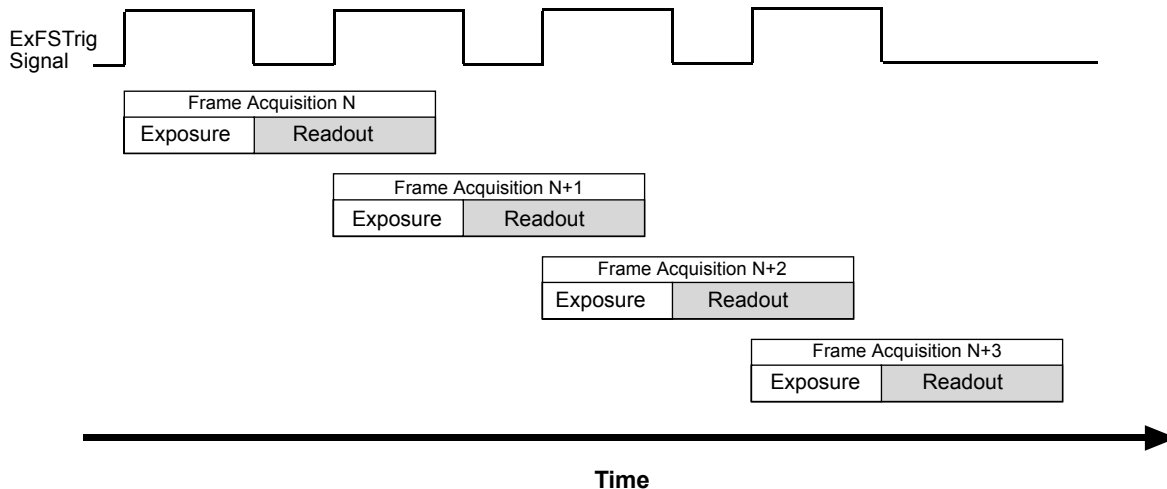


Fig. 26: Overlapped Exposure and Readout

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

- Exposure will not overlap when: $\text{Frame Period} > \text{Exposure Time} + \text{Readout Time}$
- Exposure will overlap when: $\text{Frame Period} \leq \text{Exposure Time} + \text{Readout Time}$

For more information about determining the frame readout time, see Section 7.7 on [page 113](#).

Guideline for Overlapped Operation with Trigger Width Exposure

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

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

If this guideline is violated, the camera will drop the frame for which the exposure was just ended and will declare an overtrigger error. This situation is illustrated in Figure 27 with the camera set for the trigger width exposure mode with rising edge triggering.

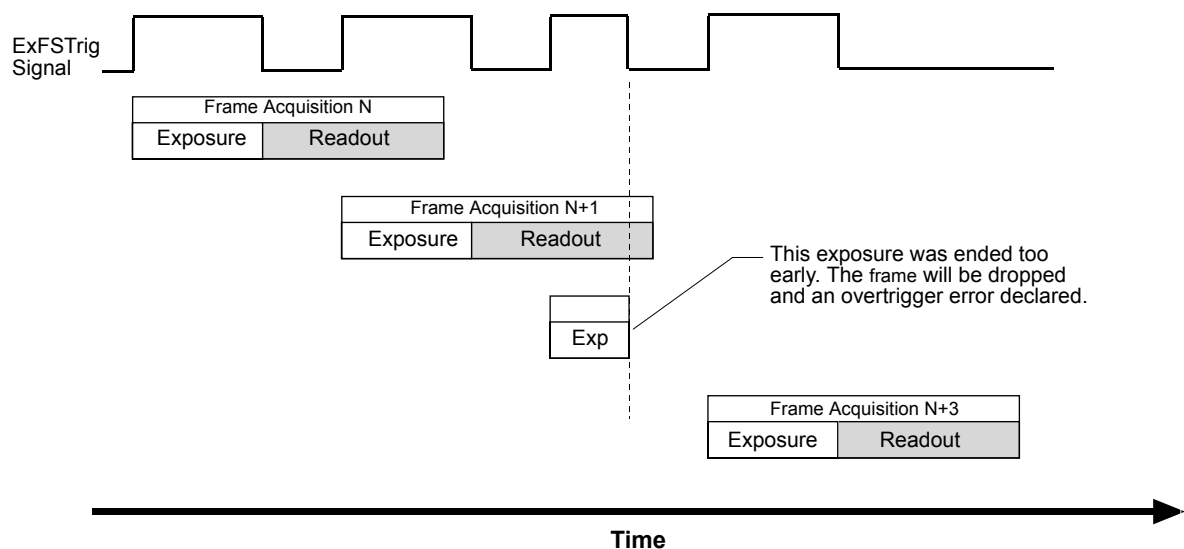


Fig. 27: Overtriggering Caused by an Early End of Exposure

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

For more information about the Frame Trigger Wait signal and the Exposure Overlap Time Max Abs parameter, see Section 7.6.2.2 on [page 103](#).

For more information about trigger width exposure, see Section 7.3.3.2 on [page 83](#).

For more information about the overtrigger error, see Section 10.13 on [page 240](#).

7.6 Acquisition Monitoring Tools

7.6.1 Acquisition Status Indicator

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

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

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

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

For more information about the overtrigger error, see Section 10.13.2 on [page 241](#).

Checking the Acquisition Status Using Basler pylon

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

1. Use the Acquisition Status Selector to select the Acquisition Trigger Wait status or the Frame Trigger Wait status.
2. Read the value of the Acquisition Status parameter.
If the value is set to "false", the camera is not waiting for the trigger signal.
If the value is set to "true", the camera is waiting for the trigger signal.

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

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

// Check the frame start trigger acquisition status
// Set the acquisition status selector
Camera.AcquisitionStatusSelector.SetValue(AcquisitionStatusSelector_FrameTrigger
Wait);
```

```
// Read the acquisition status
bool IsWaitingForFrameTrigger = Camera.AcquisitionStatus.GetValue();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Checking the Acquisition Status Using Direct Register Access

To determine the acquisition start trigger status:

1. Read the value of the Status Acquisition Trigger Wait register.
If the value is set to 0, the camera is not waiting for the trigger signal.
If the value is set to 1, the camera is waiting for the trigger signal.

To determine the frame start trigger status:

1. Read the value of the Status Frame Trigger Wait register.
If the value is set to 0, the camera is not waiting for the trigger signal.
If the value is set to 1, the camera is waiting for the trigger signal.

For more information about direct register access, see Section 3.2 on [page 20](#).

7.6.2 Trigger Wait Signals

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

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

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

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

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

For more information about the overtrigger error, see Section 10.13.2 on [page 241](#).

7.6.2.1 Acquisition Trigger Wait Signal

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

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

Figure 28 illustrates the Acquisition Trigger Wait signal with the Acquisition Frame Count parameter set to 3 and with exposure and readout overlapped. The figure assumes that the trigger mode for the frame start trigger is set to off, so the camera is internally generating frame start trigger signals.

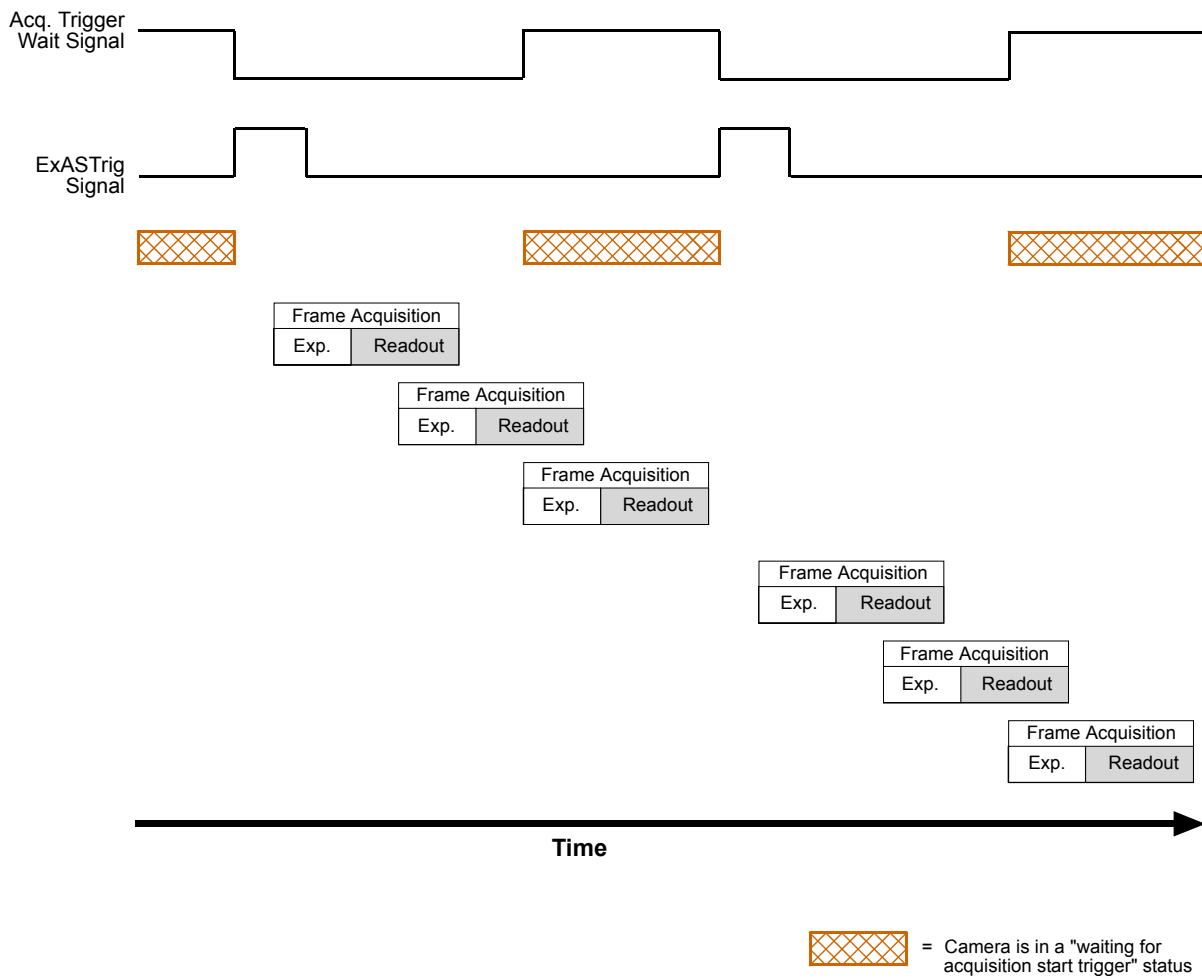


Fig. 28: Acquisition Trigger Wait Signal



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

For more information about the overtrigger error, see Section 10.13.2 on [page 241](#).

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

You can select the acquisition trigger wait signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select a source signal for an output line:

1. Configure the GPIO line as an output (if you want to use the GPIO line).
2. Use the Line Selector to the desired line.
3. Set the value of the Line Source parameter to the acquisition trigger wait signal.

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

```
// Configure the GPIO line as an output
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Output);
//Select the GPIO line
Camera.LineSelector.SetValue(LineSelector_Line1);
//Set the source for the selected line
Camera.LineSource.SetValue(LineSource_AcquisitionTriggerWait);

//Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_ClSpare);
//Set the source for the selected line
Camera.LineSource.SetValue(LineSource_AcquisitionTriggerWait);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Selecting the Acquisition Trigger Wait Signal as the Source Signal for an Output Line Using Direct Register Access

You can select the acquisition trigger wait signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select the acquisition trigger wait signal as the source signal for the GPIO line:

1. Set the GPIO line to operate as an output by setting the value of the Line Mode Line 1 register to Output.
2. Set the value of the Line Source Line 1 register to Acquisition Trigger Wait.

To select the acquisition trigger wait signal as the source signal for the CL Spare line:

1. Set the value of the Line Source CL Spare register to Acquisition Trigger Wait.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.6.2.2 Frame Trigger Wait Signal

Overview

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

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

Figure 29 illustrates the Frame Trigger Wait signal. The camera is set for the trigger width exposure mode with rising edge triggering and with exposure and readout overlapped.

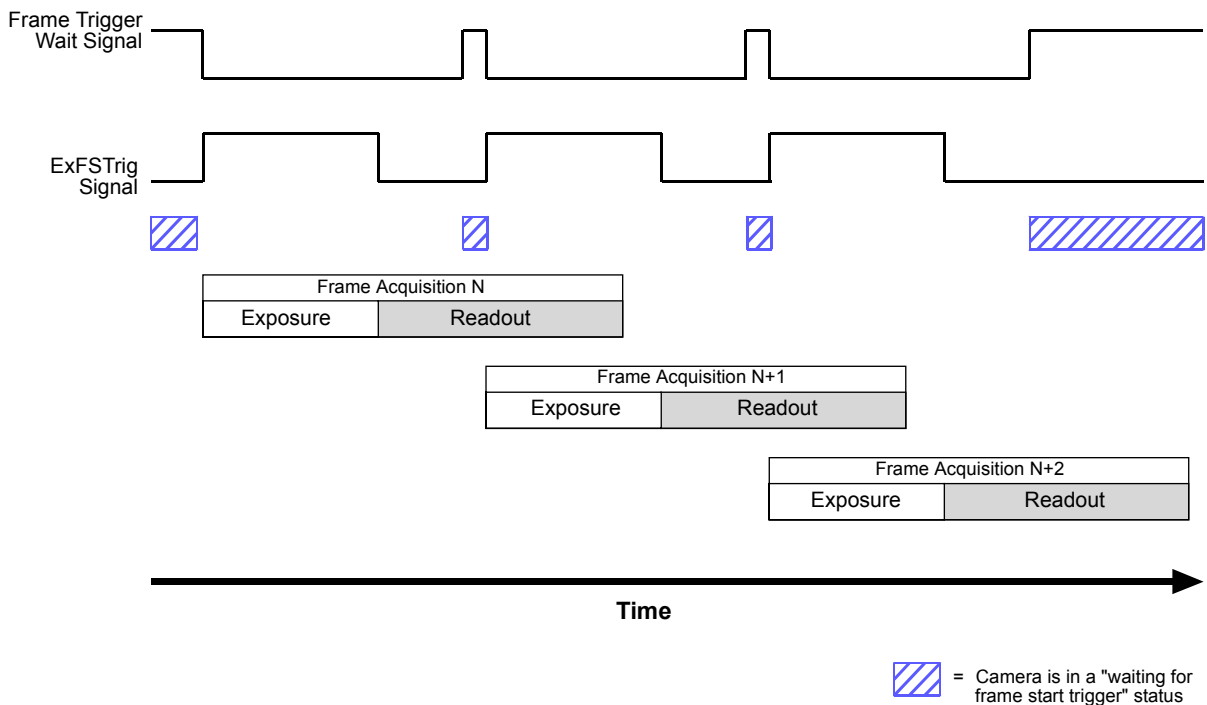


Fig. 29: Frame Trigger Wait Signal



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

For more information about the overtrigger error, see Section 10.13 on [page 240](#).

Frame Trigger Wait Signal Details

When the camera is set for the timed exposure mode, the rise of the Frame Trigger Wait signal is based on the current exposure time parameter setting and on when readout of the current frame will end. This functionality is illustrated in Figure 30.

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

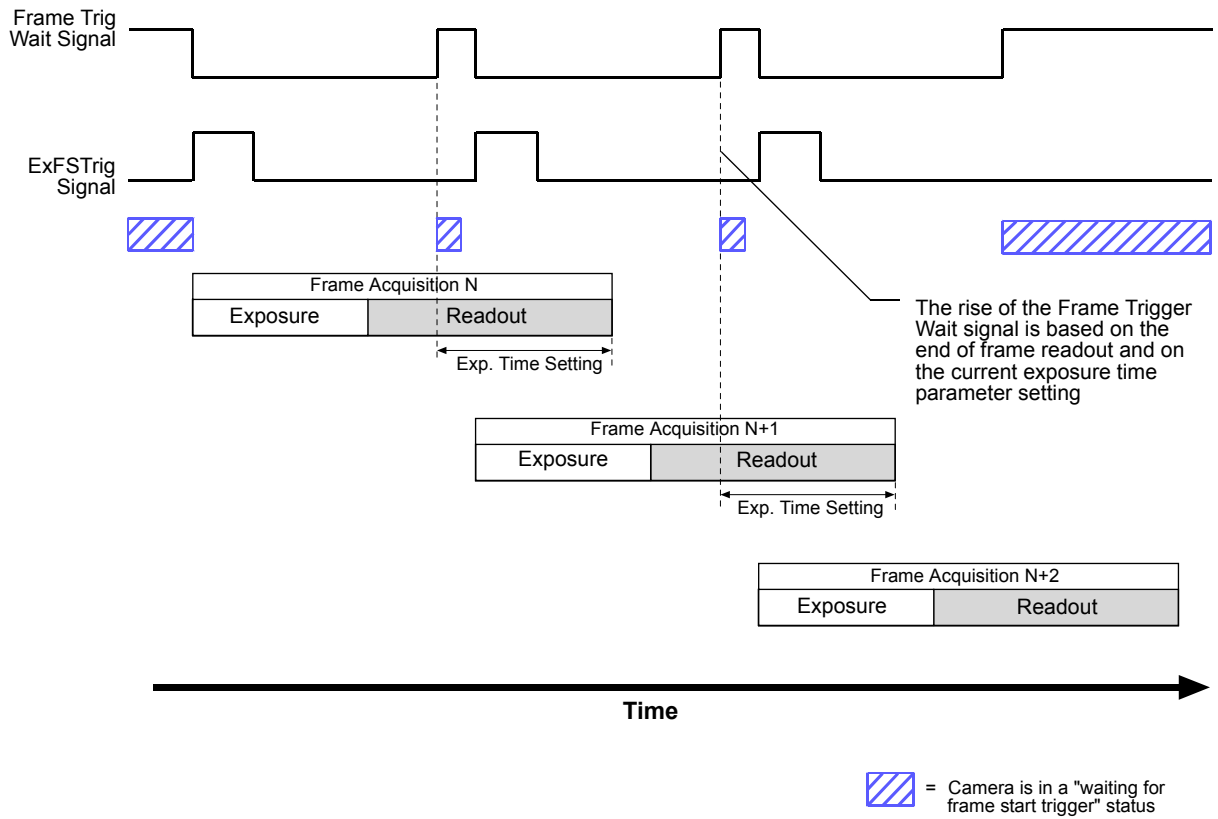


Fig. 30: Frame Trigger Wait Signal with the Timed Exposure Mode

When the camera is set for the trigger width exposure mode, the rise of the Frame Trigger Wait signal is based on the Exposure Overlap Time Max parameter setting and on when readout of the current frame will end. This functionality is illustrated in Figure 31.

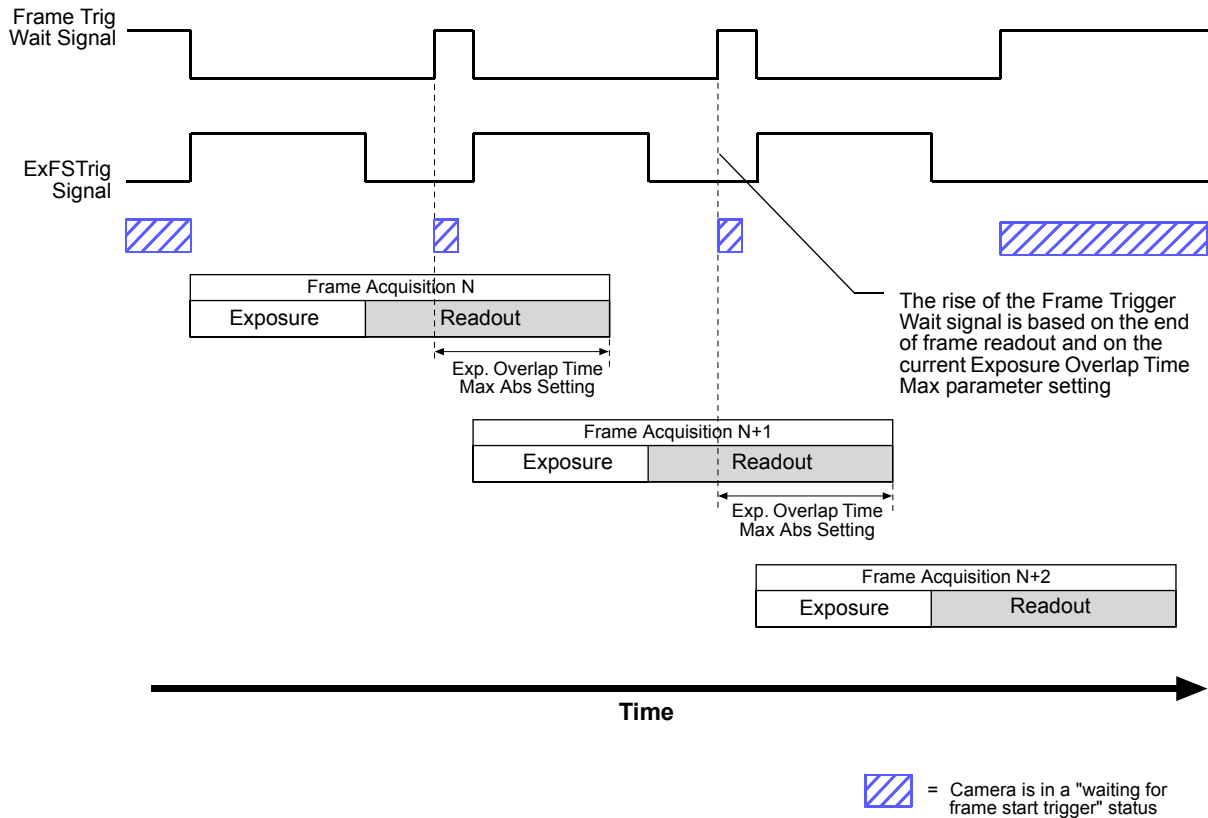


Fig. 31: Frame Trigger Wait Signal with the Trigger Width Exposure Mode

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

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

You should set the Exposure Overlap Time Max parameter value to represent the shortest exposure time you intend to use. For example, assume that you will be using trigger width exposure mode and that you intend to use the ExFSTrig signal to vary the exposure time in a range from 3000 μ s to 5500 μ s. In this case you would set the camera's Exposure Overlap Time Max parameter to 3000 μ s.

Setting the Exposure Overlap Time Max Using Basler pylon

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

```
// Set the Exposure Overlap Time Max to 3000 µs
Camera.ExposureOverlapTimeMaxAbs.SetValue(3000);
```

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

Setting the Exposure Overlap Time Max Using Direct Register Access

If you are parameterizing the camera via direct register access, the Exposure Overlap Time Max Raw register sets the exposure overlap time max.

To set the exposure overlap time max:

1. Set the value of the Exposure Overlap Time Max Raw register.

A value in a raw register is simply an integer value with no units. To determine what the actual exposure time will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 µs.

For example, if you set the Exposure Overlap Time Max Raw register to 3000, the exposure overlap time max would be 3000 µs ($3000 \times 1 \mu\text{s} = 3000 \mu\text{s}$).

Selecting the Frame Trigger Wait Signal as the Source Signal for an Output Line Using Basler pylon

You can select the frame trigger wait signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select a source signal for an output line:

1. Configure the GPIO line as an output (if you want to use the GPIO line).
2. Use the Line Selector to select the desired line.
3. Set the value of the Line Source parameter to the frame trigger wait signal.

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

```
// Configure the GPIO line as an output
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Output);
//Select the GPIO line
Camera.LineSelector.SetValue(LineSelector_Line1);
//Set the source for the selected line
Camera.LineSource.SetValue(LineSource_FrameTriggerWait);
//Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_ClSpare);
```

```
//Set the source for the selected line  
Camera.LineSource.SetValue(LineSource_FrameTriggerWait);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Selecting the Frame Trigger Wait Signal as the Source Signal for an Output Line Using Direct Register Access

You can select the frame trigger wait signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select the frame trigger wait signal as the source signal for the GPIO line:

1. Set the GPIO line to operate as an output by setting the value of the Line Mode Line 1 register to Output.
2. Set the value of the Line Source Line 1 register to Frame Trigger Wait.

To select the frame trigger wait signal as the source signal for the CL Spare line:

1. Set the value of the Line Source CL Spare register to Frame Trigger Wait.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.6.3 Exposure Active Signal

The camera can provide an "exposure active" (ExpAc) output signal. The signal goes high when the exposure time for each frame acquisition begins and goes low when the exposure time ends as shown in Figure 32. (In this example, the camera is operating in the timed exposure mode.)

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.

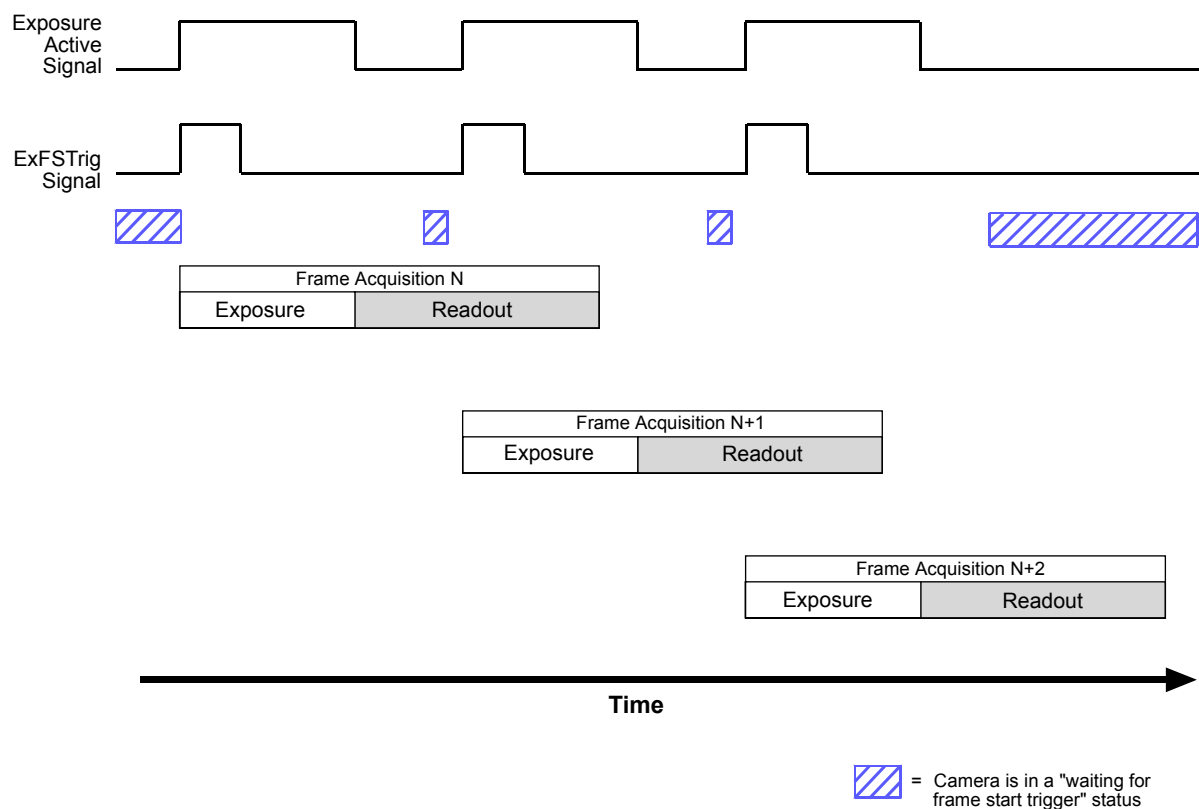


Fig. 32: Exposure Active Signal

Selecting the Exposure Active Signal as the Source Signal for an Output Line Using Basler pylon

You can select the exposure active signal as the source signal for the GPIO line (assuming it is set as an output) or the CL Spare output line.

To select a source signal for an output line:

1. Configure the GPIO line as an output (if you want to use the GPIO line).
2. Use the Line Selector to select the desired line.
3. Set the value of the Line Source parameter to the exposure active output signal.

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

```
// Configure the GPIO line as an output
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Output);
//Select the GPIO line
Camera.LineSelector.SetValue(LineSelector_Line1);
//Set the source signal for the selected line
Camera.LineSource.SetValue(LineSource_ExposureActive);

//Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_ClSpare);
//Set the source signal for the selected line
Camera.LineSource.SetValue(LineSource_ExposureActive);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

Selecting the Exposure Active Signal as the Source Signal for an Output Line Using Direct Register Access

You can select the exposure active signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select the exposure active signal as the source signal for the GPIO line:

1. Set the GPIO line to operate as an output by setting the value of the Line Mode Line 1 register to Output.
2. Set the value of the Line Source Line 1 register to Exposure Active.

To select the exposure active signal as the source signal for the CL Spare line:

1. Set the value of the Line Source CL Spare register to Exposure Active.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.6.4 Frame Cycle Signal

The camera can provide a "Frame Cycle" (FrmCyc) output signal. The signal goes high when the camera enters a waiting for frame trigger condition and goes low when the exposure time for the next triggered image ends as shown in Figure 33. (In this example, the camera is operating in the timed exposure mode.)

The intention of this signal is to let you monitor these two important points in the acquisition process with a single output signal.

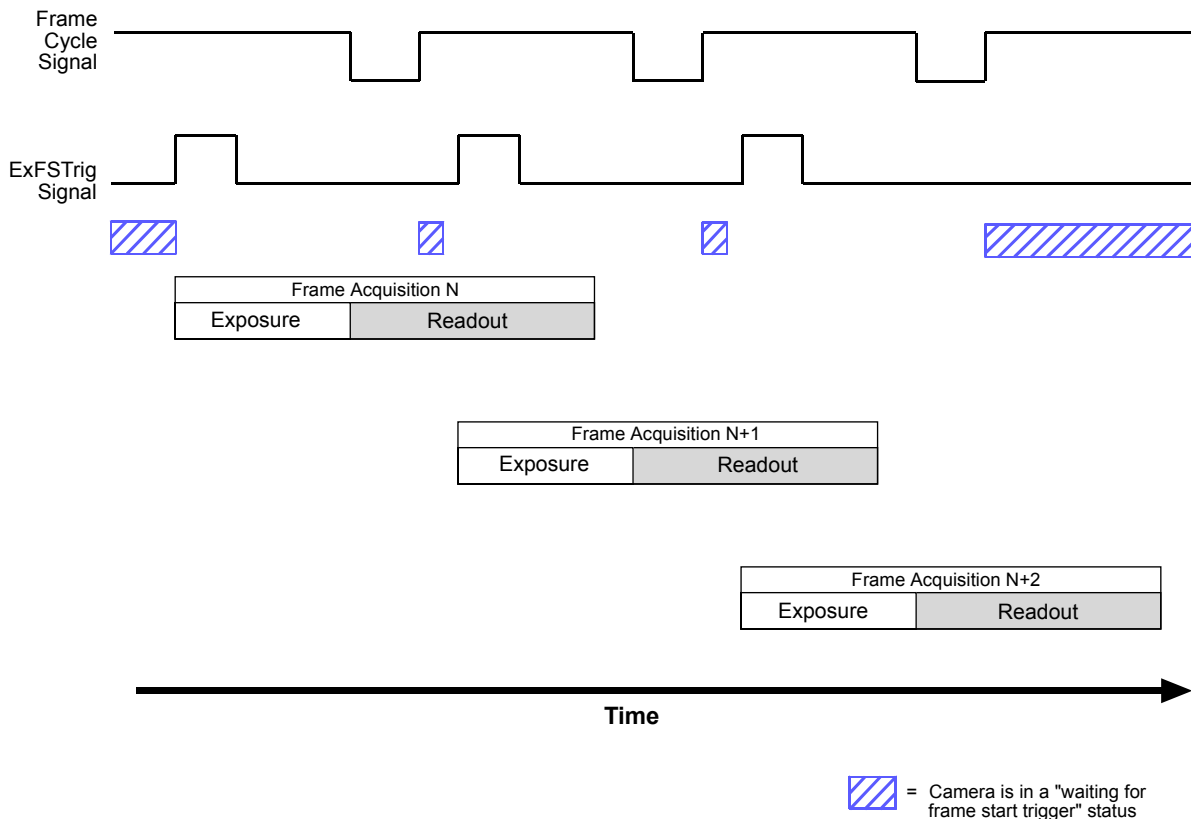


Fig. 33: Frame Cycle Signal

Selecting the Frame Cycle Signal as the Source Signal for an Output Line Using Basler pylon

You can select the frame cycle signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select a source signal for an output line:

1. Configure the GPIO line as an output (if you want to use the GPIO line).
2. Use the Line Selector to select the desired line.

3. Set the value of the Line Source parameter to the frame cycle output signal.

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

```
// Configure the GPIO line as an output
Camera.LineSelector.SetValue(LineSelector_Line1);
Camera.LineMode.SetValue(LineMode_Output);

//Select the GPIO line
Camera.LineSelector.SetValue(LineSelector_Line1);
//Set the source signal for the selected line
Camera.LineSource.SetValue(LineSource_FrameCycle);

//Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_ClSpare);
//Set the source signal for the selected line
Camera.LineSource.SetValue(LineSource_FrameCycle);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

Selecting the Frame Cycle Signal as the Source Signal for an Output Line Using Direct Register Access

You can select the frame cycle signal as the source signal for the camera's GPIO line (assuming it is set as an output) or the CL Spare output line.

To select the frame cycle signal as the source signal for the GPIO line:

1. Set the GPIO line to operate as an output by setting the value of the Line Mode Line 1 register to Output.
2. Set the value of the Line Source Line 1 register to Frame Cycle.

To select the frame cycle signal as the source signal for the CL Spare line:

1. Set the value of the Line Source CL Spare register to Frame Cycle.

For more information about direct register access, see Section 3.2 on [page 20](#).

For more detailed information about the camera's GPIO line, see Section 5.6 on [page 33](#).

7.7 Acquisition Timing Chart

Figure 34 shows a timing chart for frame acquisition and transmission. The chart assumes that exposure is triggered by an externally generated frame start trigger (ExFSTrig) signal with rising edge activation and that the camera is set for the timed exposure mode.

As Figure 34 shows, there is a slight delay between the rise of the ExFSTrig signal and the start of exposure. After the exposure time for a frame acquisition is complete, the camera begins reading out the acquired image data from the imaging sensor. After a short delay, the camera will begin transmitting image data from the camera to the host PC.

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

Note that, if the debouncer feature is used, the debouncer setting for the input line must be added to the exposure start delay shown in Figure 34 to determine the total start delay.

The **frame readout time** is the amount of time it takes to read out the data for an acquired frame from the imaging sensor. This time period will vary depending on several camera settings such as the image AOI, the Camera Link tap geometry, the Camera Link pixel clock speed, and others. As explained below, you can read the value of the Readout Time parameter to determine the sensor readout time given the current camera settings.

The **frame transmission time** is the amount of time it takes to transmit an acquired frame from the buffer in the camera to the host PC via the network. The frame transmission time will always be equal to the frame readout time.

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

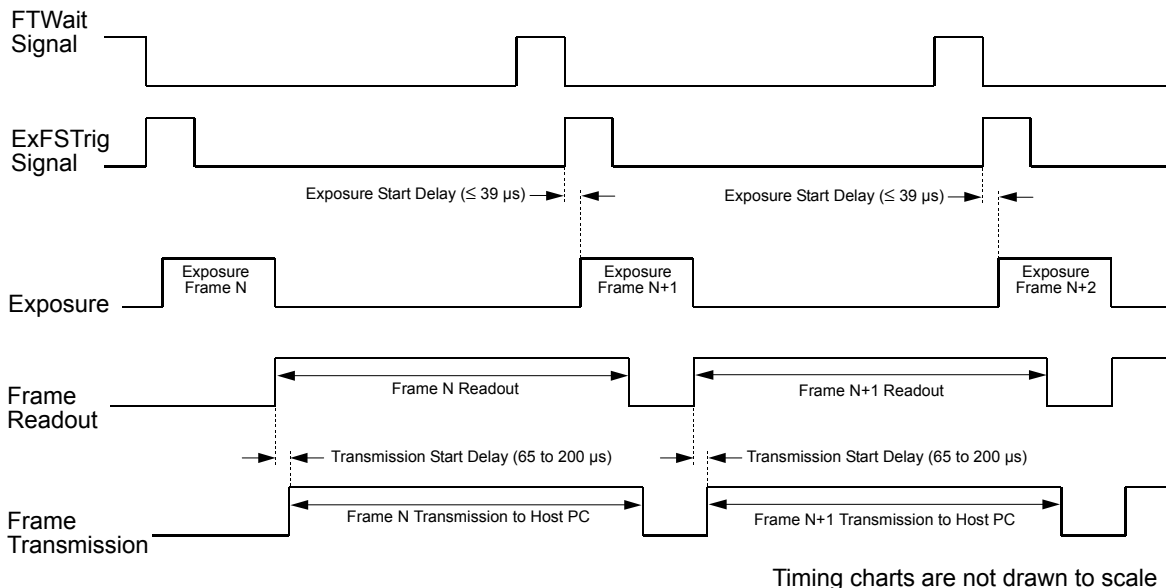


Fig. 34: Exposure Start Controlled with an ExFSTrig Signal

Determining the Frame Readout Time Using Basler pylon

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

The following code snippet illustrates using the API to get the parameter value:

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

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Determining the Frame Readout Time Using Direct Register Access

To determine the frame readout time:

1. Read the value of the Readout Time Raw register.

A value in a raw register is simply an integer value with no units. To determine what the actual exposure time will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 μ s.

For example, if you read the Readout Time Raw register value as 18000, the frame readout time would be 18000 μ s (18000 \times 1 μ s = 18000 μ s).

Note that the frame transmission time is always equal to the frame readout time.

7.8 Maximum Allowed Frame Acquisition Rate

The maximum allowed frame acquisition rate for your camera is not static. It can vary depending on how certain camera features are set. In general, the following factors can affect the maximum allowed frame rate:

- The Camera Link pixel clock speed and the Camera Link tap geometry settings.
When the pixel clock speed is set to a high value, it will take less time to transfer captured images from the camera to the frame grabber in your host PC.
When the camera is set for a tap geometry that uses more taps (e.g., the 1X2-1Y geometry uses two taps and the 1X3-1Y geometry uses three taps), it can typically transfer data out of the camera faster.
So if the camera is set for a high pixel clock speed and a high number of taps, it typically will have a much higher maximum allowed frame rate than when it is set for a low pixel clock speed and a low number of taps.
- The sensor bit depth setting. The sensor can capture pixel data at 12-bit or at 10-bit depth. When the sensor is set for 12-bit depth, it operates slower than at 10-bit depth. If you have the sensor set for 10-bit depth, you can acquire images at a significantly higher maximum allowed rate.
- The amount of time it takes to read the data for an acquired image (known as a frame) out of the imaging sensor. This time varies depending on the height of the frame. The frame height is determined by the camera's AOI settings. If you use shorter AOIs (i.e., AOIs that include fewer lines), it takes less time to read an acquired frame out of the sensor and you can acquire more frames per second.
Decreasing the width of the AOI can also yield a significant increase in the maximum allowed frame rate, but this is only true if the width is reduced by 50 % of the sensor width (e.g., from 2048 to 1024 or from 1024 to 512).
- Whether binning is enabled. Enabling either horizontal or vertical binning may yield a significant increase in the maximum allowed frame rate, but this will only be true if the camera is not already reading out image data at a rate that is near to the current data carrying capacity of the Camera Link interface.
- The exposure time for acquired frames. If you use very long exposure times, you can acquire fewer frames per second.

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

- You can go to the Support section of the Basler website and use the online frame rate calculator: www.baslerweb.com
- You can use Basler pylon to read the value of the camera's Resulting Frame Rate Abs parameter (see below).
- You can use direct register access to read the value of the Resulting Frame Period register (see below).

For more information about the image AOI settings, see Section 10.4 on [page 156](#).

For more information about selectable pixel clock speeds, see Section 10.1 on [page 152](#).

For more information about the sensor bit depth, see Section 10.1 on [page 152](#).

For more information about Camera Link tap geometries, see Section 9.3 on [page 141](#).

Using Basler pylon to Check the Maximum Allowed Frame Rate

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

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

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

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Using Direct Register Access to Check the Maximum Allowed Rate

When using direct register access, you work with the "resulting frame period" rather than the frame acquisition rate. Once the resulting frame period is known, the maximum allowed frame acquisition rate can be determined by:

$$\text{Max Frame Acquisition Rate} = \frac{1}{\text{Resulting Frame Period in Seconds}}$$

To determine the resulting frame period:

1. Check the value of the Resulting Frame Period Raw register.

The Resulting Frame Period Raw register takes all camera settings that can influence the frame period into account and indicates the minimum allowed frame period given the current settings.

A value in a raw register is simply an integer value with no units. To determine what the actual frame period will be, you must multiply the value in the raw register by the camera's time base. The time base on ace cameras is 1 μs .

For example, if you read the Resulting Frame Period Raw register and find that its value is 10000, the resulting frame period would be 10000 μs (10000 x 1 μs = 10000 μs).

For more information about direct register access, see Section 3.2 on [page 20](#).

7.8.1 Increasing the Maximum Allowed Frame Rate

You may find that you would like to acquire frames at a rate higher than the maximum allowed with the camera's current settings. In this case, you must adjust one or more of the factors that can influence the maximum allowed rate. Afterwards, check whether the maximum allowed rate has increased:

- If you have the Camera Link pixel clock speed on your camera set to a low value, consider setting it to a higher value. Be aware, however, that some frame grabbers can't handle higher pixel clock speeds. Before you increase the setting for the camera's pixel clock, make sure that your frame grabber is compatible with the higher setting.
- If you are using a Camera Link configuration with a low number of taps, consider using a configuration with a higher number.
- If you have the sensor bit depth set to 12 bits, consider changing the value to 10 bits. This will typically yield a significant increase in the maximum allowed frame rate.
- Decreasing the height of the AOI can have a significant impact on the maximum allowed frame rate. If possible in your application, decrease the height of the AOI.

Decreasing the width of the AOI can also yield a significant increase in the maximum allowed frame rate, but this is only true if the width is reduced by 50 % of the sensor width (e.g., from 2048 to 1024 or from 1024 to 512).

- If possible in your application, enable either horizontal or vertical binning (or both). This may yield a significant increase in the maximum allowed frame rate, but this will only be true if the camera is not already reading out image data at a rate that is near to the current data carrying capacity of the Camera Link interface.



Decreasing the sensor bit depth, decreasing the AOI height, decreasing the AOI width, and enabling binning can each result in an increase in the maximum allowed frame rate because they decrease the time that it takes to read out pixel data from the imaging sensor. But in order to actually see an increase in the maximum allowed frame rate, you must have the pixel clock speed and the number of taps set high enough so that the Camera Link interface can transfer the larger amount of image data that will be generated when the camera is operated at a higher frame rate.

If you decrease the bit depth, decrease the AOI Height, decrease the AOI Width, or enable binning and you have the pixel clock speed and the number of taps set to low values, you will see no increase in the maximum allowed frame rate.

- If you are using normal exposure times and you are using the camera at its maximum resolution, your exposure time will not normally restrict the frame rate. However, if you are using long exposure times or a small area of interest, it is possible that your exposure time is limiting the maximum allowed frame rate. If you are using a long exposure time or a small AOI, try using a shorter exposure time and see if the maximum allowed frame rate increases. (You may need to compensate for a shorter exposure time by using a brighter light source or increasing the opening of your lens aperture.)



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

For more information about the sensor bit depth feature, see Section 10.1 on [page 152](#).

For more information about the image AOI settings, see Section 10.4 on [page 156](#).

For more information about selectable pixel clock speeds, see Section 10.1 on [page 152](#).

For information about Camera Link tap geometries, see Section 9.3 on [page 141](#).

For more information about the exposure time, see Section 7.4 on [page 89](#).

7.9 Use Case Descriptions and Diagrams

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

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

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



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

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

Use case 1 is illustrated on [page 120](#).

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

The rate at which the camera will acquire images will normally be determined by the camera's Acquisition Frame Rate Abs parameter when the camera is parameterized via Basler pylon or by the Acquisition Frame Period Raw register when the camera is parameterized via direct register access. If the Acquisition Frame Rate Abs parameter or the Acquisition Frame Period Raw register (respectively) is disabled, the camera will acquire frames at the maximum allowed frame rate.

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

For more information about the Acquisition Frame Rate Abs parameter and the Acquisition Frame Period Raw register, see Section 7.3.1.1 on [page 74](#) and Section 7.3.1.3 on [page 77](#).

Use Case: "Free Run" (Acquisition Start Trigger Off and Frame Start Trigger Off)
 The acquisition start trigger is off. The camera will generate acquisition start trigger signals internally with no action by the user.
 The frame start trigger is off. The camera will generate frame start trigger signals internally with no action by the user.

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

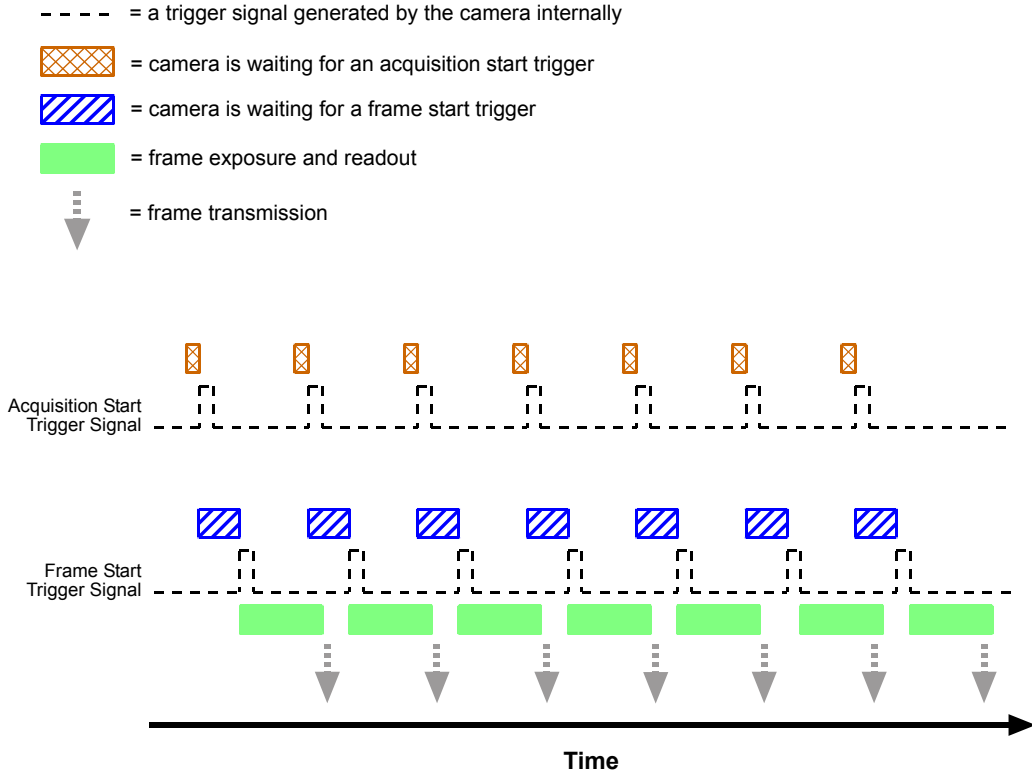


Fig. 35: Use Case 1: Acquisition Start Trigger Off and Frame Start Trigger Off

Use Case 2: Acquisition Start Trigger Off, Frame Start Trigger On

Use case 2 is illustrated on [page 122](#).

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

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

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

(Note that the GPIO line is designated as line 1 and in this case it has been set to act as an input line.)

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

Use Case: Acquisition Start Trigger Off and Frame Start Trigger On
 The acquisition start trigger is off. The camera will generate acquisition start trigger signals internally with no action by the user.
 The frame start trigger is on, and the frame start trigger source is set to line 1. The user must apply a frame start trigger signal to line 1 to start each frame exposure.

Settings: Trigger Mode for the acquisition start trigger = Off
 Trigger Mode for the frame start trigger = On
 Trigger Source for the frame start trigger = Line 1
 Trigger Activation for the frame start trigger = Rising Edge

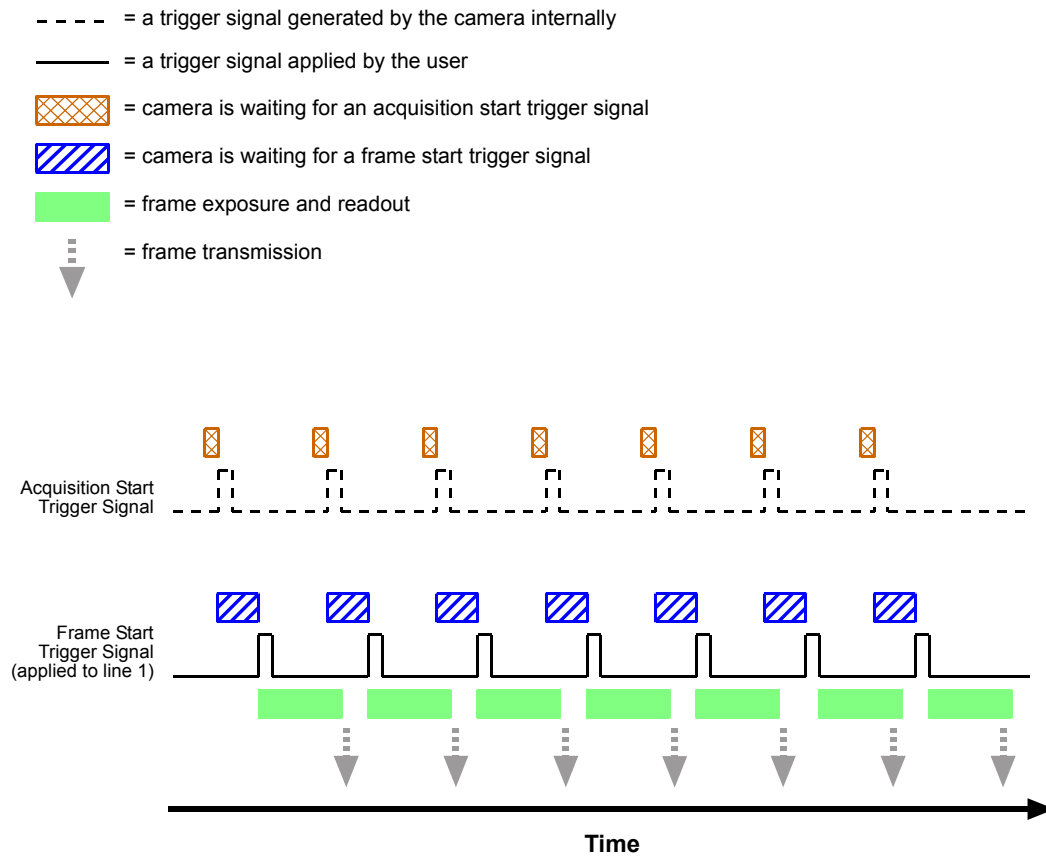


Fig. 36: Use Case 2: Acquisition Start Trigger Off and Frame Start Trigger On

Use Case 3: Acquisition Start Trigger On, Frame Start Trigger Off

Use case 3 is illustrated on [page 124](#).

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

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

(Note that the GPIO line is designated as line 1 and in this case it has been set to act as an input line.)

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

Because the frame start trigger is set to off, the user does not need to apply frame start trigger signals to the camera. The camera will generate all required frame start trigger signals internally. The rate at which the frame start trigger signals will be generated is normally determined by the camera's Acquisition Frame Rate Abs parameter when the camera is parameterized via Basler pylon or by the Acquisition Frame Period Raw register the camera is parameterized via direct register access. If the Acquisition Frame Rate Abs parameter or the Acquisition Frame Period Raw register (respectively) is disabled, the camera will acquire frames at the maximum allowed frame rate.

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

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





For more information about the Acquisition Frame Rate Abs parameter and the Acquisition Frame Period Raw register, see Section 7.3.1.1 on [page 74](#) and Section 7.3.1.3 on [page 77](#).

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

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

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

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

- - - - = a trigger signal generated by the camera internally
- = a trigger signal applied by the user
-  = camera is waiting for an acquisition start trigger signal
-  = camera is waiting for a frame start trigger signal
-  = frame exposure and readout
-  = frame transmission

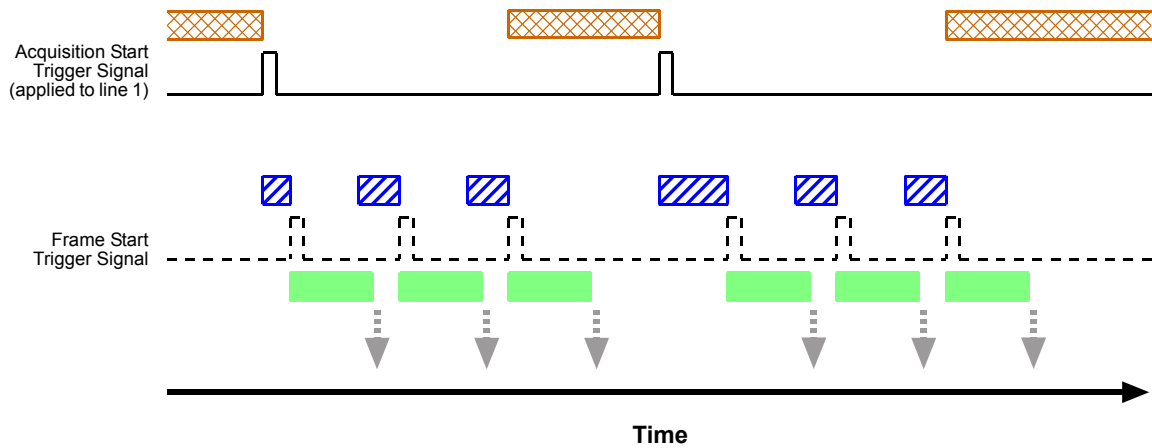


Fig. 37: Use Case 3: Acquisition Start Trigger On and Frame Start Trigger Off

Use Case 4: Acquisition Start and Frame Start Triggers On

Use case 4 is illustrated on [page 126](#).

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

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

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

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

(Note that the GPIO line is designated as line 1 and in this case it has been set to act as an input line.)

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

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

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

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

The acquisition start trigger is on, and the acquisition start trigger source is set to software. The user must execute an acquisition start trigger software command to make the camera exit the "waiting for acquisition start trigger" acquisition status. Because the acquisition frame count is set to 3, the camera will re-enter the "waiting for acquisition start trigger" acquisition status after 3 frame trigger signals have been applied.

The frame start trigger is on, and the frame start trigger source is set to line 1. The user must apply a frame start trigger signal to line 1 to start each frame exposure.

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

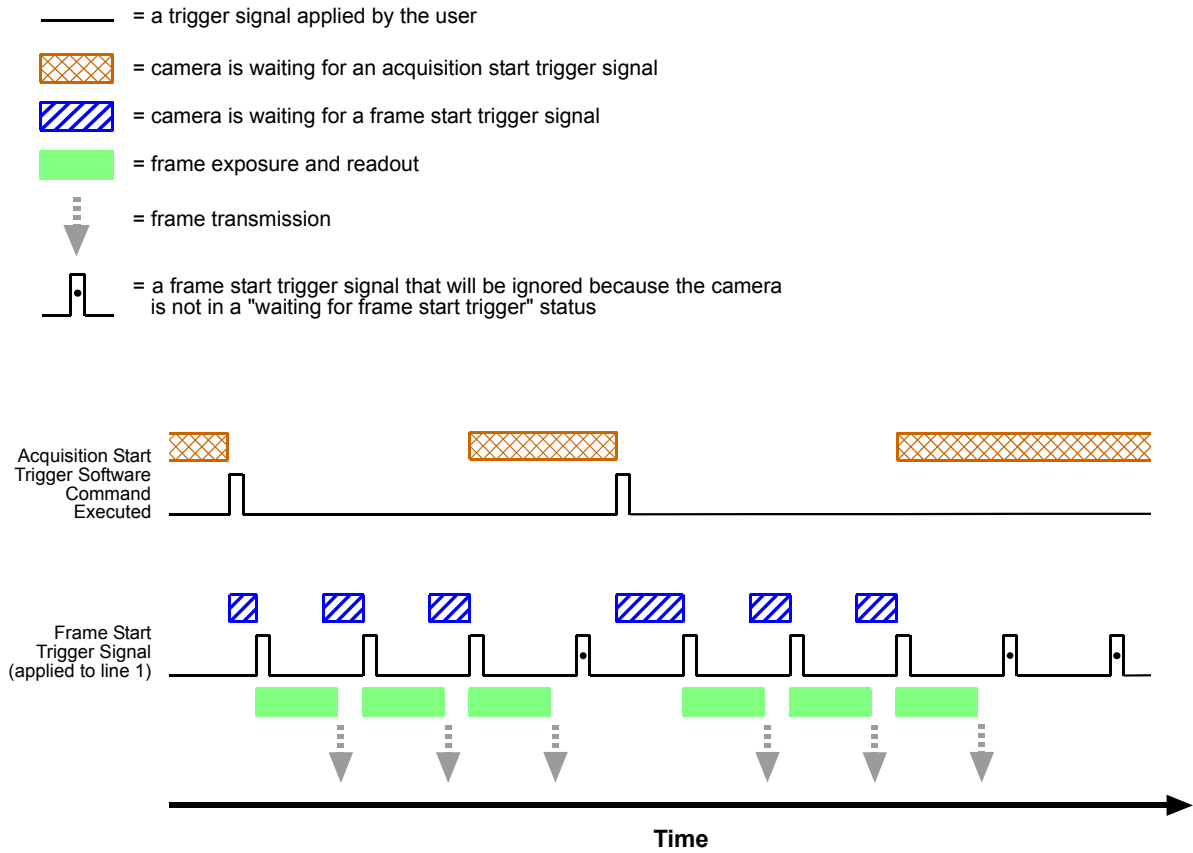


Fig. 38: Use Case 4: Acquisition Start Trigger On and Frame Start Trigger On

8 Color Creation and Enhancement

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

8.1 Color Creation

The sensor used in the cameras 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 part of the filter that allows light of only one color to strike the pixel. The pattern of the Bayer filter used on the camera is as shown in Figure 39 (the alignment of the Bayer filter with respect to the sensor is shown as an example only; the figure shows the "GB" filter alignment). As the figure illustrates, within each square of four pixels, one pixel sees only red light, one sees only blue light, and two pixels see only green light. (This combination mimics the human eye's sensitivity to color.)

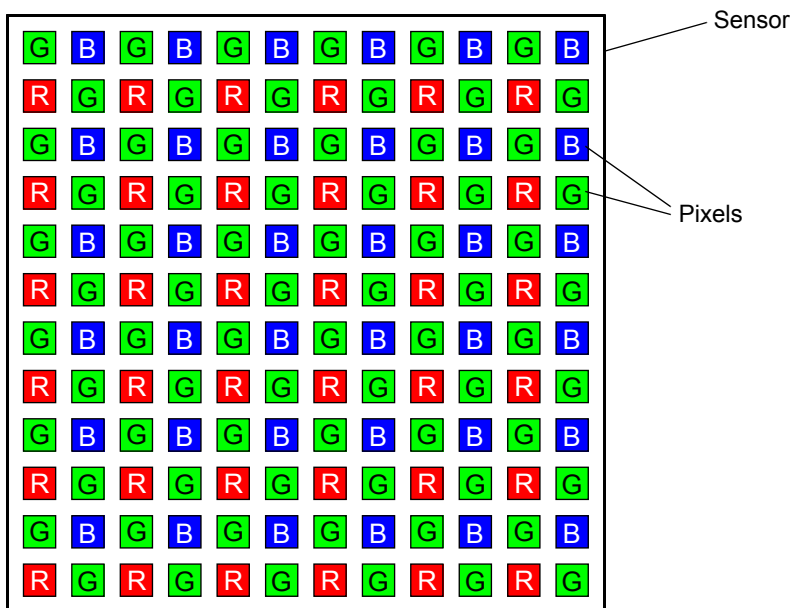


Fig. 39: Bayer Filter Pattern

8.1.1 Bayer Color Filter Alignment

On all color camera models, the alignment of the filter to the pixels in the acquired images is Bayer GB. Bayer GB alignment means that pixel one and pixel two of the first line in each image transmitted will be green and blue respectively. And for the second line transmitted, pixel one and pixel two 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.

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.

The image area of interest (AOI) parameters can only be changed in certain defined increments. The increments are designed so that the alignment of the Bayer filter to the pixels in the transmitted images will stay the same regardless of the size of the image AOI.

When either the reverse X feature or the reverse Y feature or both are used, the alignment of the color filter to the image remains Bayer GB. The camera includes a mechanism that keeps the filter alignment constant when these features are used.

For more information about the camera's AOI feature, see Section 10.4 on [page 156](#).

For more information about the reverse X and reverse Y features, see Section 10.9 on [page 209](#).

8.1.2 Pixel Data Formats Available on Color Cameras

Bayer Formats

Cameras equipped with a Bayer pattern color filter can output pixel data in the Bayer GB 8, the Bayer GB 10, or the Bayer GB 12 pixel data format.

When a color camera is set for one of these three pixel data output formats, the pixel data is not processed or interpolated in any way. For each pixel covered with a red portion of the filter, you get 8, 10, or 12 bits of red data, for each pixel covered with a green portion of the filter, you get 8, 10, or 12 bits of green data, and for each pixel covered with a blue portion of the filter, you get 8, 10, or 12 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

8.2 Integrated IR Cut Filter (All Color Models)

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

Monochrome cameras include a filter holder in the lens mount, but the holder does not contain an IR-cut filter.

NOTICE

On all cameras, the lens thread length is limited.

All cameras (mono and color) are equipped with a plastic filter holder located in the lens mount. The location of the filter holder limits the length of the threads on any lens you use with the camera. If a lens with a very long thread length is used, the filter holder or the lens mount will be damaged or destroyed and the camera will no longer operate.

For more information about the location of the IR cut filter, see Section 1.4.2 on [page 9](#).

8.3 Color Enhancement Features

8.3.1 White Balance



This section describes how a color camera's white balance can be adjusted "manually", i.e., by setting the value of the Balance Ratio Abs parameters for red, green, and blue.

The camera also has a White Balance Auto function that can automatically adjust the white balance. **Manual adjustment of the Balance Ratio Abs parameters for red, green, and blue will only work correctly if the Balance White Auto function is disabled.**

For more information about auto functions in general, see Section 10.11 on [page 219](#).

For more information about the Balance White Auto function, see Section 10.11.7 on [page 236](#).

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.

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 15.984. 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.

Setting the White Balance Using Basler pylon

To set the Balance Ratio parameter for a color:

1. Set the Balance Ratio Selector to red, green, or blue.
2. Set the Balance Ratio Abs parameter to the desired value for the selected color.

You can use the pylon API to set the Balance Ratio Selector and the Balance Ratio Abs parameter value from within your application software. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set the red balance ratio
Camera.BalanceRatioSelector.SetValue(BalanceRatioSelector_Red);
Camera.BalanceRatioAbs.SetValue(1.20);

// Set the green balance ratio
Camera.BalanceRatioSelector.SetValue(BalanceRatioSelector_Green);
Camera.BalanceRatioAbs.SetValue(1.20);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the White Balance Using Direct Register Access

To set the balance ratio parameters:

1. Set the value of White Balance Red register to adjust the red ratio.
2. Set the value of the White Balance Green register to adjust the green ratio.
3. Set the value of the White Balance Blue register to adjust the blue ratio.

For more information about direct register access, see Section 3.2 on [page 20](#).

8.3.2 Gamma Correction

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

sRGB Gamma

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

User Gamma

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

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

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

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, 1023 for 10-bit output, and 4095 for 12-bit output.

The gamma correction value can be set in a range from 0 to 3.99998. For information on setting the gamma correction using hexadecimal instead of decimal figures, see "Setting Gamma Correction Using Hexadecimal Figures" on [page 133](#).

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

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

In all cases, black (output pixel brightness equals 0) and white (output pixel brightness equals 255 at 8-bit output, 1023 at 10-bit output, and 4095 at 12-bit output) will not be corrected.

Enabling and Setting Gamma Correction Using Basler pylon

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

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

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

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

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

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

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

```
// Set the Gamma value to 1.2  
Camera.Gamma.SetValue(1.2);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Enabling and Setting Gamma Correction Using Direct Register Access

To enable gamma correction and to set the gamma value:

1. Set the value of the Gamma Enable register to Enabled.
2. Set the value of the Gamma Selector register to SRGB or User.
3. Set the value in the Gamma register to the desired gamma value.

For more information about direct register access, see Section 3.2 on [page 20](#).

Setting Gamma Correction Using Hexadecimal Figures

If you want to change the gamma correction value using hexadecimal figures, a conversion factor is required. A gamma correction value of 1 (neutral gamma) corresponds to 65536 in the decimal system. If you want to set a gamma correction value other than 1, multiply the desired value by 65536 and convert the result to a hexadecimal figure. The sample calculation below shows how to set a gamma correction value of 0.42.

To set a gamma correction value of 0.42:

1. Perform the following calculation:
 $0.42 \times 65536 = 27525$
2. Convert 27525 to a hexadecimal figure.

8.3.3 Matrix Color Transformation

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

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

The first camera parameter associated with matrix color transformation is the Color Transformation Selector parameter. This parameter is used to select the type of transformation that will be performed. For ace Camera Link cameras, RGB to RGB is the only setting available. This setting simply means that the camera will not transform the red, green, and blue pixel values from the sensor into a different color space (such as YUV).

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

- Off: No alterations will be made to the pixel values.
- Daylight: This setting will automatically populate the matrix with a pre-selected set of values that will make appropriate corrections for images captured with daylight lighting that has a color temperature of about 5000K.
- Tungsten: This setting will automatically populate the matrix with a pre-selected set of values that will make appropriate corrections for images captured with tungsten lighting that has a color temperature of about 2500K to 3000K.
- Daylight 6500K: This setting will automatically populate the matrix with a pre-selected set of values that will make appropriate corrections for images captured with daylight lighting that has a color temperature of about 6500K.
- Custom: The user can set the values in the matrix as desired.

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

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

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

Setting Matrix Color Transformation Using Basler pylon

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

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

// Set the light source selector so that no correction will be done
Camera.LightSourceSelector.SetValueLightSourceSelector_Off);
// Set the light source selector for daylight (at about 5000K)
Camera.LightSourceSelector.SetValueLightSourceSelector_Daylight);
// Set the light source selector for tungsten lighting
Camera.LightSourceSelector.SetValueLightSourceSelector_Tungsten);
// Set the light source selector for daylight (at about 6500K)
Camera.LightSourceSelector.SetValueLightSourceSelector_Daylight6500K);

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

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting Matrix Color Transformation Using Direct Register Access

To set matrix color transformation:

1. Set the value of the Light Source Selector register to Off, Daylight, Tungsten, Daylight 6500K, or Custom as desired.
2. Set the value of the Color Matrix Factor register as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

8.3.3.1 "Custom" Light Source Setting



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

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

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

Where Gain00, Gain01, etc. are settable values.

Keep in mind that ace Camera Link cameras use sensors with a Bayer pattern filter and that they do not interpolate the pixel values. This means that when the camera is doing matrix correction for a particular pixel, only one actual color value will be available for that pixel. To fill in the other two color values in the 1 x 3 matrix, the camera uses values from neighboring pixels. After making the correction calculations, the camera only transmits the result for the actual pixel color and discards the other two calculated values. For example, if the camera is correcting the value for a red pixel, it will populate the 1 x 3 matrix with the actual value for the red pixel. It will then populate the green and blue positions in the matrix with values from a neighboring green and a neighboring blue pixel. After making the matrix correction calculations, the camera will discard the green and blue values from the result and only transmit the red value.

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

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

A reference article that explains the basics of color matrix transformation for video data can be found under the following address: <http://www.its.bldrdoc.gov/pub/ntia-rpt/04-406/index.php>

Setting Custom Matrix Values Using Basler pylon

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

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

// Select a position in the matrix
Camera.ColorTransformationValueSelector.SetValueColorTransformationValueSelector
_Gain01);
// Set the value for the selected position as a floating point value
Camera.ColorTransformationValue.SetValue (2.11);

// Select a position in the matrix
Camera.ColorTransformationValueSelector.SetValueColorTransformationValueSelector
_Gain12);
// Set the value for the selected position as an integer value
Camera.ColorTransformationValueRaw.SetValue (135);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting Custom Matrix Values Using Direct Register Access

To set Gain XY values in the matrix:

1. Set the value of the Light Source Selector register to Custom.
2. Set the value of the following registers as desired:
 - Color Matrix RGB 2 RGB 00
 - Color Matrix RGB 2 RGB 01
 - Color Matrix RGB 2 RGB 02
 - Color Matrix RGB 2 RGB 10
 - Color Matrix RGB 2 RGB 11
 - Color Matrix RGB 2 RGB 12
 - Color Matrix RGB 2 RGB 20
 - Color Matrix RGB 2 RGB 21
 - Color Matrix RGB 2 RGB 22

For more information about direct register access, see Section 3.2 on [page 20](#).

9 Sensor Bit Depth, Pixel Formats, Tap Geometries, and Clock Speeds

This chapter provides information about the sensor bit depths, pixel formats, Camera Link tap geometries and pixel clock speeds available on the camera.



If you plan to design your own frame grabber, or if you would like complete details regarding the way that pixel data is handled by the camera, refer to the Basler document named *Camera Link Information for Frame Grabber Designers* (AW000990). You can download the document from the Basler website: www.baslerweb.com

9.1 Sensor Bit Depth

As mentioned in the "Functional Description" chapter of this manual, the camera's imaging sensor can be set to acquire image data at either 10-bit or 12-bit depth.

When the sensor is operating at 10-bit depth, it operates significantly faster than at 12-bit depth. So when the sensor is set for 10-bit acquisition, the camera can be operated at a higher maximum frame rate.

Keep in mind that the data depth of the pixel values actually reported out of the camera will depend on the camera's current pixel format setting.

For more information about pixel formats, see Section 9.2 on [page 138](#).

Setting the Sensor Bit Depth Using Basler pylon

You can use the pylon API to set the sensor bit depth from within your application software. The following code snippet illustrates using the API to set the exposure mode:

```
// Set the sensor bit depth to 10
Camera.SensorBitDepth.SetValue(SensorBitDepth_BitDepth10);

// Set the sensor bit depth to 12
Camera.SensorBitDepth.SetValue(SensorBitDepth_BitDepth12);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Sensor Bit Depth Using Direct Register Access

To set the sensor bit depth via direct register access:

1. Set the value of the Sensor Bit Depth register to 10-Bit Depth or 12-Bit Depth.

For more information about direct register access, see Section 3.2 on [page 20](#).

9.2 Pixel Formats

Pixel Formats for Monochrome Cameras

The choice of a pixel format determines the bit depth of the data transmitted from the camera for each pixel in the acquired frames. The pixel formats available on the mono cameras depend on the current setting of the Camera Link Tap Geometry parameter as shown in Table 8.

Camera Link Tap Geometry Setting	Available Pixel Formats		
	Mono 8	Mono 10	Mono 12
1X2-1Y	•	•	•
1X3-1Y	•		
1X4-1Y	•	•	•
1X6-1Y	•		
1X8-1Y	•	•	
1X10-1Y	•		


Table 8: Mono Pixel Formats Available for Each Tap Geometry (• = format available)

Pixel Formats for Color Cameras

The choice of a pixel format determines the bit depth of the data transmitted from the camera for each pixel in the acquired frames. The pixel formats available on the color cameras depend on the current setting of the Camera Link Geometry parameter as shown in Table 9.

Camera Link Tap Geometry Setting	Available Pixel Formats		
	Bayer GB 8	Bayer GB 10	Bayer GB 12
1X2-1Y	•	•	•
1X3-1Y	•		
1X4-1Y	•	•	•
1X6-1Y	•		
1X8-1Y	•	•	
1X10-1Y	•		

Table 9: Pixel Formats Available for Each Geometry (• = format available)

	<p>The pixels output from the color camera are not interpolated in any way; they are output as "raw" data. This means that for each pixel, there is only one color value (i.e., for each red pixel there is only a red value, for each green pixel there is only a green value, and for each blue pixel there is only a blue value).</p> <p>Color cameras do not output interpolated RGB values for each pixel, even when they are set for a multiple tap Camera Link tap geometry such as 1X3-1Y or 1X6-1Y.</p>
---	---

Pixel Format Interaction with the Sensor Bit Depth Setting

The camera's sensor can capture image data at 12-bit depth or at 10-bit depth (depending on the Sensor Bit Depth setting). There is an interaction between the Sensor Bit Depth setting and the Pixel Format setting as described in Table 10.

Sensor Bit Depth Setting	Pixel Format Setting		
	Mono 8 or Bayer GB 8	Mono 10 or Bayer GB 10	Mono 12 or Bayer GB 12
12 bits	8 most significant bits of data from the sensor are transmitted	10 most significant bits of data from the sensor are transmitted	All 12 bits of data from the sensor are transmitted
10 bits	8 most significant bits of data from the sensor are transmitted	All 10 bits of data from the sensor are transmitted	All 10 bits of data from the sensor are transmitted and 2 zero-packed least significant bits are added

Table 10: Interaction Between the Sensor Bit Depth and the Pixel Format

For more information about the sensor bit depth, see Section 9.1 on [page 137](#).

Setting the Pixel Format Using Basler pylon

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

Mono cameras

```
// Set pixel format to Mono 8
Camera.PixelFormat.SetValue(PixelFormat_Mono8);
// Set pixel format to Mono 10
Camera.PixelFormat.SetValue(PixelFormat_Mono10);
// Set pixel format to Mono 12
Camera.PixelFormat.SetValue(PixelFormat_Mono12);
```

Color Cameras:

```
// Set pixel format to Bayer GB 8
Camera.PixelFormat.SetValue(PixelFormat_BayerGB8);
// Set pixel format to Bayer GB 10
Camera.PixelFormat.SetValue(PixelFormat_BayerGB10);
// Set pixel format to Bayer GB 12
Camera.PixelFormat.SetValue(PixelFormat_BayerGB12);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Pixel Format Using Direct Register Access

To set the sensor pixel format:

Mono cameras

1. Set the value of the Pixel Format register to Mono 8, Mono 10, or Mono 12 as desired.

Color cameras

1. Set the value of the Pixel Format register to Bayer GB 8, Bayer GB 10, or Bayer GB 12 as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

9.3 Camera Link Tap Geometry

9.3.1 Overview

The Camera Link tap geometry determines how the data that is read out of the imaging sensor will be transmitted from the camera to the frame grabber in your host PC via the Camera Link interface. The selection of a camera link tap geometry also determines whether your camera will be operating in the base, medium, or full Camera Link configuration and which Camera Link connectors on the camera will be used to transmit pixel data. Table 11 indicates how the Camera Link interface will operate with each available tap geometry setting.

The X2, X3, X4, X6, X8, or X10 in the tap geometry names indicates the number of Camera Link taps that will be used for a given configuration (i.e., X2 means 2 taps, X3 means 3 taps, etc.). As a general rule of thumb, a camera will have a higher maximum allowed frame rate when it is operating with a tap geometry that uses Camera Link taps.

Tap Geometry Setting	Camera Link Configuration	Camera Link Connectors Used to Transmit Data
1X2-1Y	Base	Base Only
1X3-1Y	Base	Base Only
1X4-1Y	Medium	Base and Medium/Full
1X6-1Y	Medium	Base and Medium/Full
1X8-1Y	Full	Base and Medium/Full
1X10-1Y	Full	Base and Medium/Full

Table 11: Camera Link Operation at Various Tap Geometry Settings

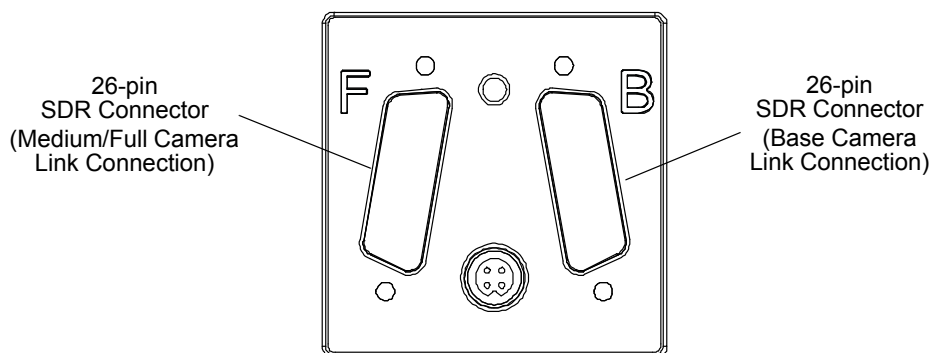


Fig. 40: Camera Link Connections

The tap geometries available by model are as shown in Table 12.

Tap Geometry Setting	Camera Model	
	acA2000-340 km/kmNIR/kc	acA2040-180 km/kmNIR/kc
1X2-1Y	•	•
1X3-1Y	•	•
1X4-1Y	•	•
1X6-1Y	•	•
1X8-1Y	•	•
1X10-1Y	•	•

Table 12: Available Tap Geometries by Model (• = geometry available)

For more information about the camera's maximum allowed frame rate, see Section 7.8 on [page 115](#).

This section describes the basics of Camera Link tap geometry. For more complete tap geometry details, refer to the Basler document named *Camera Link Information for Frame Grabber Designers* (AW000990) You can download the document from the Basler website: www.baslerweb.com

9.3.2 1X2-1Y Tap Geometry Description

The characteristics of the 1X2-1Y tap geometry are:

- On each cycle of the Camera Link pixel clock, the data for two pixels are transmitted via the Camera Link interface. This is commonly referred to as a "two tap" Camera Link configuration.
- The camera will begin transmitting data from line one. It will transmit the data for the first two pixels in line one on the first Camera Link pixel clock cycle, the next two pixels in line one on the second pixel clock cycle, the next two pixels in line one on the third clock cycle, and so on until the line is complete.
- When line one is complete, transmission of line two will begin and will proceed in a similar fashion. Data transmission will continue line-by-line until all of the data for the image have been transmitted.

When the camera is set for the 1X2-1Y tap geometry, it uses the base Camera Link configuration.

The 1X2-1Y transmission scheme is shown graphically in Figure 41.

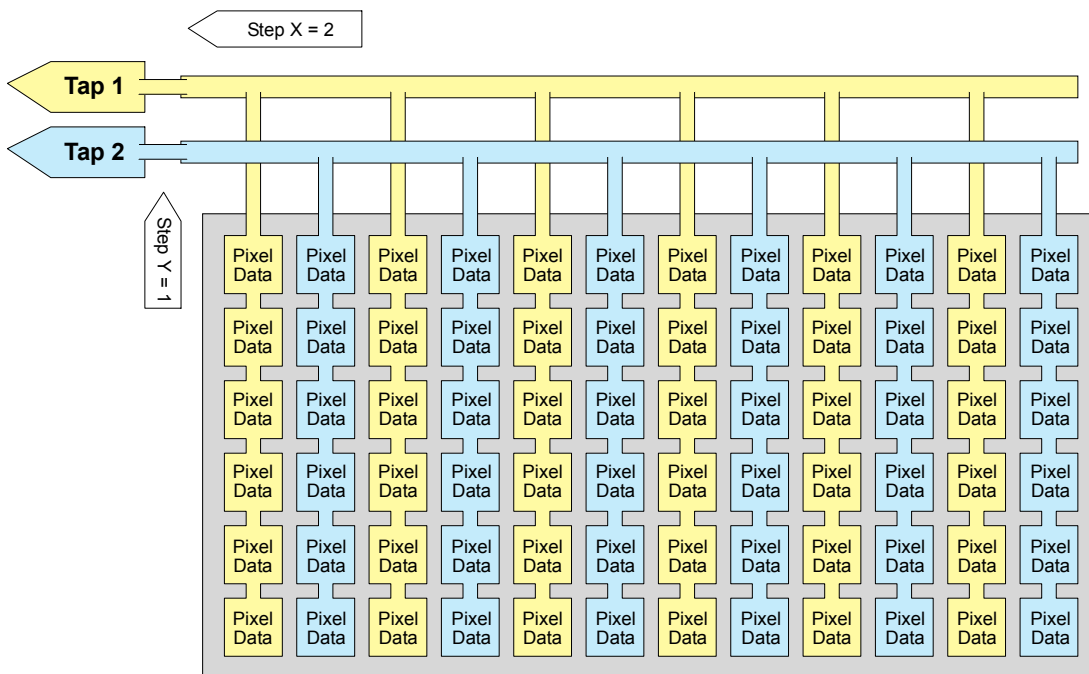


Fig. 41: 1X2-1Y Tap Geometry

9.3.3 1X3-1Y Tap Geometry Description

The characteristics of the 1X3-1Y tap geometry are:

- On each cycle of the Camera Link pixel clock, the data for three pixels are transmitted via the Camera Link interface. This is commonly referred to as a "three tap" Camera Link configuration.
- The camera will begin transmitting data from line one. It will transmit the data for the first three pixels in line one on the first Camera Link pixel clock cycle, the next three pixels in line one on the second pixel clock cycle, the next three pixels in line one on the third clock cycle, and so on until the line is complete.
- When line one is complete, transmission of line two will begin and will proceed in a similar fashion. Data transmission will continue line-by-line until all of the data for the image have been transmitted.

When the camera is set for the 1X3-1Y tap geometry, it uses the base Camera Link configuration.

The 1X3-1Y transmission scheme is shown graphically in Figure 42.

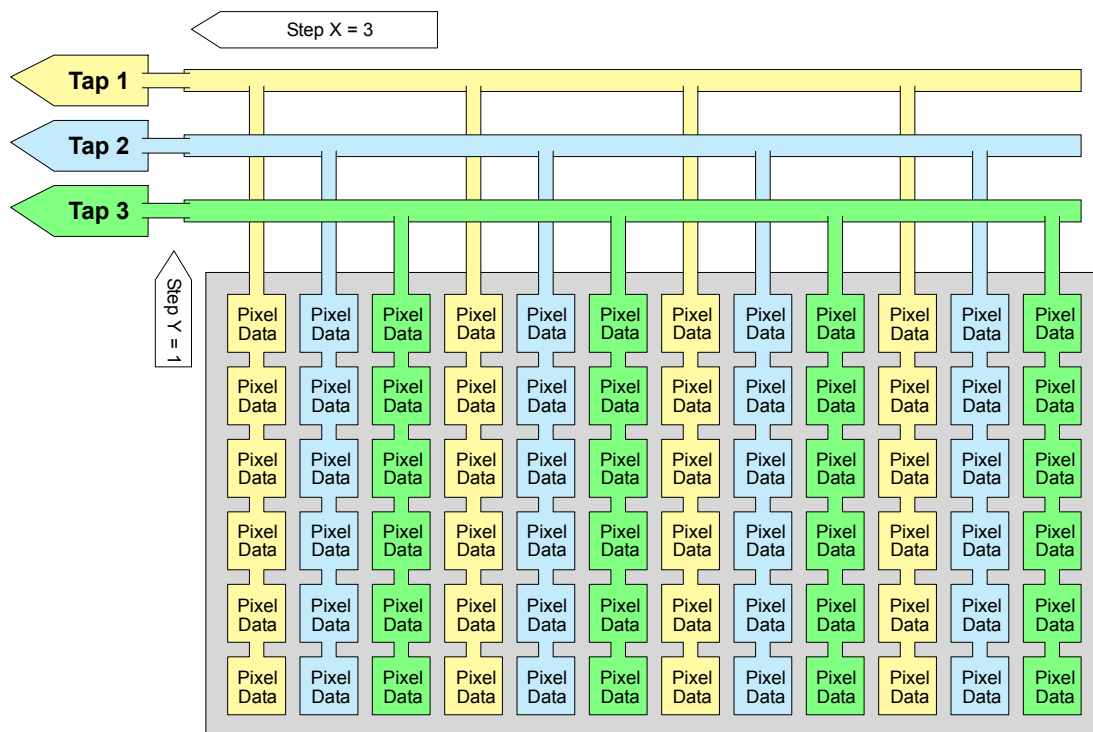


Fig. 42: 1X3-1Y Tap Geometry

9.3.4 1X4-1Y Tap Geometry Description

The characteristics of the 1X4-1Y tap geometry are:

- On each cycle of the Camera Link pixel clock, the data for four pixels are transmitted via the Camera Link interface. This is commonly referred to as a "four tap" Camera Link configuration.
- The camera will begin transmitting data from line one. It will transmit the data for the first four pixels in line one on the first Camera Link pixel clock cycle, the next four pixels in line one on the second pixel clock cycle, the next four pixels in line one on the third clock cycle, and so on until the line is complete.
- When line one is complete, transmission of line two will begin and will proceed in a similar fashion. Data transmission will continue line-by-line until all of the data for the image have been transmitted.

When the camera is set for the 1X4-1Y tap geometry, it uses the medium Camera Link configuration.

The 1X4-1Y transmission scheme is shown graphically in Figure 43.

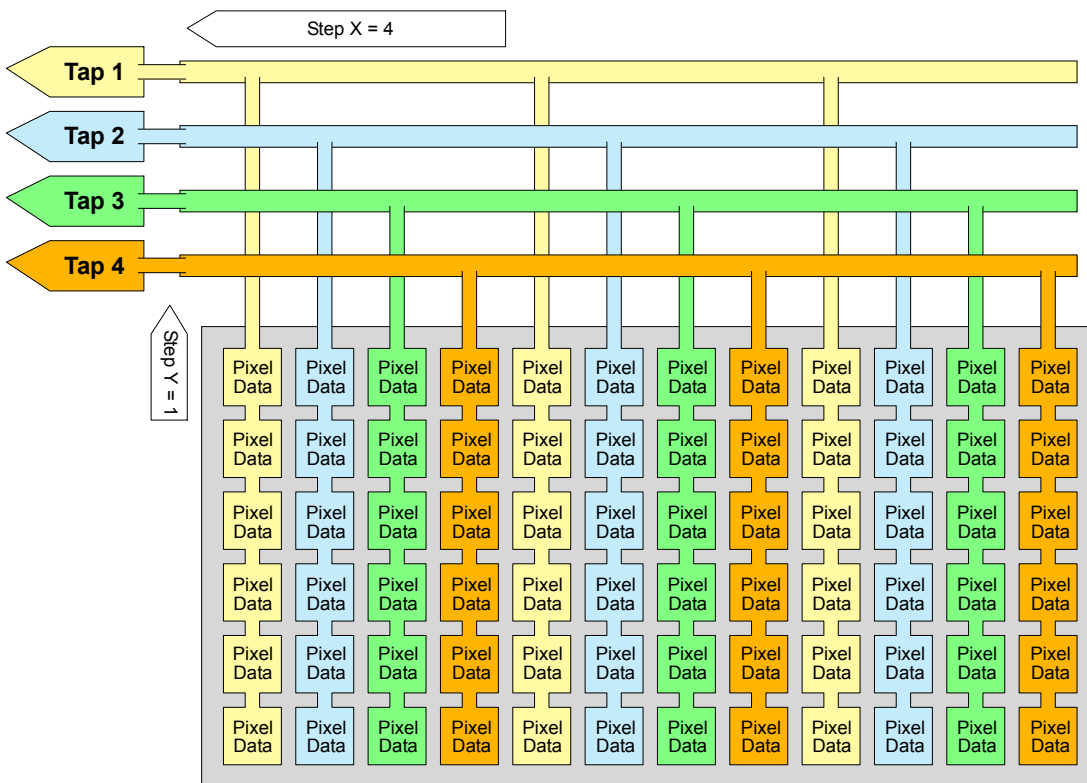


Fig. 43: 1X4-1Y Tap Geometry

9.3.5 1X6-1Y Tap Geometry Description

The characteristics of the 1X6-1Y tap geometry are:

- On each cycle of the Camera Link pixel clock, the data for six pixels are transmitted via the Camera Link interface. This is commonly referred to as a "six tap" Camera Link configuration.
- The camera will begin transmitting data from line one. It will transmit the data for the first six pixels in line one on the first Camera Link pixel clock cycle, the next six pixels in line one on the second pixel clock cycle, the next six pixels in line one on the third clock cycle, and so on until the line is complete.
- When line one is complete, transmission of line two will begin and will proceed in a similar fashion. Data transmission will continue line-by-line until all of the data for the image have been transmitted.

When the camera is set for the 1X6-1Y tap geometry, it uses the medium Camera Link configuration.

The 1X6-1Y transmission scheme is shown graphically in Figure 44.

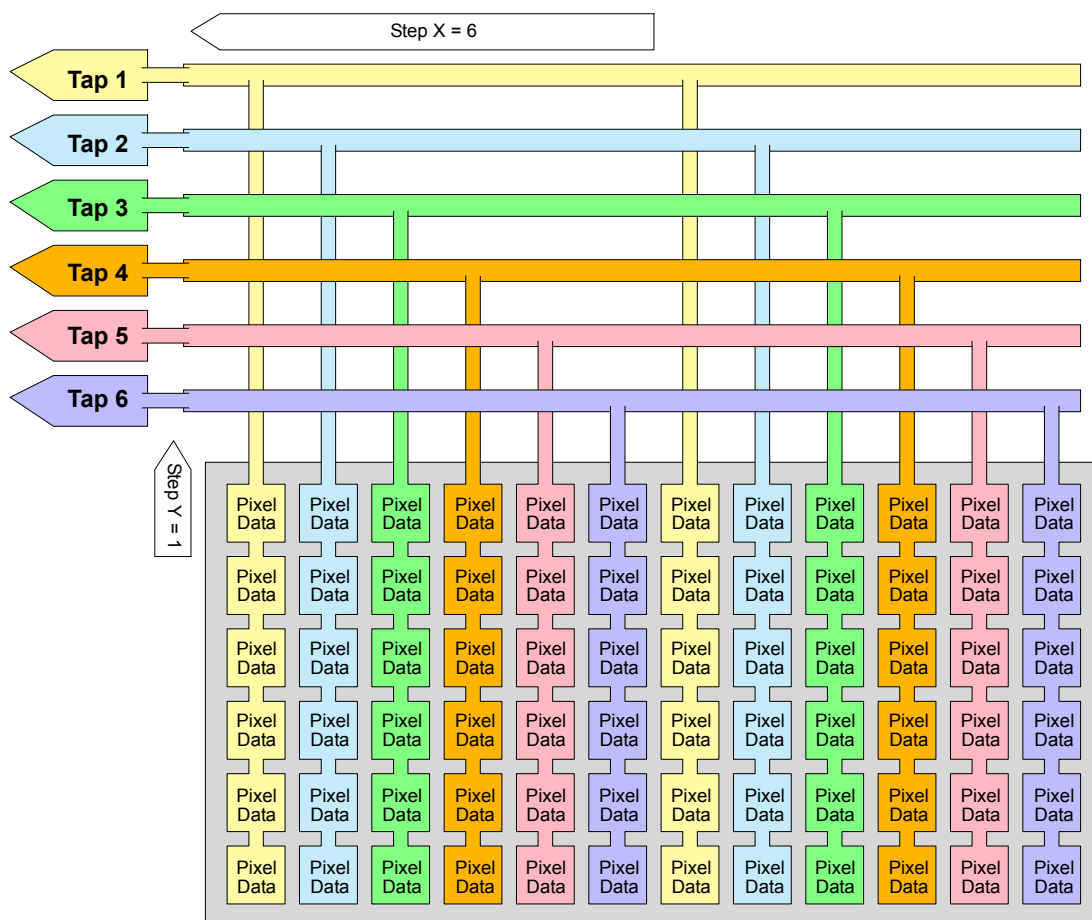


Fig. 44: 1X6-1Y Tap Geometry

9.3.6 1X8-1Y Tap Geometry Description

The characteristics of the 1X8-1Y tap geometry are:

- On each cycle of the Camera Link pixel clock, the data for eight pixels are transmitted via the Camera Link interface. This is commonly referred to as an "eight tap" Camera Link configuration.
- The camera will begin transmitting data from line one. It will transmit the data for the first eight pixels in line one on the first Camera Link pixel clock cycle, the next eight pixels in line one on the second pixel clock cycle, the next eight pixels in line one on the third clock cycle, and so on until the line is complete.
- When line one is complete, transmission of line two will begin and will proceed in a similar fashion. Data transmission will continue line-by-line until all of the data for the image have been transmitted.

When the camera is set for the 1X8-1Y tap geometry, it uses the full Camera Link configuration.

The 1X8-1Y transmission scheme is shown graphically in Figure 45.

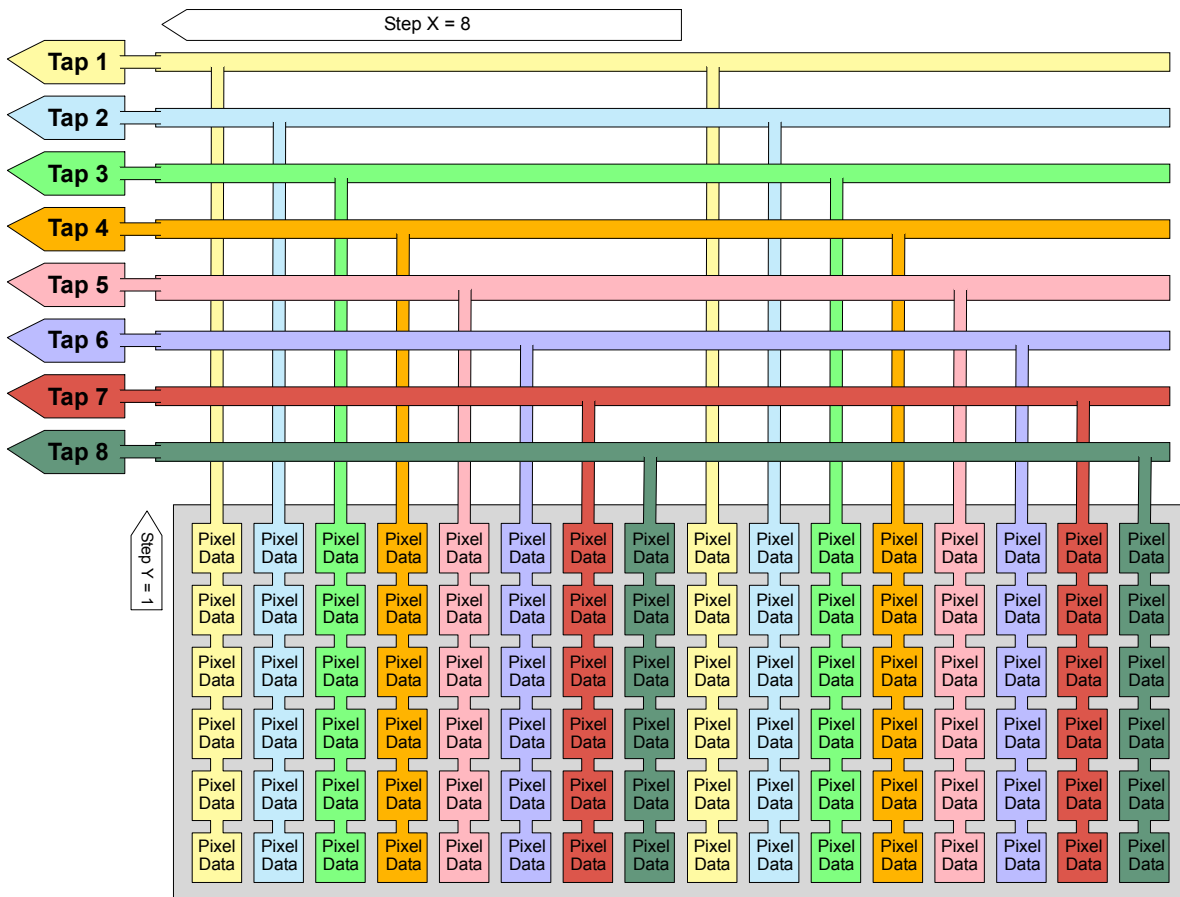


Fig. 45: 1X8-1Y Tap Geometry

9.3.7 1X10-1Y Tap Geometry Description

The characteristics of the 1X10-1Y tap geometry are:

- On each cycle of the Camera Link pixel clock, the data for ten pixels are transmitted via the Camera Link interface. This is commonly referred to as an "ten tap" Camera Link configuration.
- The camera will begin transmitting data from line one. It will transmit the data for the first ten pixels in line one on the first Camera Link pixel clock cycle, the next ten pixels in line one on the second pixel clock cycle, the next ten pixels in line one on the third clock cycle, and so on until the line is complete.
- When line one is complete, transmission of line two will begin and will proceed in a similar fashion. Data transmission will continue line-by-line until all of the data for the image have been transmitted.

When the camera is set for the 1X10-1Y tap geometry, it uses the full Camera Link configuration.

The 1X10-1Y transmission scheme is shown graphically in Figure 46.

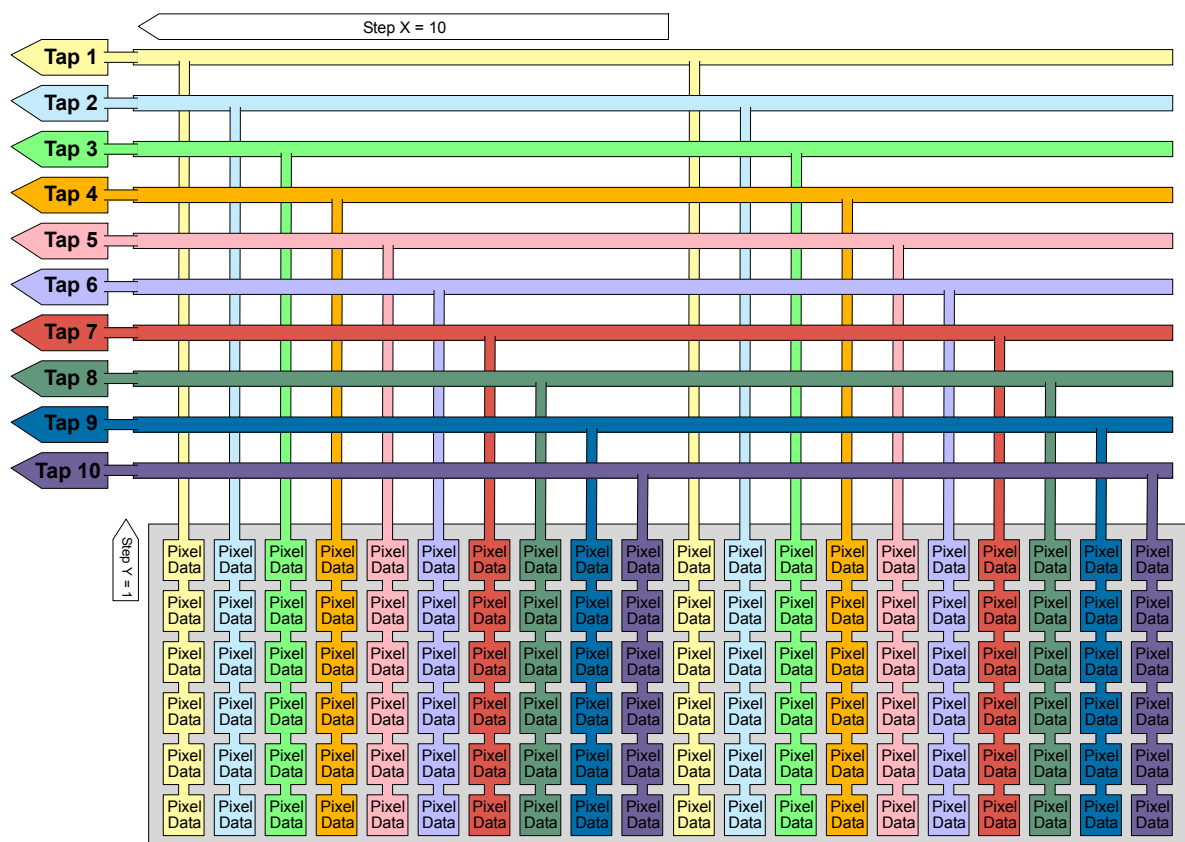


Fig. 46: 1X10-1Y Tap Geometry

9.3.8 Setting the Tap Geometry

Setting the Tap Geometry Using Basler pylon

You can use the pylon API to set the Camera Link tap geometry from within your application software. The following code snippet illustrates using the API to set the tap geometry:

```
// Set the tap geometry to 1X2-1Y
Camera.ClTapGeometry.SetValue(ClTapGeometry_Geometry1X2_1Y);

// Set the tap geometry to 1X3-1Y
Camera.ClTapGeometry.SetValue(ClTapGeometry_Geometry1X3_1Y);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Tap Geometry Using Direct Register Access

To set the Camera Link tap geometry:

1. Set the value of the CL Tap Geometry register for 1X2-1Y, 1X3-1Y, 1X4-1Y, 1X6-1Y, 1X8-1Y, or 1X10-1Y as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

9.4 Camera Link Pixel Clock Speed

The camera features selectable Camera Link pixel clock speeds. The pixel clock speed determines the rate at which pixel data will be transmitted from the camera to the frame grabber in your PC via the Camera Link interface. The four available pixel clock speeds are: 32.5 MHz, 48 MHz, 65 MHz, and 82 MHz. The default clock speed is 82 MHz.

Setting the camera for a higher pixel clock speed will increase the rate at which image data is transferred from the camera to the frame grabber. Some frame grabbers, however, can't operate at the higher clock speeds. So it is important that you determine the maximum clock speed that your frame grabber can handle and that you set the camera's speed no higher than the frame grabber's maximum.

Keep in mind that if you set the camera for one of the lower pixel clock speeds, it may limit the camera's maximum allowed frame acquisition rate.

If you change the clock speed while the camera is in the process of acquiring images:

- Triggering and image acquisition will stop.
- Any acquired image that is already in the process of being transferred will be delivered.
- The camera's clock speed will be changed internally.
- Once the change is complete, triggering and image acquisition will resume.

The Camera Link clock speed setting will be stored in the camera's configuration sets. This means, for example, that if you have a different clock speed setting stored in user set 1 and user set 2 and you change the active set from user set 1 to user set 2, the clock speed will change.

For more information about the maximum allowed frame acquisition rate, see Section 7.8 on [page 115](#).

For more information about configuration sets, see Section 10.17 on [page 249](#).

Setting the Camera Link Pixel Clock Using Basler pylon

You can use the pylon API to set the Camera Link pixel clock speed from within your application software. The following code snippet illustrates using the API to set the clock speed:

```
// Set the Camera Link pixel clock speed 32.5
Camera.ClPixelClock.SetValue(ClPixelClock_PixelClock32_5);

// Set the Camera Link pixel clock speed to 48
Camera.ClPixelClock.SetValue(ClPixelClock_PixelClock48);
```



You can use the pylon API to set the pixel clock speed to 32.5, 48, 65, or 82 MHz. These are the only valid values for the pixel clock speed. If you attempt to use the API to set the clock speed to a value other than these, the camera will automatically move the setting to the nearest lower valid speed.

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Camera Link Pixel Clock Using Direct Register Access

To set the Camera Link pixel clock speed:

1. Set the value of the CL Pixel Clock register to 32.5 MHz, 48 MHz, 65 MHz, or 82 MHz.

For more information about direct register access, see Section 3.2 on [page 20](#).

10 Features

This chapter provides detailed information about the standard features available on each camera. It also includes an explanation of their operation and the parameters associated with each feature.

10.1 Gain



This section describes the basic theory of gain and how gain can be adjusted "manually", i.e., by setting the value of the gain raw parameter.

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

For more information about auto functions in general, see Section 10.11 on [page 219](#).

For more information about the Gain Auto function in particular, see Section 10.11.4 on [page 229](#).

The camera's gain setting is adjustable. As shown in Figure 47, 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), 1023 (in modes that output 10 bits per pixel), or 4095 (in modes that output 12 bits per pixel) 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.

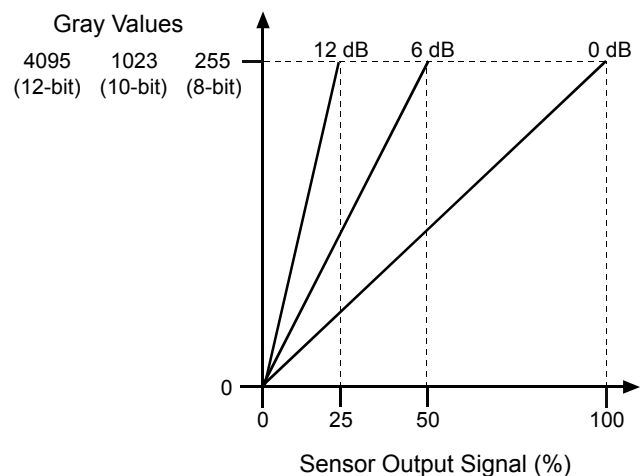


Fig. 47: Gain in dB

The camera's gain is determined by the value of the Gain Raw parameter. Raw gain is adjusted on an integer scale. The minimum setting is 33 and the maximum setting is 511. If you know the current setting for the Gain Raw parameter, you can calculate the camera's gain in dB using the following formula:

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

For example, if the current Gain Raw setting is 128, then:

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

$$\text{Gain in dB} = 12.0$$

Setting the Gain Using Basler pylon

To set the Gain Raw parameter value:

1. Set the Gain Selector to All.
2. Set the Gain Raw parameter to your desired value.

You can use the pylon API to set the Gain Selector and the Gain Raw parameter values from within your application software. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set the gain
Camera.GainSelector.SetValue(GainSelector_All);
Camera.GainRaw.SetValue(128);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Gain Using Direct Register Access

To set the gain:

1. Set the value of the Gain All register.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.2 Black Level

Adjusting the camera's black level will result in an offset to the pixel values output by the camera. Increasing the black level setting will result in a positive offset in the digital values output for the pixels. Decreasing the black level setting will result in a negative offset in the digital values output for the pixels.

The black level can be set on an integer scale ranging from 0 to 255.

If the camera is set for a pixel format that yields an 8-bit pixel depth, an increase of 16 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. And a decrease of 16 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

If the camera is set for a pixel format that yields a 10-bit pixel depth, an increase of 4 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. A decrease of 4 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

If the camera is set for a pixel format that yields a 12-bit pixel depth, an increase of 1 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. A decrease of 1 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

Setting the Black Level Using Basler pylon

To set the black level:

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

You can use the pylon API to set the Black Level Selector and the Black Level Raw parameter values from within your application software. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set the black level
Camera.BlackLevelSelector.SetValue(BlackLevelSelector_All);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Black Level Using Direct Register Access

To set the black level:

1. Set the value of the Black Level All register.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.3 Remove Parameter Limits

For each camera feature, the allowed range of any associated parameter values is normally limited. The factory limits are designed to ensure optimum camera operation and, in particular, good image quality. For special camera uses, however, it may be helpful to set parameter values outside of the factory limits.

The remove parameter limits feature lets you remove the factory limits of parameters associated with certain camera features. When the factory limits are removed, the parameter values can be set within extended limits. Typically, the range of the extended limits is dictated by the physical restrictions of the camera's electronic components, such as the absolute limits of the camera's variable gain control.

The values for any extended limits can be determined by using the Basler pylon Viewer or from within your application via the pylon API.

Currently, the limits can be removed from:

- The Gain feature.
Removing the parameter limits of the gain feature will only remove the lower limit. The lower limit of the Gain parameter will be reduced to 0. (For those cameras where the lower limit is already 0, removing the limits has no effect.)
- The Exposure Time feature.
Removing the limits of the exposure time feature will change the minimum allowed exposure time to 2 μ s.

For more information about the gain feature, see Section 10.1 on [page 152](#).

For more information about the exposure time feature, see Section 7.4 on [page 89](#).

Removing Parameter Limits

To remove the limits of a parameter:

1. Use the Parameter Selector to select the parameter whose limits you want to remove.
2. Set the value of the Remove Limits parameter.

You can set the Parameter Selector and the value of the Remove Limits parameter from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the feature whose factory limits will be removed.
Camera.ParameterSelector.SetValue(ParameterSelector_Gain);
// Remove the limits for the selected feature.
Camera.RemoveLimits.SetValue(true);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

10.4 Image Area of Interest (AOI)

The image area of interest (Image AOI or AOI for short) feature lets you specify a portion of the sensor array and after each image is acquired, only the pixel information from the specified portion of the array will be read out of the sensor and transmitted from the camera.

The area of interest is referenced to the top left corner of the sensor array. The top left corner is designated as column 0 and row 0 as shown in Figure 48.

The location and size of the area of interest is defined by declaring an offset X, a width, an offset Y, and a height. For example, suppose that you specify the offset X as 10, the width as 16, the offset Y as 6, and the height as 10. The area of the array that is bounded by these settings is shown in Figure 48.

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

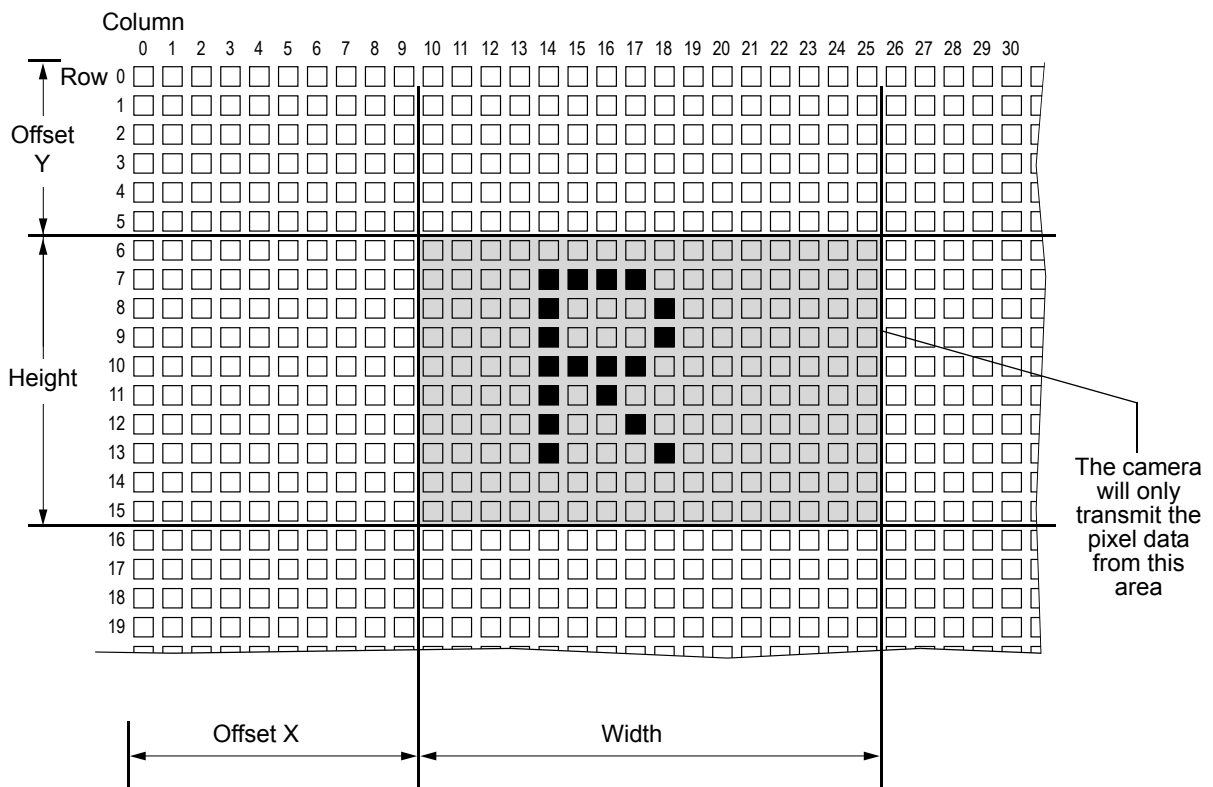


Fig. 48: Area of Interest

One of the main advantages of the AOI feature is that decreasing the size of the AOI can increase the camera's maximum allowed acquisition frame rate.

The AOI feature also includes Center X and a Center Y capabilities. When Center X is enabled, the camera will automatically center the AOI along the sensor's X axis (and will disable the Offset X

setting). When Center Y is enabled, the camera will automatically center the AOI along the sensor's Y axis (and will disable the Offset Y setting).

For more information about how changing the AOI size affects the maximum allowed frame rate, see Section 7.8 on [page 115](#).

10.4.1 Setting the Image AOI

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

When you are setting the camera's area of interest, you must follow these guidelines:

On all camera models:

- The sum of the Offset X setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the acA2000-340km, the sum of the Offset X setting plus the Width setting must not exceed 2048.
- The sum of the Offset Y setting plus the Height setting must not exceed the height of the camera's sensor. For example, on the acA2000-340km, the sum of the Offset Y setting plus the Height setting must not exceed 1088.

On monochrome camera models:

- The Offset Y and Height parameters can each be set in increments of 1.
- The Offset X and Width parameters can be set in increments of:
 - 2 on a camera set for the 1X2-1Y Camera Link tap geometry
 - 3 on a camera set for the 1X3-1Y Camera Link tap geometry
 - 4 on a camera set for the 1X4-1Y Camera Link tap geometry
 - 6 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 8 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 10 on a camera set for the 1X10-1Y Camera Link tap geometry

On color camera models:

- The Offset Y and Height parameters can each be set in increments of 2.
- The Offset X and Width parameters can be set in increments of:
 - 2 on a camera set for the 1X2-1Y Camera Link tap geometry
 - 6 on a camera set for the 1X3-1Y Camera Link tap geometry
 - 4 on a camera set for the 1X4-1Y Camera Link tap geometry
 - 6 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 8 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 10 on a camera set for the 1X10-1Y Camera Link tap geometry

For more information about Camera Link tap geometries, see Section 9.3 on [page 141](#).



Your frame grabber may place additional restrictions on how the AOI size must be set. Check the documentation included with your frame grabber to determine its AOI requirements.

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 about binning, see Section 10.10 on [page 214](#).

Setting the Image AOI Using Basler pylon

You can set the Offset X, Offset Y, Width, and Height parameter values from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to get the maximum allowed settings and the increments for the Width and Height parameters. They also illustrate setting the Offset X, Offset Y, Width, and Height parameter values and enabling automatic AOI centering.

```
int64_t widthMax = Camera.Width.GetMax();
int64_t widthInc = 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);

// Enable automatic X and Y centering
Camera.CenterX.SetValue(true);
Camera.CenterY.SetValue(true);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Image AOI Using Direct Register Access

To set the AOI Offset X, Offset Y, Width, and Height parameters:

1. Set the value of the Offset X register.
2. Set the value of the Offset Y register.
3. Set the value of the Width register.
4. Set the value of the Height register.

To enable Center X and Center Y:

1. Set the value of the Center X register.
2. Set the value of the Center Y register.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.5 Stacked Zone Imaging

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

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

The Offset X and Width parameters are used to begin the process of setting up stacked zone imaging. Since all of the zones must be the same width and all of the zones must be vertically aligned, these two parameters define the left and right borders for all of the zones as shown in Figure 49 on [page 161](#). In the figure, Offset X is set to 10 and the Width is set to 16.

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

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

In Figure 49, for example, three zones have been enabled: zone 1, zone 2, and zone 3.

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

For zone 1:

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

For zone 2:

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

For zone 3:

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

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

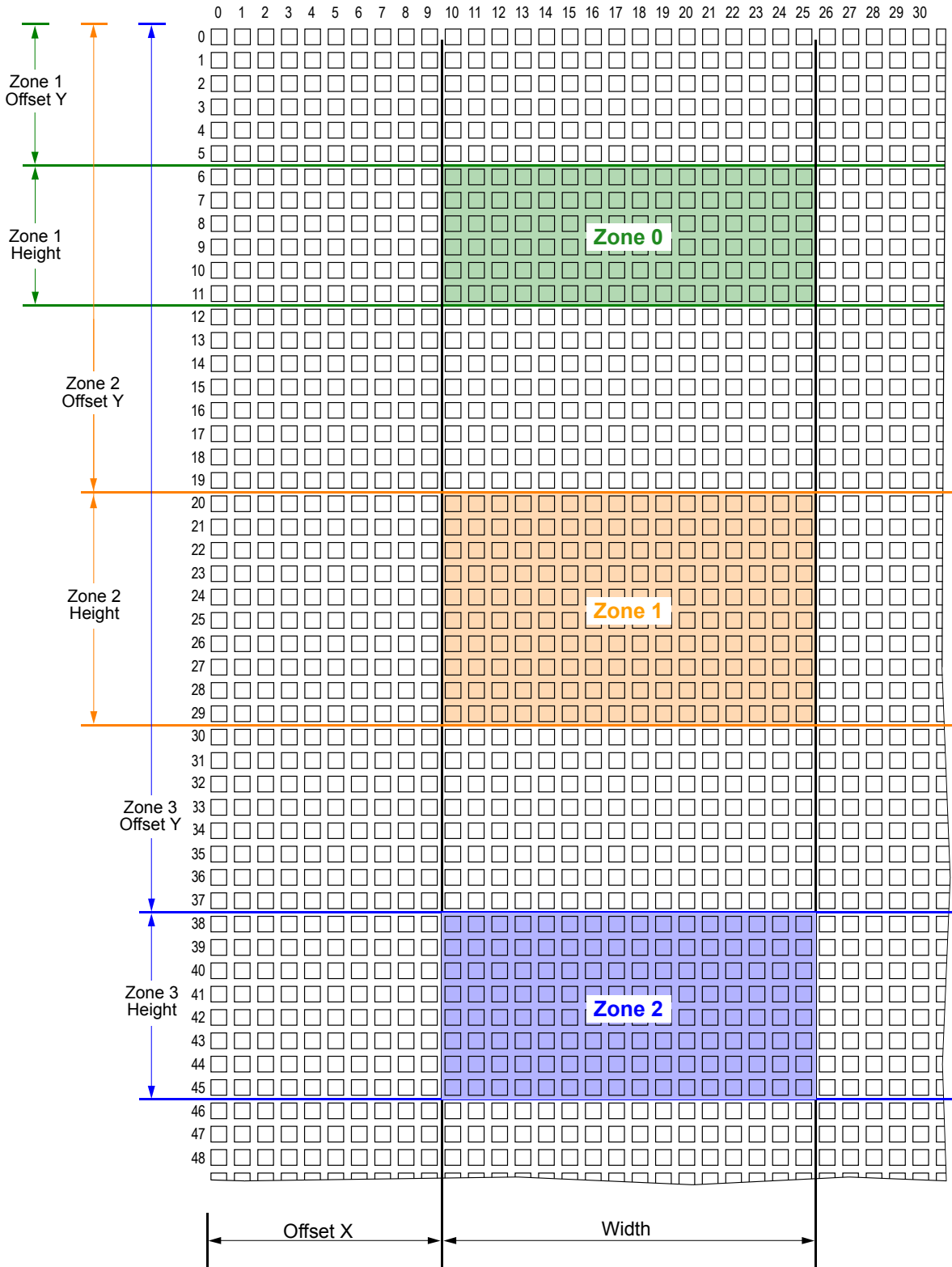


Fig. 49: Stacked Zone Imaging

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

- You are not required to enable the zones in sequence. For example, you can enable zones 2, 4, and 6 and not enable zones 1, 3, and 5.
- You do not need to order the zones from top to bottom on the sensor. For example, you could place zone 1 near the bottom of the sensor, zone 3 near the top, and zone 2 in the middle.

Note that the camera always reads out and transmits the zones starting from the top of the sensor and going to the bottom, regardless of how the zone numbers are ordered. So the lines in the transmitted images will always be ordered from top to bottom in relation to the sensor.

- The zones can be set so that they overlap. When this happens, the camera will internally transform the overlapped zones into a single large zone that will be read out and transmitted as if it were simply a single large zone. (The lines included in the overlapping area will only be read out and transmitted once.)

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

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

10.5.1 Setting Stacked Zone Imaging

Guidelines

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

On all camera models:

- The sum of the Offset X setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the acA2000-340km, the sum of the Offset X setting plus the Width setting must not exceed 2048.
- For any given zone, the sum of the Stacked Zone Imaging Zone Offset Y setting plus the Stacked Zone Imaging Zone Height setting must not exceed the height of the camera's sensor. For example, on the acA2000-340km, the sum of the Stacked Zone Imaging Zone Offset Y setting plus the Stacked Zone Imaging Zone Height setting must not exceed 1088.

On monochrome camera models:

- The Stacked Zone Imaging Zone Offset Y and Stacked Zone Imaging Zone Height parameters for any given zone can be set in increments of 1.
- The Offset X and Width parameters can be set in increments of:
 - 2 on a camera set for the 1X2-1Y Camera Link tap geometry
 - 3 on a camera set for the 1X3-1Y Camera Link tap geometry
 - 4 on a camera set for the 1X4-1Y Camera Link tap geometry
 - 6 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 8 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 10 on a camera set for the 1X10-1Y Camera Link tap geometry

On color camera models:

- The Stacked Zone Imaging Zone Offset Y and Stacked Zone Imaging Zone Height parameters for any given zone can be set in increments of 2.
- The Offset X and Width parameters can be set in increments of:
 - 2 on a camera set for the 1X2-1Y Camera Link tap geometry
 - 3 on a camera set for the 1X3-1Y Camera Link tap geometry
 - 4 on a camera set for the 1X4-1Y Camera Link tap geometry
 - 6 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 8 on a camera set for the 1X6-1Y Camera Link tap geometry
 - 10 on a camera set for the 1X10-1Y Camera Link tap geometry

Setting Stacked Zone Imaging Using Basler pylon

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

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

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

// Set zone 1

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

// Set zone 2

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

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting Stacked Zone Imaging Using Direct Register Access

To enable stacked zone imaging:

1. Set the value of the Stacked Zone Imaging Enable register to 1 (true).

To set the Offset X and Width parameters for the zones:

1. Set the value of the Offset X register.
2. Set the value of the Width register.

To set up zone 1:

1. Enable zone 1 by setting the value of the Stacked Zone Imaging Zone 1 Enable register to 1 (true).
2. Set the Y offset for zone 1 by setting the value of the Stacked Zone Imaging Zone 1 Offset Y register.
3. Set the height for zone 1 by setting the value of the Stacked Zone Imaging Zone 1 Height register.

Zones 2 through 8 are set up in similar fashion.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.6 Sequencer



The sequencer feature will not work, if the auto functions feature is enabled. For more information about the auto functions feature, see Section 10.11 on [page 219](#).

The sequencer feature allows you to apply specific sets of configuration parameter settings, called sequence sets, to a sequence of image acquisitions. As the images are acquired, one sequence set after the other is applied. This enables you to respond to different imaging requirements and conditions, that may, for example, result from changing illumination, while a sequence of images is acquired.

Three sequence advance modes (auto, controlled and free selection sequence advance modes) provide different methods for advancing from one sequence set to the next (see below for details).

The number of sequence sets available depends on the selected sequence advance mode.

The Sequencer and the Active Configuration Set

During operation, the camera is controlled by a set of configuration parameters that reside in the camera's volatile memory. This set of parameters is known as the active configuration set or "active set" for short. When you use the pylon API or the pylon Viewer to make a change to a camera parameter such as the Gain, you are making a change to the active set. Since the active set controls camera operation, you will see a change in camera operation when you change a parameter in the active set. For more information about the active set, see the "Configuration Sets" section on [page 249](#).

The parameters in the active set can be divided into two types (as shown in Figure 50 on [page 166](#)):

- "non-sequence" parameters
The values of the non-sequence parameters can't be changed using the sequencer feature.
- "sequence" parameters
The values of the sequence parameters can be set very quickly by using sequence sets: Because the sequence sets reside in the camera, you can replace the values in the active set with values from one of the sequence sets almost instantaneously as images are acquired. Using the sequencer feature doesn't put any additional load on the camera, i.e. enabling the sequencer doesn't slow down the camera's frame rate.

The sequence set currently defining the parameter values of the active set is also called the "current set".

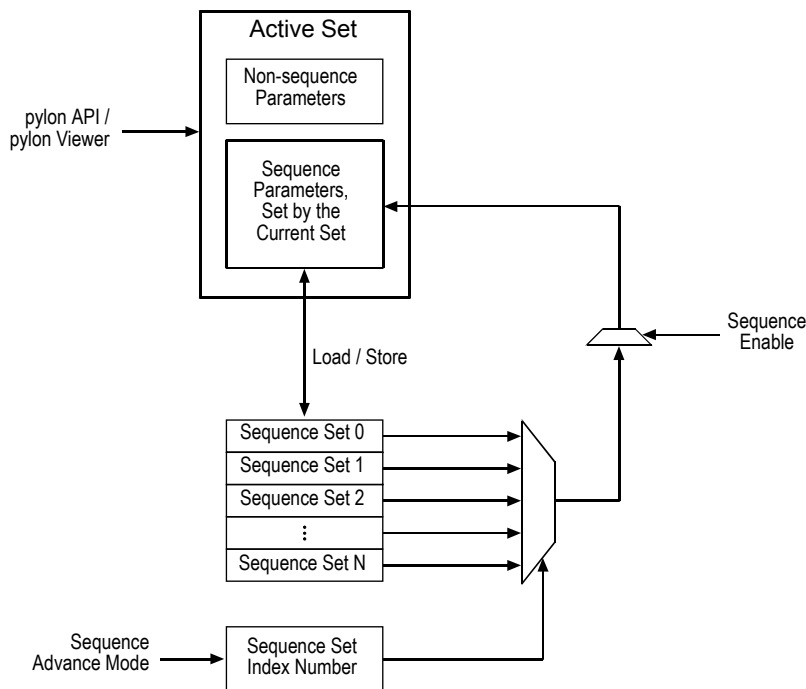


Fig. 50: Sequence Feature Block Diagram

The following parameters are included in each sequence set:

Gain	Enable Zone
Black Level	Zone Y Offset
Pixel Format	Zone Height
Test Image	Light Source Selector**
Width	Balance Ratio
Height	Color Transformation Value**
X Offset	Color Transformation Matrix Factor**
Y Offset	Exposure Time
Center X	Enable Acquisition Frame Rate
Center Y	Acquisition Frame Rate
Binning Horizontal	Sync User Output Value
Binning Vertical	Sequence Set Executions*
Decimation Vertical	LUT Enable
Enable Stacked Zone Imaging	

* This parameter is only available in auto sequence advance mode.

** This parameter is only available for color cameras.

Sequence Set Configuration

When you enable the sequencer feature for the first time, the sequence sets contain the parameter settings that are currently in the active set of the camera. If the camera was started for the first time, this would be the standard factory setup. If you have specified a user set, the sequencer features would be populated with the values specified in that user set.

For more information about configuration setups, see Section 10.17 on [page 249](#).

To prepare the sequencer for your specific application, you have to change the default settings of the sequence parameters to suit your needs and store the sequence sets in the camera's memory. Each sequence set is identified by a sequence set index number starting from zero. After having been stored, the sequence sets are available for use by the sequencer feature.

Some sequence advance modes require the storing of additional settings—for example, the total number of sequence sets you want to use, the number of consecutive uses of a sequence set, or the source to control sequence set advance. For details about populating sequence sets and making related settings see the sections below explaining the sequence advance modes.



When the sequencer feature is enabled, the sequence parameter values of the current sequence set can't be read or changed using the pylon API or the pylon Viewer. Only those sequence parameter values will be displayed that were active before the sequencer was enabled. You will not be able to "see" the parameter values set by the current set.

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



Because the sequence sets only reside in volatile memory, they are lost if the camera is reset or switched off. If you are using the sequencer feature, you must populate the sequence sets after each camera reset or startup.

Note also that sequence sets can't be saved in user sets.

Sequence Advance

As explained above, a sequence set can only control the operation of the camera after its parameter values have been loaded into the active set. The loading into the active set and therefore the selection of a sequence set as the current set for a specific image acquisition are performed according to the selected sequence advance mode. The selection of a sequence set as the current set is always linked to the frame start trigger signals unless software commands are used (see below). Accordingly, a sequence advance mode provides a scheme for advancing from one sequence set to the next as frames are triggered.

The following three sequence advance modes are available:

- **Auto sequence advance mode:** Sequence set advance is controlled by the camera. The camera will cycle through the available sequence sets in ascending sequence set index numbers as frames are triggered. Individual sequence sets can be used consecutively. After

one sequence set cycle is complete, another one will start automatically.

For more information, see Section 10.6.1 on [page 170](#).

- **Controlled sequence advance mode:** Sequence set advance is controlled by a source that can be selected. The available sources are automatic control by the camera (the "always active" setting), an input line or the "disabled" setting allowing sequence set advance only by software commands. The camera will cycle through the available sequence sets in ascending sequence set index numbers as frames are triggered. After one sequence set cycle is complete, another one will start automatically.
For more information, see Section 10.6.2 on [page 175](#).
- **Free selection sequence advance mode:** Sequence set advance is controlled by selecting sequence sets from the available sequence sets as desired. The selection is controlled by the states of the input lines.
For more information, see Section 10.6.3 on [page 194](#).

The regular cycling through the sequence sets according to the Auto or Controlled advance modes can be modified at any time during the cycling:

- a **restart** starts a new sequence set cycle before the previous cycle is completed. The restart can be controlled by the following:
 - the states of an input line (controlled sequence advance only)
 - a software command
- a non-cyclical **advance** allows skipping a sequence set and advancing to the next sequence set in line. The non-cyclical advance can be controlled by the following:
 - the states of an input line
 - a software command

Advance or restart controlled by an input line are also called "synchronous advance" and "synchronous restart" because the checking of the state of an input line and the subsequent loading of a sequence set is always linked to a frame start trigger signal.

Advance or restart controlled by a software command are also called "asynchronous advance" and "asynchronous restart" because they are not linked to a frame start trigger signal.



Synchronous advance and restart are part of the standard operation of the sequencer feature and can be used in most situations. Asynchronous advance and restart are not suitable for standard operation because of the associated delays:

The delay between sending a software command and it becoming effective will depend on the specific installation and the current load on the network.

Accordingly, the number of image acquisitions that may occur between sending the software command and it becoming effective can't be predicted.

Asynchronous advance and restart are therefore not suitable for real-time applications; they may, however, be useful for testing purposes.

We strongly recommend to **only** use synchronous advance and synchronous restart for real-time applications.

Using the Load Command

There is also the Sequence Set Load command that may be useful when working with the sequence sets for testing purposes. If you use the Sequence Set Selector parameter to select a sequence set and then you execute the Sequence Set Load command, the sequence parameter values in the active set will be replaced by the values stored in the selected sequence set.

This can be useful in two situations:

- If you simply want to see how the parameters currently stored in one of the sequence sets will affect camera's operation, you can load the parameters from that sequence set into the active parameter set and see what happens.
- If you want to prepare a new sequence set and you know that an existing set is already close to what you will need, you can load the existing sequence set into the active set, make some changes to the active set, and save the active set as a new sequence set.

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



Replacing the sequence parameter values in the active set via the Sequence Set Load command is associated with a delay between sending the software command and it becoming effective. The delay will depend on the specific installation and the current load on the network. Accordingly, the number of image acquisitions that may occur between sending the command and it becoming effective can't be predicted. The Sequence Set Load command is therefore not suitable for real-time applications; it may, however, be useful for testing purposes.

The following code snippet illustrates using the API to load the sequence parameter values from sequence set 0 into the active set:

```
// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue(0);
// Load the sequence parameter values from the sequence set into the
active set
Camera.SequenceSetLoad.Execute();
```

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

Use Case Diagrams Illustrating Sequencer Operation

The sections below explain the sequence advance modes in detail. Use case descriptions and diagrams are designed to illustrate how the sequence advance modes work in some common situations and with some common combinations of parameter settings.

These use cases do not represent every possible combination of the parameters associated with sequence advance mode operation. They are simply intended to aid you in developing an initial understanding of how the sequence advance modes work.

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



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

10.6.1 Auto Sequence Advance Mode

If the auto sequence advance mode is selected, the advance from one sequence set to the next occurs automatically as frame start triggers are received. The advance proceeds in ascending sequence set index numbers and subject to the Sequence Set Executions parameter value. It specifies how many times each sequence set is used consecutively. After the sequence set with the highest index number was used as many times as specified by the Sequence Set Executions parameter value, the sequence set cycle starts again with sequence set 0.

The Sequence Set Total Number parameter specifies the total number of different sequence sets that are available and included within a sequence set cycle. The maximum number is 64.

10.6.1.1 Operation

Operating the Sequencer

The following use case (see also Figure 51) illustrates the operation of the sequencer in auto sequence advance mode. As images are captured continuously, the camera advances automatically, without any input from the user, from one sequence set to the next in ascending sequence set index numbers. The advance is also subject to the Sequence Set Executions parameter settings. After one sequence set cycle is complete, another one starts.

In this use case, the Sequence Set Total Number parameter was set to 6. Accordingly, the available sequence set index numbers range from 0 to 5. The Sequence Set Executions parameter was set to 1 for sequence sets 0, 2, 3, and 4, to 2 for sequence set 5, and to 3 for sequence set 1. The frame start trigger is set for rising edge triggering.



The rising and the falling edge of an analog signal have different propagation delays. Therefore, choosing rising edge triggering may not be the best choice for you if you need to precisely control the trigger moment.

For more information about the temporal performance of I/O lines, see Section 5.6.2.3 on [page 36](#).

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled, the sequence set cycle starts: the parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

When a frame start trigger is received, sequence set 0 is used for the image acquisition.

- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 0 is set to 1, this sequence set is only used once and therefore the camera advances to the next sequence set: the parameter values of sequence set 1 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 1 is set to 3, this sequence set is used a second time: the parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 1 is set to 3, this sequence set is used a third time: the parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 1 is set to 3, this sequence set can't, after three uses, be used again in the current sequence set cycle. Therefore, the camera advances to the next sequence set: the parameter values of sequence set 2 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 2 is set to 1, this sequence set is only used once and therefore the camera advances to the next sequence set: the parameter values of sequence set 3 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 3 is set to 1, this sequence set is only used once and therefore the camera advances to the next sequence set: the parameter values of sequence set 4 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 4 is set to 1, this sequence set is only used once and therefore the camera advances to the

next sequence set: the parameter values of sequence set 5 are loaded into the active set and used for the image acquisition.

- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 5 is set to 2, this sequence set is used a second time: the parameter values of sequence set 5 are used for the image acquisition.

The camera has cycled once through the complete sequence set cycle.

- When the next frame start trigger is received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter for sequence set 5 is set to 2, this sequence set can't, after two uses, be used again in the current sequence set cycle. Therefore, the camera advances to the next sequence set: the parameter values of sequence set 0 are loaded into the active set and used for the image acquisition.

Another sequence set cycle has started.

If the sequencer feature is disabled while frame exposure and readout are in progress, the complete frame is transmitted and the cycling through sequence sets is terminated afterwards. The sequencer parameter values in the active set return to the values before the sequencer feature was enabled.

Use Case: Operation in auto sequence advance mode:
Automatic cycling through the sequence set cycles without input from the user. Enabling and disabling of the sequencer feature.

Settings: Sequence Set Total Number = 6
Sequence Set Executions = 1 for sequence sets 0, 2, 3, and 4
Sequence Set Executions = 2 for sequence set 5
Sequence Set Executions = 3 for sequence set 1
Frame start trigger is set for rising edge triggering

- ▼ = camera selects a sequence set as the current sequence set (x) = count of frame start trigger
- 0 = current sequence set that is used for the image acquisition (the sequence set index number is indicated)
- █ = frame exposure and readout
- ⋮ = frame transmission

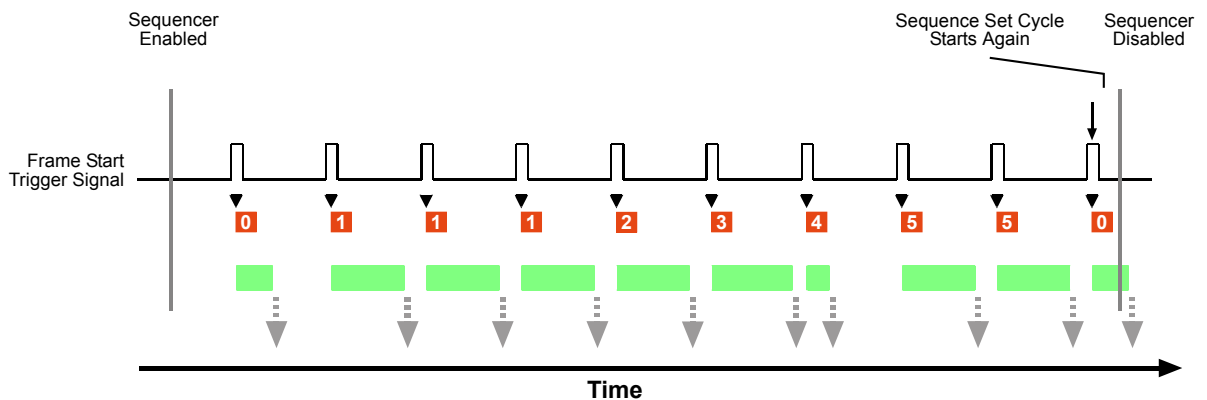


Fig. 51: Sequencer in Auto Sequence Advance Mode

Operating the Sequencer Using Basler pylon

You can use the pylon API to set the parameters for operating the sequencer in Auto sequence advance mode from within your application software.

The following code snippet illustrates enabling the sequencer and disabling the sequencer. The example assumes that sequence sets were previously configured and are currently available in the camera's memory.

```
// Enable the sequencer feature
Camera.SequenceEnable.SetValue(true);

// Disable the sequencer feature
Camera.SequenceEnable.SetValue(false);
```

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

10.6.1.2 Configuration

Configuring Sequence Sets and Advance Control

To populate the sequence sets and set the related parameters:

1. Make sure that the sequencer feature is disabled.
2. Set the Sequence Advance Mode parameter to Auto.
3. Set the Sequence Set Total Number parameter. The maximum number is 64.
4. Select a sequence set index number by setting the Sequence Set Index parameter. The numbers available range from 0 to 63.
When configuring sequence sets make sure to always use a continuous series of index numbers starting with index number 0 and ending with the Sequence Set Total Number parameter value minus one. For example, specifying a series of sequence sets with index numbers 5, 6, and 8 only is not allowed. If you did this, the missing sequence sets in between would, within the scope of the sequence set total number, be populated by default parameter values.
5. Set up your first acquisition scenario (i.e., lighting, object positioning, etc.).
6. Adjust the camera parameters to get the best image quality with this scenario (you are adjusting all parameters in the active set).
7. Set the Sequence Set Executions parameter. The numbers available range from 1 to 256.
8. Execute the Sequence Set Store command to copy the sequence parameter values currently in the active set into the selected sequence set. Any already existing parameter values in the sequence set will be overwritten.
9. Repeat the above steps starting from step 4 for the other sequence sets.

Configuring Sequence Sets and Advance Control Using Basler pylon

You can use the pylon API to set the parameters for configuring sequence sets from within your application software.

The following code snippet contains sample settings. It illustrates using the API to set the auto sequence advance mode, set the total number of sequence sets to 2, set the numbers of consecutive sequence set executions and populate sequence sets 0 and 1 by storing the sequence parameter values from the active set in the sequence sets:

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue(false);

// Set the Auto sequence advance mode
Camera.SequenceAdvanceMode.SetValue(SequenceAdvanceMode_Auto);

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

// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue(0);

// Set up the first acquisition scenario (lighting, object position,
etc.) and
//adjust the camera parameters for the best image quality.

// Set the number of sequence set uses
Camera.SequenceSetExecutions.SetValue(1);

// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();

// Select sequence set with index number 1
Camera.SequenceSetIndex.SetValue(1);

// Set up the second acquisition scenario (lighting, object position,
etc.) and
// adjust the camera parameters for the best image quality.

// Set the number of sequence set uses
Camera.SequenceSetExecutions.SetValue(4);
```

```
// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();
```

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

10.6.2 Controlled Sequence Advance Mode

If the controlled sequence advance mode is selected, the advance from one sequence set to the next proceeds in ascending sequence set index numbers according to the selected sequence control source:

- **Always Active**
The advance from one sequence set to the next proceeds automatically as frame start triggers are received.
- **Line 1, CC1, CC2, or CC3**
The advance from one sequence set to the next proceeds according to the states of the selected input line.
- **Disabled**
The advance from one sequence set to the next is only controlled by AsyncAdvance software commands.

The Sequence Set Total Number parameter specifies the total number of different sequence sets that are available and included within a sequence set cycle. The maximum number is 64.

10.6.2.1 Operation with the "Always Active" Sequence Control Source

Operating the Sequencer

If the Always Active sequence control source is selected, the advance from one sequence set to the next proceeds automatically in ascending sequence set index numbers as frame start triggers are received.

The following use case (see also Figure 52) illustrates the operation of the sequencer in controlled sequence advance mode with Always Active selected as the sequence control source. As images are captured continuously, the camera advances automatically without any input from the user from one sequence set to the next in ascending sequence set index numbers. After one sequence set cycle is complete, another one starts.



This way of operating the sequencer feature is similar to operating it in auto sequence advance mode when each sequence set is used only once per sequence set cycle.

Here, however, the first sequence set used for image acquisition after the sequencer feature has been enabled is sequence set 1 as opposed to sequence set 0 in auto sequence advance mode.

In this use case, the Sequence Set Total Number parameter is set to 6. Accordingly, the available sequence set index numbers range from 0 to 5. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled, the sequence set cycle starts: the parameter values of the sequence set with sequence set index number 0 are loaded into the active set, thus modifying the active set.
When a frame start trigger is received, the camera automatically advances to the next sequence set: the parameter values of sequence set 1 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera advances to the next sequence set: the parameter values of sequence set 2 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera advances to the next sequence set: the parameter values of sequence set 3 are loaded into the active set and used for the image acquisition.
- The sequencer carries on cycling through the sequence sets. Note that the camera has completed a sequence set cycle when sequence set 5 has been used. With the next frame start trigger, a new sequence set cycle starts where sequence set 0 is used.
- After the sequencer feature is disabled, the cycling through sequence sets is terminated. The sequencer parameter values in the active set return to the values before the sequencer feature was enabled.

Use Case: Operation in controlled sequence advance mode with Always Active as the sequence control source:
Automatic cycling through the sequence set cycles without input from the user. Enabling and disabling of the sequencer feature.

Settings: Sequence Set Total Number = 6
Frame start trigger is set for rising edge triggering

▼ = camera selects a sequence set as the current sequence set

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

■ = frame exposure and readout

⋮ = frame transmission

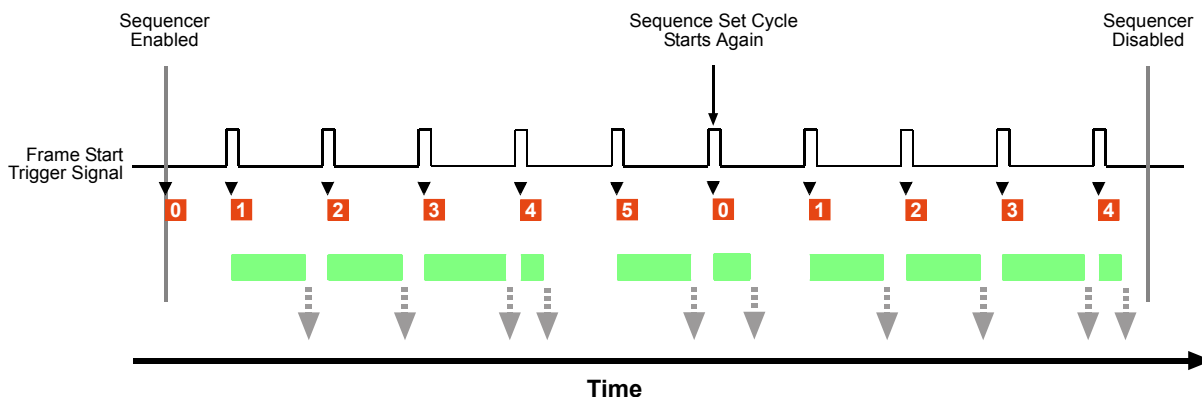


Fig. 52: Sequencer in Controlled Sequence Advance Mode with Always Active as the Sequence Control Source

Synchronous Restart

You can restart the sequence cycle with input lines Line 1, CC1, CC2, or CC3 as the source for controlling sequence cycle restart.

In the following use case (see also Figure 53), the same settings were chosen as in the previous use case: The Sequence Set Total Number parameter was set to 6. Accordingly, the available sequence set index numbers range from 0 to 5. The frame start trigger is set for rising edge triggering. In addition, Line 1 was selected as the source for controlling restart. Line 1 is not set for invert.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:


- When the sequencer feature becomes enabled, the sequence set cycle starts: the parameter values of the sequence set with sequence set index number 0 are loaded into the active set, thus modifying the active set.

When a frame start trigger is received, the camera automatically advances to the next sequence set: the parameter values of sequence set 1 are loaded into the active set and used for the image acquisition.


- When the next frame start trigger is received, the camera advances to the next sequence set: the parameter values of sequence set 2 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, the camera advances to the next sequence set: the parameter values of sequence set 3 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, Line 1 is found to be high. Accordingly, another sequence set cycle is started and the parameter values of sequence set 0 are loaded into the active set and used for the image acquisition.

Note that the synchronous restart has priority here over the automatic sequence set advance that results from the Always Active sequence control source. Without the priority rule, sequence set 4 would be used.

- Note that the state of Line 1 goes high well ahead of the frame start trigger.

	<p>To ensure reliable synchronous sequence set restart, allow at least one microsecond to pass between the time when the signal of the sequencer trigger source has reached the high state and the rise of the frame start trigger signal.</p> <p>Also, maintain the state of the input line for at least one microsecond after the frame start trigger signal has risen.</p>
---	---






Note also that the camera briefly exits the "waiting for frame start trigger" status while the input line changes its state. This happens when input line 1 changes its state before the fourth frame start trigger is received (see also Figure 53).

	<p>Make sure not to send a frame start trigger while the input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.</p> <p>Make sure to only send a frame start trigger when the camera is in the "waiting for frame start trigger" status.</p> <p>For more information about the "waiting for frame start trigger" status, see the "Acquisition Monitoring Tools" section on page 98.</p>
---	---

- When the next frame start trigger is received, the camera advances to the next sequence set: the parameter values of sequence set 1 are loaded into the active set and used for the image acquisition.
- When the next frame start trigger is received, Line 1 is found to be high. Accordingly, another sequence set cycle is started and the parameter values of sequence set 0 are loaded into the active set and used for the image acquisition. As explained above, synchronous restart has priority here over the automatic sequence set advance.
- When the next frame start triggers are received, the camera continues advancing from one sequence set to the next in ascending sequence set index numbers and using them for image acquisition unless another synchronous restart signal is received.

Use Case: Operation in controlled sequence advance mode with Always Active as the sequence control source:
Automatic cycling through the sequence set cycles with two synchronous restarts controlled by Line 1.

Setting: Sequence Set Total Number = 6
Line 1 (not set for invert) is selected as the source for controlling restart
Frame start trigger is set for rising edge triggering

-  = camera is waiting for a frame start trigger
-  = camera selects a sequence set as the current sequence set
-  = current sequence set that is used for the image acquisition (the sequence set index number is indicated)
-  = frame exposure and readout
-  = frame transmission

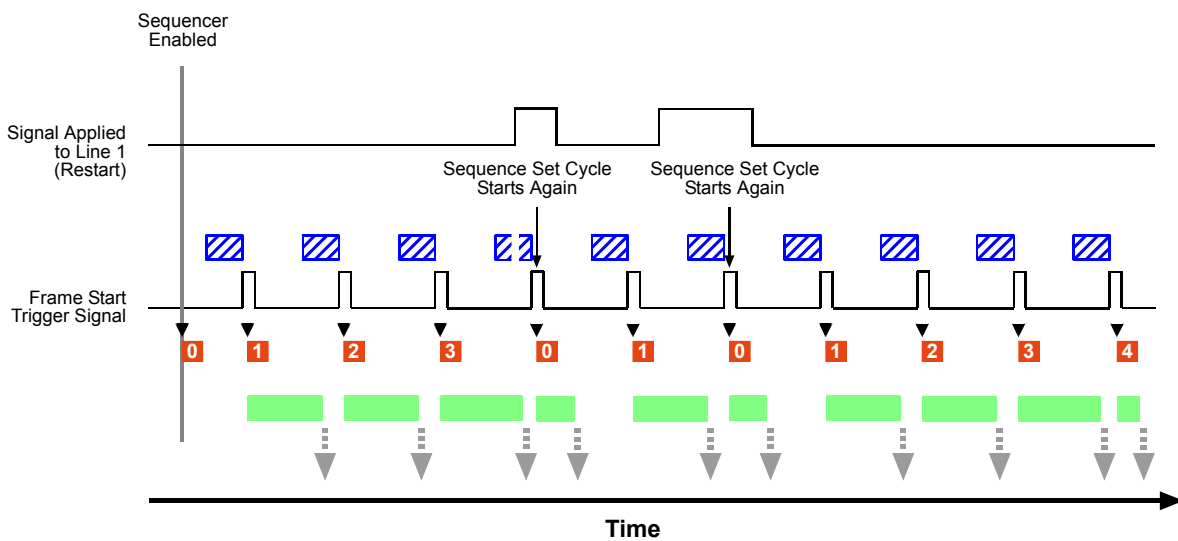


Fig. 53: Sequencer in Controlled Sequence Advance Mode with Always Active as the Sequence Control Source and Synchronous Restart Controlled by Line 1

10.6.2.2 Operation with an Input Line as Sequence Control Source

Operating the Sequencer

If Line 1 is selected as sequence control source, the advance from one sequence set to the next is controlled by the state of Line 1. The advance proceeds in ascending sequence set index numbers as frame start triggers are received.



This section assumes that Line 1 is selected as the sequence control source. All explanations, however, apply equally when the CC1, CC2, or CC3 lines are used as the sequence control source.

The following use case (see also Figure 54) illustrates the operation of the sequencer in controlled sequence advance mode with Line 1 selected as the sequence control source. The camera advances from one sequence set to the next in ascending sequence set index numbers. After one sequence set cycle is complete, another one starts. The sequence set advance is controlled by the state of Line 1. Line 1 is not set for invert.

In this use case, the Sequence Set Total Number parameter was set to 6. Accordingly, the available sequence set index numbers range from 0 to 5. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled, the sequence set cycle starts: the parameter values of the sequence set with sequence set index number 0 are loaded into the active set, thus modifying the active set.
When a frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be low (the line status equals zero) and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 0 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be high (the line status equals one) and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

Note that the state of Line 1 goes high well ahead of the frame start trigger.



To ensure reliable selection of a sequence set, allow at least one microsecond to pass between the time when the signal of the sequencer trigger source has reached the high state and the rise of the frame start trigger signal.

Also, maintain the state of the input line for at least one microsecond after the frame start trigger signal has risen.

Note also that the camera briefly exits the "waiting for frame start trigger" status while an input line changes its state. This happens when input line 1 changes its state before the second frame start trigger is received (see also Figure 54).



Make sure not to send a frame start trigger while the input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.

Make sure to only send a frame start trigger when the camera is in the "waiting for frame start trigger" status.

For more information about the "waiting for frame start trigger" status, see the "Acquisition Monitoring Tools" section on [page 98](#).

- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 2 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 3 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 4 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 5 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 5 are used for the image acquisition.

The camera has cycled once through the complete sequence set cycle.






- When the next frame start trigger is received, the camera checks the state of Line 1. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 0 are used for the image acquisition.

Another sequence set cycle has started.

- After frame exposure and readout are completed, the sequencer feature is disabled. The cycling through sequence sets is terminated. The sequencer parameter values in the active set return to the values before the sequencer feature was enabled.

Use Case: Operation in controlled sequence advance mode with Line 1 as the sequence control source:
Cycling through the sequence set cycles according to the state of input line 1 (not set for invert). Enabling and disabling of the sequencer feature.

Setting: Sequence Set Total Number = 6
Frame start trigger is set to rising edge triggering

-  = camera is waiting for a frame start trigger
-  = camera selects a sequence set as the current sequence set
-  = current sequence set that is used for the image acquisition (the sequence set index number is indicated)
-  = frame exposure and readout
-  = frame transmission

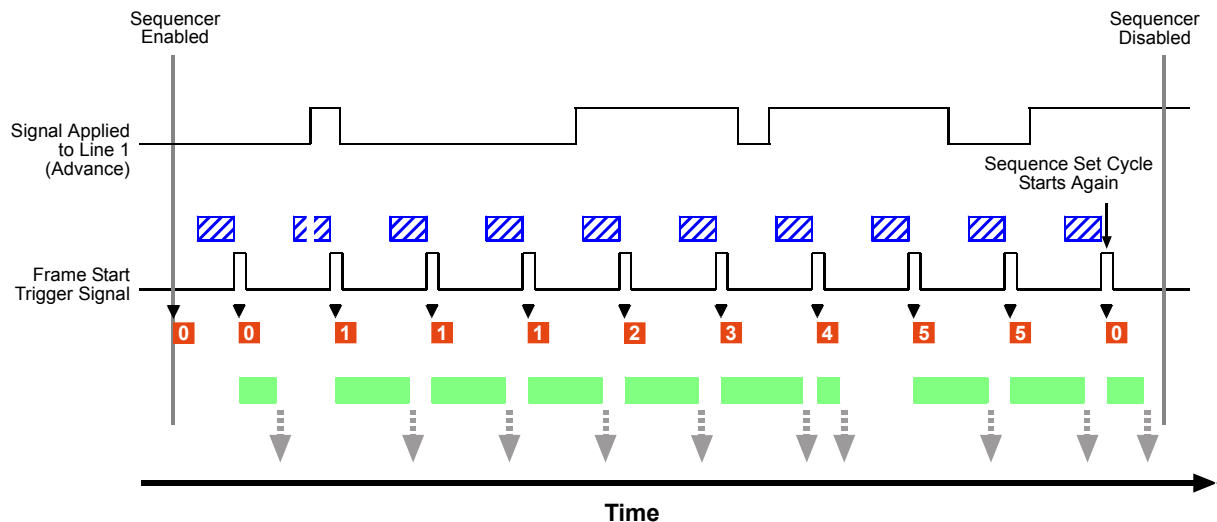


Fig. 54: Sequencer in Controlled Sequence Advance Mode with Line 1 as the Sequence Control Source

Synchronous Restart

You can restart the sequence cycle by selecting an input line that is not used for sequence advance control as the source for controlling sequence cycle restart.

In the following use case (see also Figure 55), the same settings were made as in the previous use case: the Sequence Set Total Number parameter was set to 6. Accordingly, the sequence set index numbers range from 0 to 5. The frame start trigger is set for rising edge triggering. Line 1 is selected as the sequence control source for controlling sequence set advance. In addition, CC1 is selected as the source for controlling sequence cycle restart. Both input lines are not set for invert.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:


- When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

When a frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 0 are used for the image acquisition.


- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 2 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be high and therefore the sequence cycle is restarted. Line 1 is found to be low but this has no significance: Synchronous restart has priority over the sequence set advance control. The parameter values of sequence set 0 are used for the image acquisition.

Another sequence set cycle has started.

Note that the state of CC1 goes high well ahead of the frame start trigger.

	<p>To ensure reliable selection of a sequence set, allow at least one microsecond to pass between the time when the signal of the sequence trigger source has reached the high state and the rise of the frame start trigger signal.</p> <p>Also, maintain the states of the input lines at least for one microsecond after the frame start trigger signal has risen.</p>
---	---






Note also that the camera briefly exits the "waiting for frame start trigger" status while an input line changes its state. This happens, for example, when CC1 changes its state before the fourth frame start trigger is received (see also Figure 55).

	<p>Make sure not to send a frame start trigger while an input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.</p> <p>Make sure to only send a frame start trigger when the camera is in the "waiting for frame start trigger" status.</p> <p>For more information about the "waiting for frame start trigger" status, see the "Acquisition Monitoring Tools" section on page 98.</p>
---	--

- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 2 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 3 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be high and therefore the sequence cycle is restarted. Line 1 is found to be high but this has no significance: Synchronous restart has priority over the sequence set advance control. The parameter values of sequence set 0 are used for the image acquisition. Another sequence set cycle has started.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. CC1 is found to be low and therefore the sequence cycle is not restarted. Line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

Use Case: Operation in controlled sequence advance mode with Line 1 as the sequence control source:
 Cycling through the sequence set cycles according to the states of Line 1 (not set for invert) with two synchronous restarts controlled by CC1.

Setting: Sequence Set Total Number = 6
 CC1 (not set for invert) is selected as the source for controlling restart
 Frame start trigger is set for rising edge triggering

-  = camera is waiting for a frame start trigger
-  = camera selects a sequence set as the current sequence set
-  = current sequence set that is used for the image acquisition (the sequence set index number is indicated)
-  = frame exposure and readout
-  = frame transmission

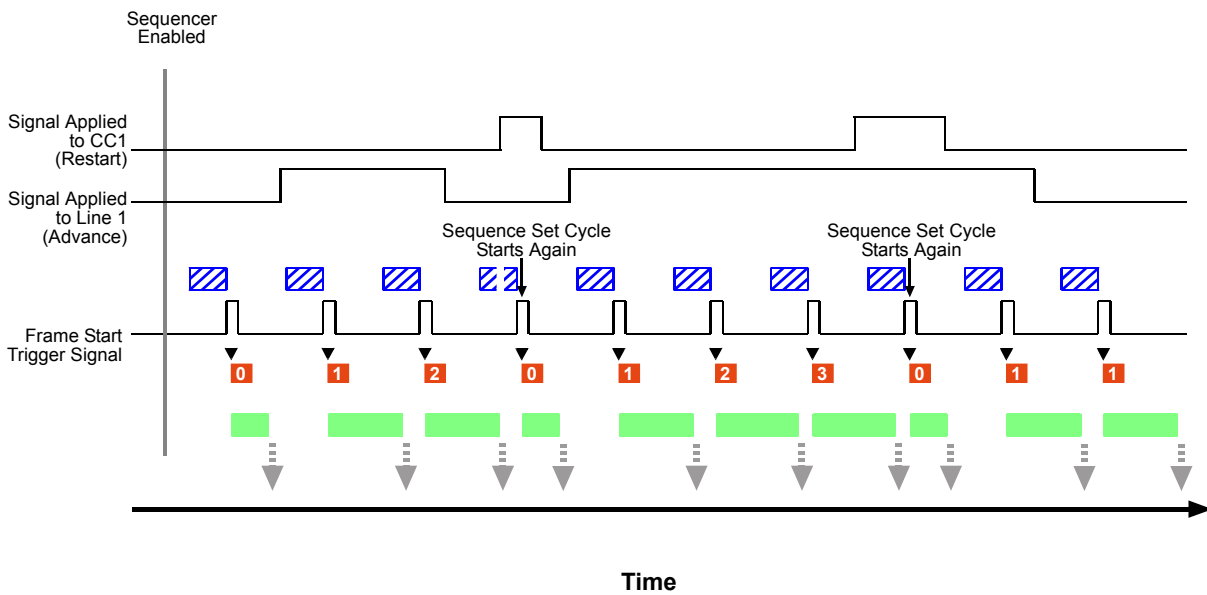


Fig. 55: Sequencer in Controlled Sequence Advance Mode with Line 1 as the Sequence Control Source and Synchronous Restart Controlled by CC1

10.6.2.3 Operation with the "Disabled" Sequence Control Source

Operating the Sequencer

If the Disabled sequence control source is selected, the advance from one sequence set to the next proceeds in ascending sequence set index numbers and is only possible by asynchronous advance.

Similarly, sequence set restart is only possible by asynchronous restart.



The delay between sending an AsyncAdvance or an AsyncRestart software command and it becoming effective will depend on the specific installation and the current load on the network. Accordingly, the number of image acquisitions that may occur between sending the software command and it becoming effective can't be predicted. Using the sequencer feature with Disabled sequence control source is therefore not suitable for real-time applications; it may, however, be useful for testing purposes.

Basler strongly discourages using the sequencer feature with Disabled sequence control source for real-time applications.

The following use case (see also Figure 56) illustrates the operation of the sequencer in controlled sequence advance mode with Disabled selected as the sequence control source. Sequence set advance proceeds in ascending sequence set index numbers subject to asynchronous advance commands. After one sequence set cycle is complete, another one starts. Sequence set cycle restarts are subject to asynchronous restart commands.

In this use case, the Sequence Set Total Number parameter was set to 6. Accordingly, the available sequence set index numbers range from 0 to 5. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled, the sequence set cycle starts: the parameter values of the sequence set with sequence set index number 0 are loaded into the active set, thus modifying the active set.
When a frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.
- An AsyncAdvance command is sent. After some delay, the parameter values of the next sequence set will be loaded into the active set. It is assumed here that the delay between sending the AsyncRestart command and it becoming effective will allow the acquisition of two more images.
- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.
The AsyncAdvance command has not yet become effective because of the assumed associated delay.

- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.

The AsyncAdvance command has not yet become effective because of the assumed associated delay.

- When the AsyncAdvance command becomes effective, the camera happens to be in the "waiting for frame start trigger" status. The parameter values of the next sequence set, i.e. of sequence set 1, are loaded into the active set. Note that the camera briefly exits the "waiting for frame start trigger" status while the parameter values of sequence set 1 are loaded into the active set (see also Figure 56).



Make sure not to send a frame start trigger while the parameter values of a sequence set are being loaded into the active set. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.

Make sure to only send a frame start trigger when the camera is in the "waiting for frame start trigger" status.

For more information about the "waiting for frame start trigger" status, see the "Acquisition Monitoring Tools" section on [page 98](#).

- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.
- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.
- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.
- An AsyncRestart command is sent. After some delay, the parameter values of sequence set 0 will be loaded into the active set. It is assumed here that the delay between sending the AsyncRestart command and it becoming effective will allow the acquisition of two more images.
- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.
The AsyncRestart command has not yet become effective because of the assumed associated delay.
- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.
The AsyncRestart command has not yet become effective because of the assumed associated delay.
- When the AsyncRestart command becomes effective, the camera happens to be in the "waiting for frame start trigger" status. The parameter values of sequence set 0 are loaded into the active set. Note that the camera briefly exits the "waiting for frame start trigger" status while the parameter values of sequence set 1 are loaded into the active set (see also Figure 56).



Make sure not to send a frame start trigger while the parameter values of a sequence set are being loaded into the active set. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.










Make sure to only send a frame start trigger when the camera is in the "waiting for frame start trigger" status.

For more information about the "waiting for frame start trigger" status, see the "Acquisition Monitoring Tools" section on [page 98](#).

- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.
Another sequence set cycle has started
- When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.
- While frame exposure and readout are in progress, the sequencer feature is disabled. The complete frame is transmitted and the cycling through sequence sets is terminated. The sequencer parameter values in the active set return to the values before the sequencer feature was enabled.

Use Case: Operation in controlled sequence advance mode with Disabled sequence control source:
 Cycling through the sequence set cycles only due to one asynchronous advance and one asynchronous restart. Enabling and disabling of the sequencer feature.

Setting: Sequence Set Total Number = 6
 Frame start trigger is set for rising edge triggering

-  = asynchronous advance command sent (AsyncAdvance command)
-  = delay between sending the advance command and it becoming effective
-  = asynchronous restart command sent (AsyncRestart command)
-  = delay between sending the restart command and it becoming effective
-  = camera is waiting for a frame start trigger
-  = camera selects a sequence set as the current sequence set
-  = current sequence set that is used for the image acquisition (the sequence set index number is indicated)
-  = frame exposure and readout
-  = frame transmission

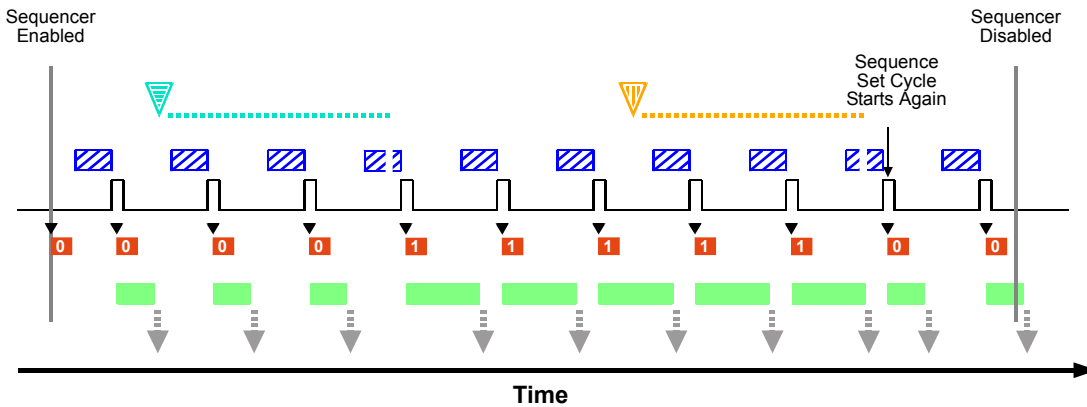


Fig. 56: Sequencer in Controlled Sequence Advance Mode with Disabled as the Sequence Control Source and Asynchronous Advance and Restart

Operating the Sequencer Using Basler pylon

You can use the pylon API to set the parameters for operating the sequencer in Controlled sequence advance mode from within your application software.

The following code snippet illustrates enabling and disabling the sequencer. The example assumes that sequence sets were previously configured and are currently available in the camera's memory.

```
// Enable the sequencer feature
Camera.SequenceEnable.SetValue(true);

// Disable the sequencer feature
Camera.SequenceEnable.SetValue(false);
```

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

10.6.2.4 Configuration

Configuring Sequence Sets and Advance Control

To populate sequence sets and set the sources for sequence set advance and sequence cycle restart:

1. Make sure that the sequencer feature is disabled.
2. Set the Sequence Advance mode to Controlled.
3. Set the Sequence Set Total Number parameter.
The maximum number is 64.
4. Set the Sequence Control Selector parameter to Advance to configure sequence set advance.
Set the Sequence Control Source parameter to specify the source that will control synchronous sequence set advance. Note that the same source will apply to all sequence sets available at the same time in the camera.
The following sources are available:
 - Always Active
 - Line 1
 - CC1, CC2, and CC3
 - Disabled
5. Set the Sequence Control Selector parameter to Restart to configure sequence set cycle restart.
6. Set the Sequence Control Source parameter to specify the source for restart.



Never choose the same source for sequence set advance and sequence set cycle restart, with one exception:

If you want to only use asynchronous advance and restart, choose Disabled as the source for advance and restart.

The following sources are available:

- Line 1
- CC1, CC2, and CC3
- Disabled

7. Select a sequence set index number by setting the Sequence Set Index parameter. The numbers available range from 0 to 63.

When configuring sequence sets make sure to always use a continuous series of index numbers starting with index number 0 and ending with the Sequence Set Total Number parameter value minus one. For example, specifying a series of sequence sets with index numbers 5, 6, and 8 only is not allowed. If you did this, the missing sequence sets in between would, within the scope of the sequence set total number, be populated by default parameter values.

8. Set up your first acquisition scenario (i.e., lighting, object positioning, etc.).
9. Adjust the camera parameters to get the best image quality with this scenario (you are adjusting the parameters in the active set).
10. Execute the Sequence Set Store command to copy the sequence parameter values currently in the active set into the selected sequence set. Any existing parameter values in the sequence set will be overwritten.
11. Repeat the above steps for the other sequence sets.

For information about setting the input line for invert, see Section 6.1.5 on [page 47](#).

Configuring Sequence Sets and Advance Control Using Basler pylon

You can use the pylon API to set the parameters for configuring sequence sets from within your application software.

The following code snippet contains sample settings. It illustrates using the API to set the controlled sequence advance mode. In the example, Line 1 is set as the sequence control source for synchronous sequence set advance, Disabled is set as the sequence control source to allow asynchronous sequence cycle reset, the total number of sequence sets is set to 2, sequence sets 0 and 1 are populated by storing the sequence parameter values from the active set in the sequence sets, and the sequencer feature is enabled:

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue(false);

// Set the Controlled sequence advance mode and set line 1 as the
sequence
// control source for synchronous sequence set advance
Camera.SequenceAdvanceMode.SetValue(SequenceAdvanceMode_Controlled);
Camera.SequenceControlSelector.SetValue(SequenceControlSelector_Advan
ce);
Camera.SequenceControlSource.SetValue(SequenceControlSource_Line1);

// Set Disabled as the source because synchronous sequence set cycle
restart
// will not be used
Camera.SequenceControlSelector.SetValue(SequenceControlSelector_Resta
rt);
Camera.SequenceControlSource.SetValue(SequenceControlSource_Disabled)
;

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

// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue(0);

// Set up the first acquisition scenario (lighting, object position,
etc.) and
// adjust the camera parameters for the best image quality.

// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();
```

```
// Select sequence set with index number 1
Camera.SequenceSetIndex.SetValue(1);

// Set up the second acquisition scenario (lighting, object position,
etc.) and
// adjust the camera parameters for the best image quality.

// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();

// Enable the sequencer feature
Camera.SequenceEnable.SetValue(true);
```

The following code snippet illustrates using the API to load the sequence parameter values from sequence set 0 into the active set:

```
// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue(0);
// Load the sequence parameter values from the sequence set into the
active set
Camera.SequenceSetLoad.Execute();
```

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

10.6.3 Free Selection Sequence Advance Mode

If the free selection sequence advance mode is selected, the advance from one sequence set to the next as frame start triggers are received does not adhere to a specific preset sequence. The sequence sets can be selected as desired using the states of input lines. The states of the input lines set the sequence set addresses. These correspond to the sequence set index numbers and the related sequence set is selected accordingly. For details about selecting sequence sets via the sequence set address, see the "Selecting Sequence Sets" section on [page 197](#).

The states of all input lines are checked if more than two sequence sets are available. The state of one input line is checked if only two sequence sets are available.

The Sequence Set Total Number parameter specifies the total number of different sequence sets that are available. The maximum number is 4.

10.6.3.1 Operation

Operating the Sequencer

The following use case (see also Figure 57) illustrates the operation of the sequencer in free selection sequence advance mode.

In this use case, the Sequence Set Total Number parameter was set to 4. Accordingly, the sequence set index numbers range from 0 to 3. Line 1 sets bit 0 of the sequence set address. CC1 sets bit 1 of the sequence set address. Both input lines are not set for invert. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled and a frame start trigger is received, the camera checks the states of Line 1 and CC1. Line 1 is found to be high and CC1 is found to be low. This corresponds to the address of sequence set 1. Accordingly, sequence set 1 is selected. Its parameter values are loaded into the active set and are used for the image acquisition.

Note that the state of Line 1 goes high well ahead of the frame start trigger.



To ensure reliable selection of a sequence set, allow at least one microsecond to pass between the time when the signal of the sequencer trigger sources has reached the high state and the rise of the frame start trigger signal.

Also, maintain the states of the input lines Line 1 and CC 1 for at least one microsecond after the frame start trigger signal has risen.

Note also that the camera briefly exits the "waiting for frame start trigger" status while an input line changes its state. This happens, for example, when input line 1 changes its state before the first frame start trigger is received (see also Figure 57).



Make sure not to send a frame start trigger while an input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.






Make sure to only send a frame start trigger when the camera is in the "waiting for frame start trigger" status.

For more information about the "waiting for frame trigger" status, see the "Acquisition Monitoring Tools" section on [page 98](#).

- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. Because the states have not changed, the parameter values of sequence set 1 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. The states of both input lines are found to be low. This corresponds to the address of sequence set 0. Accordingly, sequence set 0 is selected. The parameter values of sequence set 0 are used for the image acquisition.
- When the next frame start trigger is received, the camera checks the states of Line 1 and CC1. Line 1 is found to be low and CC1 is found to be high. This corresponds to the address of sequence set 2. Accordingly, sequence set 2 is selected. The parameter values of sequence set 2 are used for the image acquisition.
- The sequence sets for the next five frame start triggers are selected and used according to the scheme that applied to the preceding frame start triggers.
- While frame exposure and readout for the fifth frame start trigger are in progress, the sequencer feature is disabled. The complete frame is transmitted. The sequencer parameter values in the active set return to the values before the sequencer feature was enabled.

Use Case: Operation in free selection sequence advance mode.
Sequence sets are selected as desired. The selection is controlled by the states of the input lines.

Settings: Sequence Set Total Number = 4
Line 1 (not set for invert) sets bit 0 of the sequence set address.
CC1 (not set for invert) sets bit 1 of the sequence set address.
Frame start trigger is set for rising edge triggering

-  = camera is waiting for a frame start trigger
-  = camera selects a sequence set as the current sequence set
-  = current sequence set that is used for the image acquisition (the sequence set index number is indicated)
-  = frame exposure and readout
-  = frame transmission

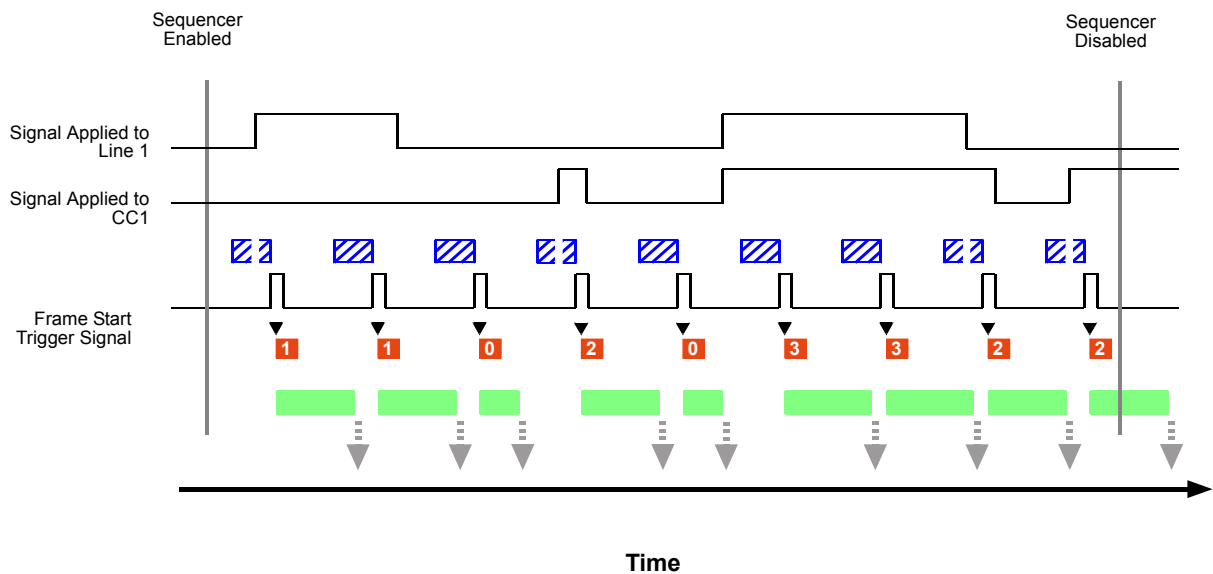


Fig. 57: Sequencer in Free Selection Mode

Operating the Sequencer Using Basler pylon

You can use the pylon API to set the parameters for operating the sequencer in Free Selection sequence advance mode from within your application software.

The following code snippet illustrates enabling and disabling the sequencer. The example assumes that sequence sets were previously configured and are currently available in the camera's memory.

```
// Enable the sequencer feature
Camera.SequenceEnable.SetValue(true);
```

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue(false);
```

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

Selecting Sequence Sets

Each sequence set is identified by a sequence set index number, starting from zero. The states of the input lines determine the sequence sets by setting the (big-endian) sequence set addresses. The addresses are simply the binary expressions of the sequence set index numbers. A maximum of four sequence sets can be used:

Sequence Set Address		Related Sequence Set
Bit 1	Bit 0	
0	0	Sequence Set 0
0	1	Sequence Set 1
1	0	Sequence Set 2
1	1	Sequence Set 3

Table 13: Sequence Set Addresses and Related Sequence Sets

The Sequence Set Total Number parameter specifies the total number of sequence sets that will be available. The parameter also specifies the length of the settable sequence set address.

- If the Sequence Set Total Number parameter is set to 2:
Bit 0 of the binary sequence set index number (see Table 13) can be set. When the bit is set to 0, sequence set 0 will be selected and when the bit is set to 1, sequence set 1 will be selected (see Table 13).

You can use the states of Line 1, CC1, CC2, or CC3 to set bit 0. You only need to set one bit in this scenario.

- If the input line is not set for invert, the high state of the input line will set bit 0 to 1 and the low state will set bit 0 to 0.
- If the input line is set for invert, the low state of the input line will set bit 0 to 1 and the high state will set bit 0 to 0.
- If the Sequence Set Total Number parameter is set to a number higher than 2:
Bits 1 and 0 of the binary sequence set index number (see Table 13) can be set. If for example bit 0 is set to 0 and bit 1 is set to 1, sequence set 2 will be selected (see Table 13).

You can use the states of Line 1 to set bit 0 and the states of a different input line to set bit 1 or vice versa.

- If the input lines are not set for invert, the high states of the input lines will set the bits to 1 and the low states will set the bits to 0.
- If the input lines are set for invert, the low states of the input lines will set the bits to 1 and the high states will set the bits to 0.

For information about setting an input line for invert, see Section 6.1.5 on [page 47](#).

10.6.3.2 Configuration

Configuring Sequence Sets and Advance Control

To populate sequence sets and set the source for sequence set advance:

1. Make sure that the sequencer feature is disabled.
2. Set the Sequence Advance Mode parameter to Free Selection.
3. Set the Sequence Set Total Number parameter.
The maximum number is 4.
4. Select the sequence set address bits and set the input lines that will act as the control sources:
 - If the Set Total Number parameter was set to 2:
 - a. Bit 0 will be selected by default as the sequence set address bit. Set Line 1 or CC1 as the control source for setting bit 0.
 - If the Set Total Number parameter was set to a number higher than 2:
 - a. Select bit 0 of the sequence set address bit.
 - b. Set Line 1 or CC1 as the control source for setting bit 0.
 - c. Select bit 1 of the sequence set address bit.
 - d. Set the input line as the control source for setting bit 1. Choose an input line not used for setting bit 0.
5. Use the Sequence Set Index parameter to select a sequence set index number for the sequence set currently being populated.
The numbers available are 0 to 3.
6. Set up your first acquisition scenario (i.e., lighting, object positioning, etc.).
7. Adjust the camera parameters to get the best image quality with this scenario (you are adjusting the parameters in the active set).
8. Execute the Sequence Set Store command to copy the sequence parameter values currently in the active set into the selected sequence set. (Any existing parameter values in the sequence set will be overwritten.)
9. Repeat the above steps for the other sequence sets, starting from step 5.

Configuring Sequence Sets and Advance Control Using Basler pylon

You can use the pylon API to configure the parameters of the sequence sets from within your application software and to configure how they are selected when images are acquired.

The following code snippet contains sample settings. It illustrates using the API to set the free selection sequence advance mode with line 1 as the control source for bit 0 and CC1 as the control source for bit 1 of the sequence set address, set the total number of sequence sets to 3, and populate sequence sets 0 to 2 by storing the sequence parameter values from the active set in the sequence sets:

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue(false);
```

```
// Set the Free Selection sequence advance mode
Camera.SequenceAdvanceMode.SetValue(SequenceAdvanceMode_FreeSelection
);

// Set the total number of sequence sets
Camera.SequenceSetTotalNumber.SetValue(3);

// Set line 1 as the control source for setting sequence set address
bit 0
Camera.SequenceAddressBitSelector.SetValue(SequenceAddressBitSelector
_Bit0);
Camera.SequenceAddressBitSource.SetValue(SequenceAddressBitSource_Lin
e1);

// Set CC1 as the control source for setting sequence set address bit 1
Camera.SequenceAddressBitSelector.SetValue(SequenceAddressBitSelector
_Bit1);
Camera.SequenceAddressBitSource.SetValue(SequenceAddressBitSource_CC1
);

// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue(0);

// Set up the first acquisition scenario (lighting, object position,
etc.) and
// adjust the camera parameters for the best image quality.

// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();

// Select sequence set with index number 1
Camera.SequenceSetIndex.SetValue(1);

// Set up the second acquisition scenario (lighting, object position,
etc.) and
// adjust the camera parameters for the best image quality.

// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();
```



```
// Select sequence set with index number 2
Camera.SequenceSetIndex.SetValue(2);

// Set up the third acquisition scenario (lighting, object position,
etc.) and
// adjust the camera parameters for the best image quality.

// Store the sequence parameter values from the active set in the
selected sequence set
Camera.SequenceSetStore.Execute();
```

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

10.7 Binning



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 reported out of the camera as a single pixel. Figure 58 illustrates vertical binning.

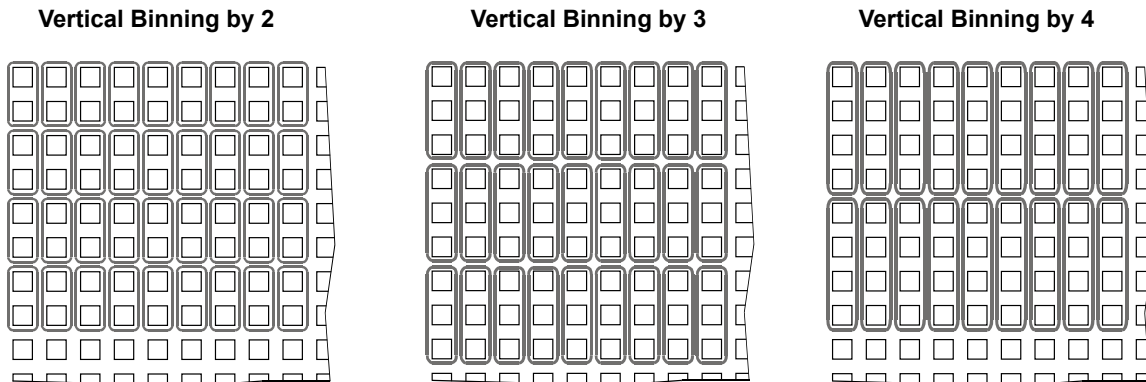


Fig. 58: Vertical Binning

With horizontal binning, adjacent pixels from 2 columns, 3 columns, or a maximum of 4 columns are summed and reported out of the camera as a single pixel. Figure 59 illustrates horizontal binning.

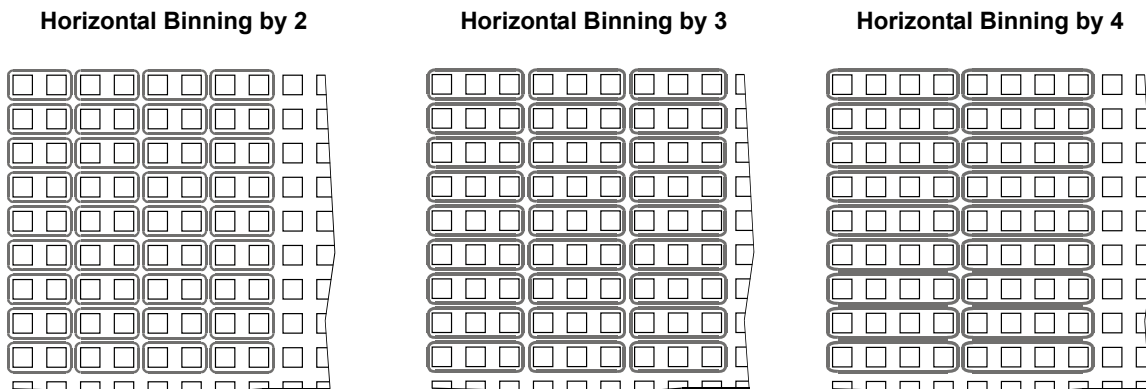


Fig. 59: Horizontal Binning

You can combine vertical and horizontal binning. This, however, may cause objects to appear distorted in the image. For more information about possible image distortion due to combined vertical and horizontal binning, see Section 10.7.2 on [page 203](#).

10.7.1 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.

Setting Binning Using Basler pylon

You can use the pylon API to set the Binning Vertical or the Binning Horizontal parameter value from within your application software. The following code snippet illustrates using the API to set the parameter values:

```
// Enable vertical binning by 2
Camera.BinningVertical.SetValue(2);

// Enable horizontal binning by 4
Camera.BinningHorizontal.SetValue(4);

// Disable vertical and horizontal binning
Camera.BinningVertical.SetValue(1);
Camera.BinningHorizontal.SetValue(1);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting Binning Using Direct Register Access

To enable vertical binning:

1. Set the value of the Binning Vertical register.
Setting the register's value to 2, 3, or 4 enables vertical binning by 2, vertical binning by 3, or vertical binning by 4 respectively. Setting the register's value to 1 disables vertical binning.

To enable horizontal binning:

1. Set the value of the Binning Horizontal register.
Setting the register's value to 2, 3, or 4 enables horizontal binning by 2, horizontal binning by 3, or horizontal binning by 4 respectively. Setting the register's value to 1 disables horizontal binning.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.7.2 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 reduces the resolution of the camera's imaging sensor. For example, the sensor in the acA2000-340km camera normally has a maximum resolution of 2048 (H) x 1088 (V) pixels. If you set this camera to use horizontal binning by 3 and vertical binning by 3, the effective maximum resolution of the sensor is reduced to 682 (H) by 362 (V). (Note that the dimensions of the sensor are not multiples of 3 and therefore can't be divided evenly by 3. To compensate for this, the values are rounded down to the nearest whole number.)

When you disable binning, the resolution will revert back to its original values.

Reduced Sensor Size

In the above example, the original resolution of the sensor can't be evenly divided by 3. Therefore, after enabling 3 by 3 binning, two columns and two lines remain. They are excluded from binning and image transmission which means that their image information is ignored.

The consequence of ignoring two columns and lines is that instead of using the full sensor size of 2048 (H) x 1088 (V), only an area of 2046 (H) x 1086 (V) is used for binning. Parts of the scene outside this area will not be part of the image you acquire unless you change the AOI settings accordingly. To do this, follow the steps below.

To ensure that the desired scene appears completely in a binned image:

1. Set the desired binning factor.
Settings made for offset, AOI width, and AOI height refer to the reduced number of rows and columns.
2. Acquire an image.
3. Check whether the desired scene appears completely in the image.
4. If necessary, adjust the settings for the virtual rows or columns to fully capture the desired scene.

When you disable binning, the sensor will revert back to its original size.

Binning's Effect on AOI Settings

When you have set the camera to use binning, the maximum area of interest (AOI) will be made up of the binned lines and columns, i.e. it is going to be smaller than the actual sensor's maximum AOI. You can think of this as a "virtual sensor". Also, any offsets refer to the virtual sensor's position.

For example, assume that you are using an acA2000-340km camera set for 3 by 3 binning as described above. In this case, the maximum AOI would be 682 columns by 362 lines. The AOI width and height parameters are adjusted automatically to reflect this. Likewise, any offsets you have defined before enabling binning will be adjusted automatically.

When you disable binning, the AOI will increase again but may be smaller than the original AOI. This happens when the original AOI values can't be evenly divided by the binning factor, leaving a remainder of lines and columns, which is then ignored when the AOI is increased again. Therefore, Basler recommends to always check the AOI and offset settings after disabling binning and, if necessary, to manually set the AOI to the desired values.

For more information about the area of interest (AOI) feature, see Section 10.4 on [page 156](#).

Possible Image Distortion

Objects will only appear undistorted in the image if the numbers of binned lines and columns are equal. With all other combinations, the imaged objects will appear distorted. If, for example, vertical binning by 2 is combined with horizontal binning by 4 the widths of the imaged objects will appear shrunk by a factor of 2 compared to the heights.

If you want to preserve the aspect ratios of imaged objects when using binning, you must use vertical and horizontal binning where equal numbers of lines and columns are binned, e.g. vertical binning by 3 combined with horizontal binning by 3.

Binning's Effect on Decimation

If vertical binning is used, vertical decimation (see below) is automatically disabled, and vice versa, i.e. if vertical decimation is used, vertical binning is disabled.

Horizontal binning works independently of the decimation vertical feature.

Binning's Effect on Stacked Zone Imaging

Using binning reduces the resolution of the camera's imaging sensor. If binning is enabled, the positions and sizes of any stacked zones are adjusted automatically to the binning factors applied: using horizontal binning will affect the AOI width and X offset and vertical binning will affect the Zone Height and Zone Y Offset values. If the stacked zone imaging parameter values can't be evenly divided by the binning factor, they are automatically rounded down to the nearest whole number.

When you disable binning, the zones will increase again but may be smaller than the original zones. This happens when the original zone values can't be evenly divided by the binning factor, leaving a remainder, which is then ignored when the zone is increased again. Therefore, Basler recommends to always check all related positions and sizes of the zones after disabling binning and, if necessary, to manually restore the zones to the desired values.

10.8 Vertical Decimation

With the vertical decimation feature (sub-sampling) you can reduce the number of rows of an acquired frame that are transmitted out of the camera. You can specify the extent of vertical decimation, i.e. you can define how many rows you want to be left out from transmission.

Examples

(Blue rows will be transmitted)

If vertical decimation is set to:

- 1: the complete frame will be transmitted out of the camera (no sub-sampling takes place); see Figure 60.
This applies to mono and color cameras.
- 2 for mono cameras: only every other row of the acquired frame will be transmitted out of the camera (Figure 61).
- 2 for color cameras: only every other pair of rows of the acquired frame will be transmitted out of the camera meaning that the Bayer filter stays properly aligned (Figure 62).

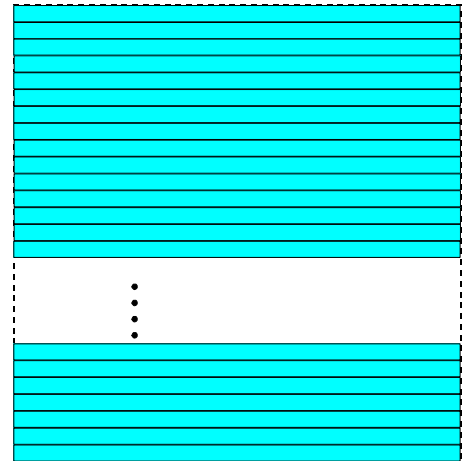


Fig. 60: Decimation Disabled (No Sub-sampling)

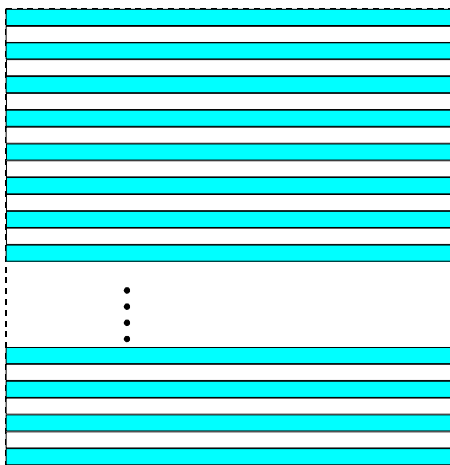


Fig. 61: Decimation of 2 (Mono Cameras)

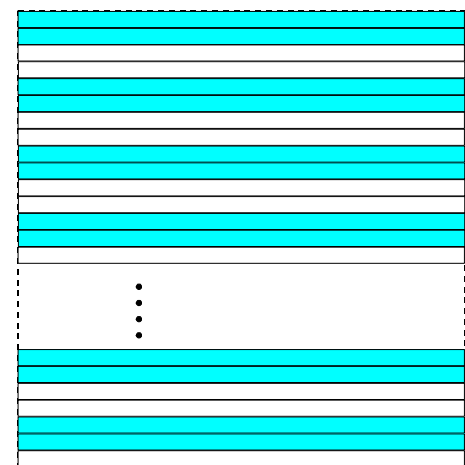


Fig. 62: Decimation of 2 (Color Cameras)

By using the vertical decimation feature, you can increase the frame rate of the camera.

Enabling Vertical Decimation

You can enable vertical decimation by setting the Vertical Decimation parameter. Setting the parameter's value to 1 disables vertical decimation.

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

```
// Enable Vertical Decimation by 8
Camera.DecimationVertical.SetValue(8);

// Disable Vertical Decimation
Camera.DecimationVertical.SetValue(1);
```

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

For more information about the pylon API and the pylon Viewer, see Chapter 3 on [page 18](#).

10.8.1 Considerations When Using Decimation

Reduced Vertical Resolution

Using vertical decimation effectively reduces the vertical resolution of the camera's imaging sensor. For example, the sensor in the acA2000-340km camera normally has a resolution of 2048 (H) x 1088 (V). This also represents the maximum AOI height. If you set this camera to use vertical decimation by 5, the effective resolution of the sensor is reduced to 2048 (H) by 218 (V). Note that the height value is rounded up to achieve an even number of rows.

If you reduce the vertical resolution by using the vertical decimation feature, you can increase the frame rate of the camera.

Possible Image Distortion

If you use vertical decimation, the resulting image will be distorted as the image information from the rows that are not transmitted will be missing. If, for example, vertical decimation is set to 2, the image will appear shrunk by a factor of 2 compared to an image without vertical decimation.

Binning and Vertical Decimation

If vertical binning is used, vertical decimation is automatically disabled, and vice versa, i.e. if vertical decimation is used, vertical binning is disabled.

Horizontal binning works independently of the decimation feature.

Decimation's Effect on AOI Settings

If vertical decimation is enabled, the camera automatically adapts the AOI settings to the reduced image size which is a result of transmitting only every n th line. In the example mentioned before, the AOI height is shrunk from 1088 to 218.



AOI height

If you use the vertical decimation feature and you set the decimation parameter back to 1, i.e. decimation is disabled, the AOI height may be smaller than the height set originally (determined by the pixel resolution in vertical direction).

In this case you can manually set the AOI height back to the maximum possible height.

10.9 Mirror Imaging

The camera's reverse X and reverse Y functions let you flip the captured images horizontally and/or vertically before they are transmitted from the camera.

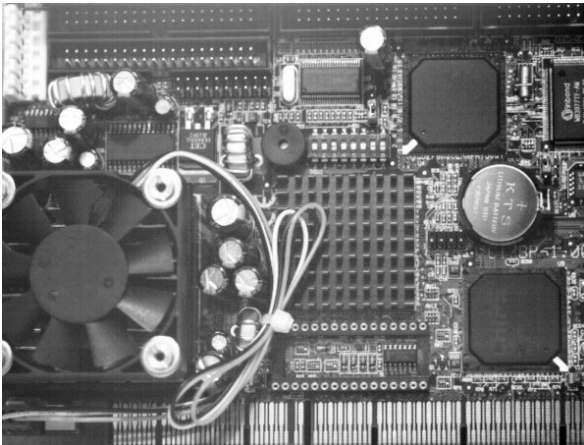
Note that the reverse X and reverse Y functions may both be enabled at the same time if so desired.

10.9.1 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 63 shows a normal image on the left and an image captured with reverse X enabled on the right.

Normal Image



Reverse X Mirror Image

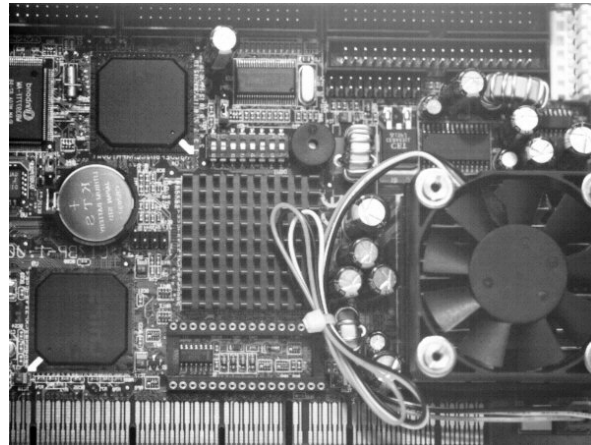


Fig. 63: Reverse X Mirror Imaging

The Effect of Reverse X on the Auto Function AOIs

If you are using the camera's auto functions, you should be aware of the effect that using the reverse X feature will have on the auto function AOIs. When reverse X is used, the position of the auto function AOIs relative to the sensor remains the same. As a consequence, each auto function AOI will include a different portion of the captured image depending on whether or not the reverse X feature is enabled. Figure 64 shows the effect of that reverse X mirroring will have on the auto function AOIs.

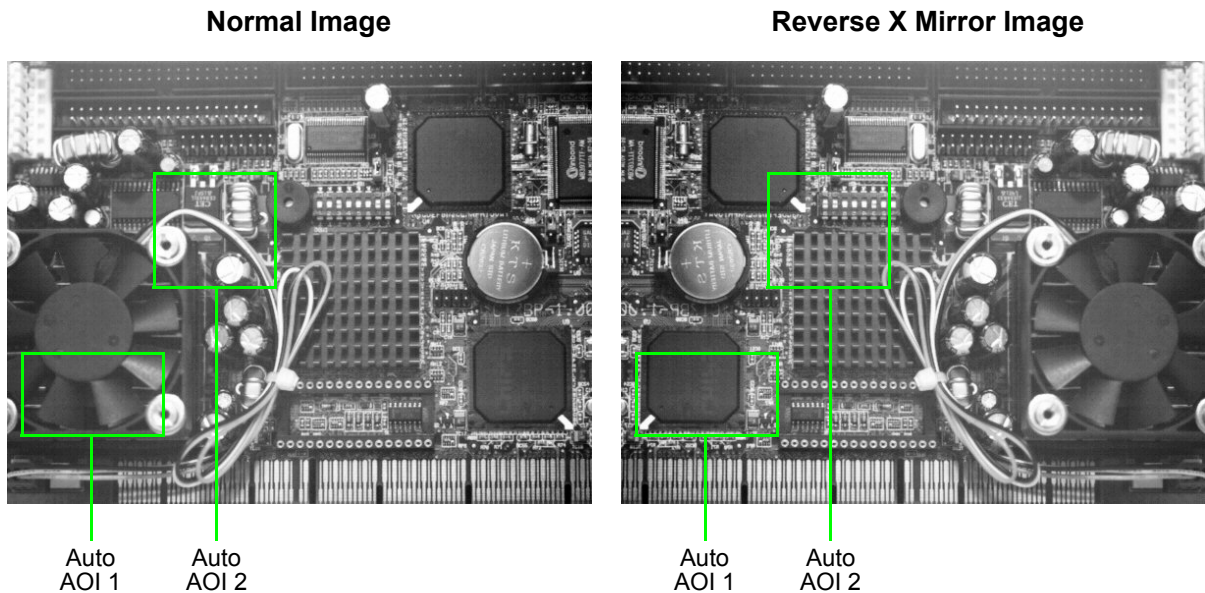


Fig. 64: Using Reverse X Mirror Imaging with Auto Functions Enabled

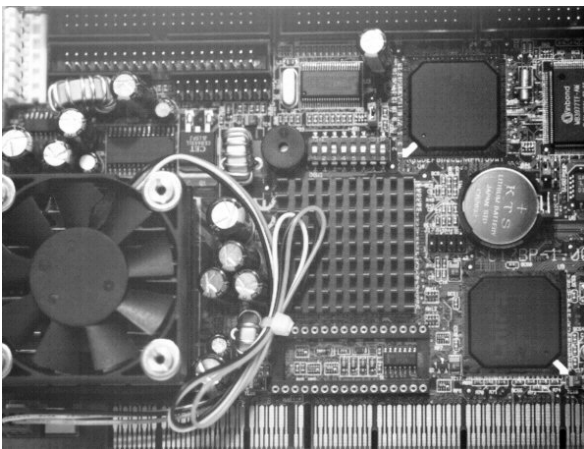
For more information about auto functions and auto function AOIs, see Section 10.11 on [page 219](#).

10.9.2 Reverse Y

The reverse Y feature is a vertical mirror image feature. When the reverse Y feature is enabled, the lines in a captured image will be swapped top-to-bottom. This means that the top line in the image will be swapped with the bottom line, the next-to-top line will be swapped with the next-to-bottom line, and so on.

Figure 63 shows a normal image on the left and an image captured with reverse Y enabled on the right.

Normal Image



Reverse Y Mirror Image

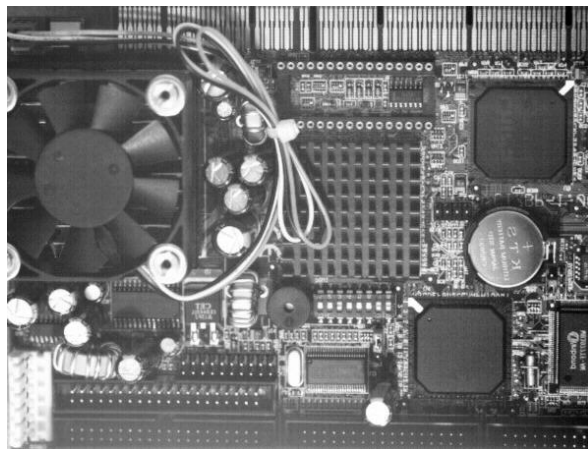


Fig. 65: Reverse Y Mirror Imaging

The Effect of Reverse Y on the Auto Function AOIs

If you are using the camera's auto functions, you should be aware of the effect that using the reverse Y feature will have on the auto function AOIs. When reverse Y is used, the position of the auto function AOIs relative to the sensor remains the same. As a consequence, each auto function AOI will include a different portion of the captured image depending on whether or not the reverse Y feature is enabled. Figure 66 shows the effect of that reverse Y mirroring will have on the auto function AOIs.

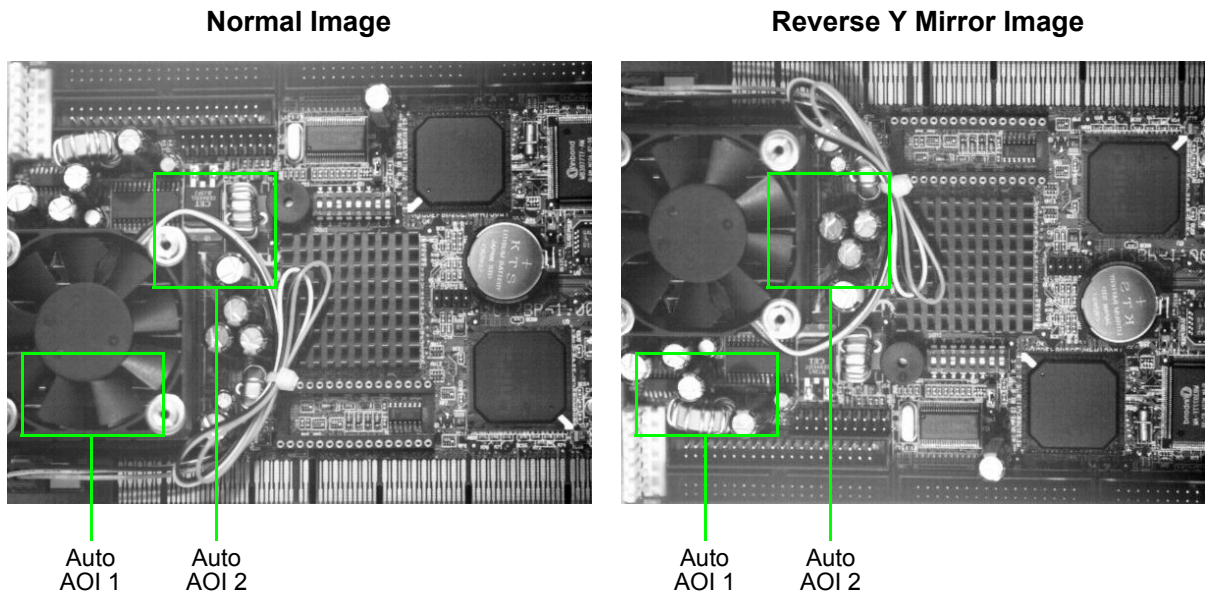


Fig. 66: Using Reverse Y Mirror Imaging with Auto Functions Enabled

For more information about auto functions and auto function AOIs, see Section 10.11 on [page 219](#).

10.9.3 Enabling Reverse X and Reverse Y

Enabling Reverse X and Y Using Basler pylon

You can enable the reverse X and reverse Y features by setting the Reverse X and the Reverse Y parameter values. You can use the pylon API to set the parameter values from within your application software. The following code snippet illustrates using the API to set the parameter values:

```
// Enable reverse X
Camera.ReverseX.SetValue(true);

// Enable reverse Y
Camera.ReverseY.SetValue(true);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Enabling Reverse X and Y Using Direct Register Access

To enable reverse X:

1. Set the value of the Reverse X register to 1 (enabled).

To enable reverse Y:

1. Set the value of the Reverse Y register to 1 (enabled).

For more information about direct register access, see Section 3.2 on [page 20](#).

10.10 Luminance Lookup Table

The type of electronics used on the camera allows the camera's sensor to acquire pixel values at a 12 bit depth. Normally, when a camera is set for a 12-bit pixel data format, the camera transmits the actual 12-bit pixel values reported by the sensor.


The luminance lookup table feature lets you create a custom 12-bit to 12-bit lookup table that maps the actual 12-bit values output from the sensor to substitute 12-bit values of your choice. When the lookup table is enabled, the camera will replace the actual pixel values output from the sensor with the substitute values from the table.

The lookup table has 4096 indexed locations with a 12-bit value stored at each index. The values stored in the table are used like this:

- When the sensor reports that a pixel has an actual 12-bit value of 0, the substitute 12-bit value stored at index 0 will replace the actual pixel value.
- The numbers stored at indices 1 through 7 are not used.
- When the sensor reports that a pixel has an actual 12-bit value of 8, the substitute 12-bit value stored at index 8 will replace the actual pixel value.
- The numbers stored at indices 9 through 15 are not used.
- When the sensor reports that a pixel has an actual 12-bit value of 16, the substitute 12-bit value stored at index 16 will replace the actual pixel value.
- The numbers stored at indices 17 through 23 are not used.
- When the sensor reports that a pixel has an actual 12-bit value of 24, the substitute 12-bit value stored at index 24 will replace the actual pixel value.
- And so on.

As you can see, the table does not include a defined 12-bit substitute value for every actual pixel value that the sensor can report. If the sensor reports an actual pixel value that is between two values that have a defined substitute, the camera performs a straight line interpolation to determine the substitute value that it should use. For example, assume that the sensor reports an actual pixel value of 12. In this case, the camera would perform a straight line interpolation between the substitute values at index 8 and index 16 in the table. The result of the interpolation would be used by the camera as the substitute.

Another thing to keep in mind about the table is that index 4088 is the last index that will have a defined substitute value associated with it (the values at indices 4089 through 4095 are not used). If the sensor reports an actual value greater than 4088, the camera will not be able to perform an interpolation. In cases where the sensor reports an actual value greater than 4088, the camera simply uses the 12-bit substitute value from index 4088 in the table.

	<p>If the imaging sensor bit depth is set to 10 bits, the sensor will only capture pixel data at 10-bit depth. In this case, the pixel values output from the sensor will be converted to 12-bit depth by padding the 10-bit values with two zeros as least significant bits. These converted 12-bit values will then be used as input to the lookup table.</p> <p>There is only one lookup table. When the lookup table is enabled on color cameras, the single table is used for red, green, and blue pixel values.</p> <p>The values for the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If you are using the lookup table feature, you must reenter the lookup table values after each camera startup or reset.</p>
---	--

The advantage of the luminance lookup table feature is that it lets a user 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 substitute values are designed so that the output of the camera increases linearly as the actual sensor output increases. The second graph is for a lookup table where the substitute values are designed so that the camera output increases quickly as the actual sensor output moves from 0 through 2048 and increases gradually as the actual sensor output moves from 2049 through 4096.

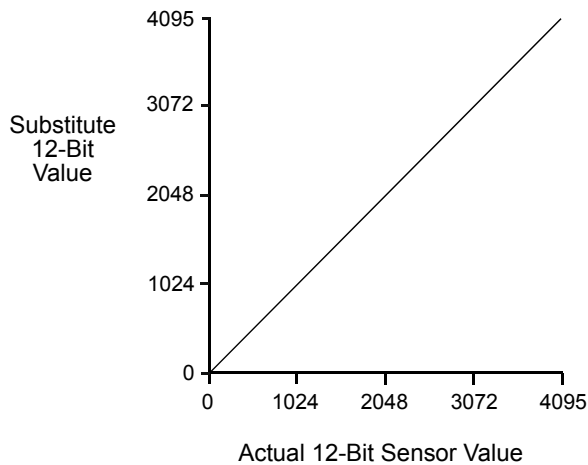


Fig. 67: Lookup Table with Values Mapped in a Linear Fashion

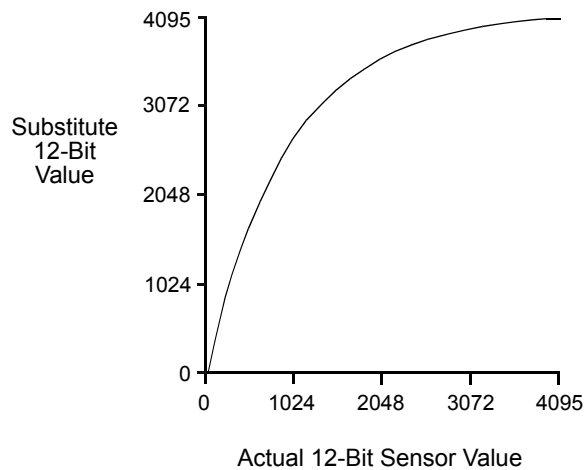


Fig. 68: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

Using the Luminance Lookup Table to Get 10-Bit or 8-Bit Output

As mentioned above, when the camera is set for a 12-bit pixel data format, the lookup table can be used to perform a 12-bit to 12-bit substitution. The lookup table can also be used in 12-bit to 10-bit or 12-bit to 8-bit fashion.

To use the table in 12-bit to 10-bit fashion, you enter 12-bit substitution values into the table and enable the table as you normally would. But instead of setting the camera for a 12-bit pixel data format, you set the camera for a 10-bit format (such as Mono 10). In this situation, the camera will first use the values in the table to do a 12-bit to 12-bit substitution. It will then truncate the least significant 2 bits of the substitute value and will transmit the remaining 10 most significant bits.

To use the table in 12-bit to 8-bit fashion, you enter 12-bit substitution values into the table and enable the table as you normally would. But instead of setting the camera for a 12-bit pixel data format, you set the camera for an 8-bit format (such as Mono 8). In this situation, the camera will first use the values in the table to do a 12-bit to 12-bit substitution. It will then truncate the least significant 4 bits of the substitute value and will transmit the remaining 8 most significant bits.

10.10.1 Entering LUT Values and Enabling the LUT

Entering Values and Enabling the LUT Using Basler pylon

You can enter values into the luminance lookup table (LUT) and enable the use of the lookup table by doing the following:

To enter values into the LUT and enable the use of the LUT:

1. Use the LUT Selector to select a lookup table. (Currently there is only one lookup table available, i.e., the "luminance" lookup table described above.)
2. Use the LUT Index parameter to select an index number.
3. Use the LUT Value parameter to enter the substitute value that will be stored at the index number that you selected in step 2.
4. Repeat steps 2 and 3 to enter other substitute values into the table as desired.
5. Use the LUT Enable parameter to enable the table.

You can use the pylon API to set the LUT Selector, the LUT Index parameter, and the LUT Value parameter from within your application software. The following code snippet illustrates using the API to set the selector and the parameter values:

```
// Select the lookup table
Camera.LUTSelector.SetValue(LUTSelector_Luminance);

// Write a lookup table to the device.
// The following lookup table causes an inversion of the sensor values
// (bright -> dark, dark -> bright)
for (int i = 0; i < 4096; i += 8)
{
    Camera.LUTIndex.SetValue(i);
    Camera.LUTValue.SetValue(4095 - i);
}
// Enable the lookup table
Camera.LUTEnable.SetValue(true);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Entering Values and Enabling the LUT Using Direct Register Access

When setting up the luminance lookup table via direct register access, two registers are involved: the LUT register and the LUT Enable register.

The LUT register is simply an array register that holds the 4096 12-bit values described earlier in this section. As a first step to using the lookup table feature. You must populate this register with 12-bit values.

Once the LUT register has been populated, you can enable the use of the lookup table by setting the value of the LUT Enable register to 1 (enabled).

For more information about direct register access, see Section 3.2 on [page 20](#).

10.11 Auto Functions

Auto functions control image properties and are the "automatic" counterparts of certain features, such as the gain feature or the white balance feature, which normally require "manually" setting the related parameter values. Auto functions are particularly useful when an image property must be adjusted quickly to achieve a specific target value and when a specific target value must be kept constant in a series of images.

An Auto Function Area of Interest (Auto Function AOI) lets you designate a specific part of the image as the base for adjusting an image property. Each auto function uses the pixel data from an Auto Function AOI for automatically adjusting a parameter value and, accordingly, for controlling the related image property. Some auto functions always share an Auto Function AOI.

An auto function automatically adjusts a parameter value until the related image property reaches a target value,.

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 10.7 on [page 201](#).

10.11.1 Auto Function Operating Modes

The following auto function modes of operation are available:

- All auto functions provide the "once" mode of operation. When the "once" mode of operation is selected, the parameter values are automatically adjusted until the related image property reaches the target value. After the automatic parameter value adjustment is complete, the auto function will automatically be set to "off" and the new parameter value will be applied to the following images.

The parameter value can be changed by using the "once" mode of operation again, by using the "continuous" mode of operation, or by manual adjustment.


	If an auto function is set to the "once" operation mode and if the circumstances will not allow reaching a target value for an image property, the auto function will try to reach the target value for a maximum of 30 images and will then be set to "off".
---	---


- Some auto functions also provide a "continuous" mode of operation where the parameter value is adjusted repeatedly while images are acquired.

Depending on the current frame rate, the automatic adjustments will usually be carried out for every or every other image.

The repeated automatic adjustment will proceed until the "once" mode of operation is used or until the auto function is set to "off", in which case the parameter value resulting from the latest automatic adjustment will operate, unless the parameter is manually adjusted.

- When an auto function is set to "off", the parameter value resulting from the latest automatic adjustment will operate, unless the parameter is manually adjusted.

	You can enable auto functions and change their settings while the camera is capturing images ("on the fly").
---	--

	If you have set an auto function to "once" or "continuous" operation mode while the camera was continuously capturing images, the auto function will become effective with a short delay and the first few images may not be affected by the auto function.
---	---

10.11.2 Auto Function AOIs

Each auto function uses the pixel data from an Auto Function AOI for automatically adjusting a parameter value, and accordingly, for controlling the related image property. Some auto functions always share an Auto Function AOI and some auto functions can use their own individual Auto Function AOIs. Within these limitations, auto functions can be assigned to Auto Function AOIs as desired.

Each Auto Function AOI has its own specific set of parameter settings, and the parameter settings for the Auto Function AOIs are not tied to the settings for the AOI that is used to define the size of captured images (Image AOI). For each Auto Function AOI, you can specify a portion of the sensor array and only the pixel data from the specified portion will be used for auto function control. Note that an Auto Function AOI can be positioned anywhere on the sensor array.

An Auto Function AOI is referenced to the top left corner of the sensor array. The top left corner of the sensor array is designated as column 0 and row 0 as shown in Figure 69.

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 69.

Only the pixel data from the area of overlap between the Auto Function AOI defined by your settings and the Image AOI will be used by the related auto function.

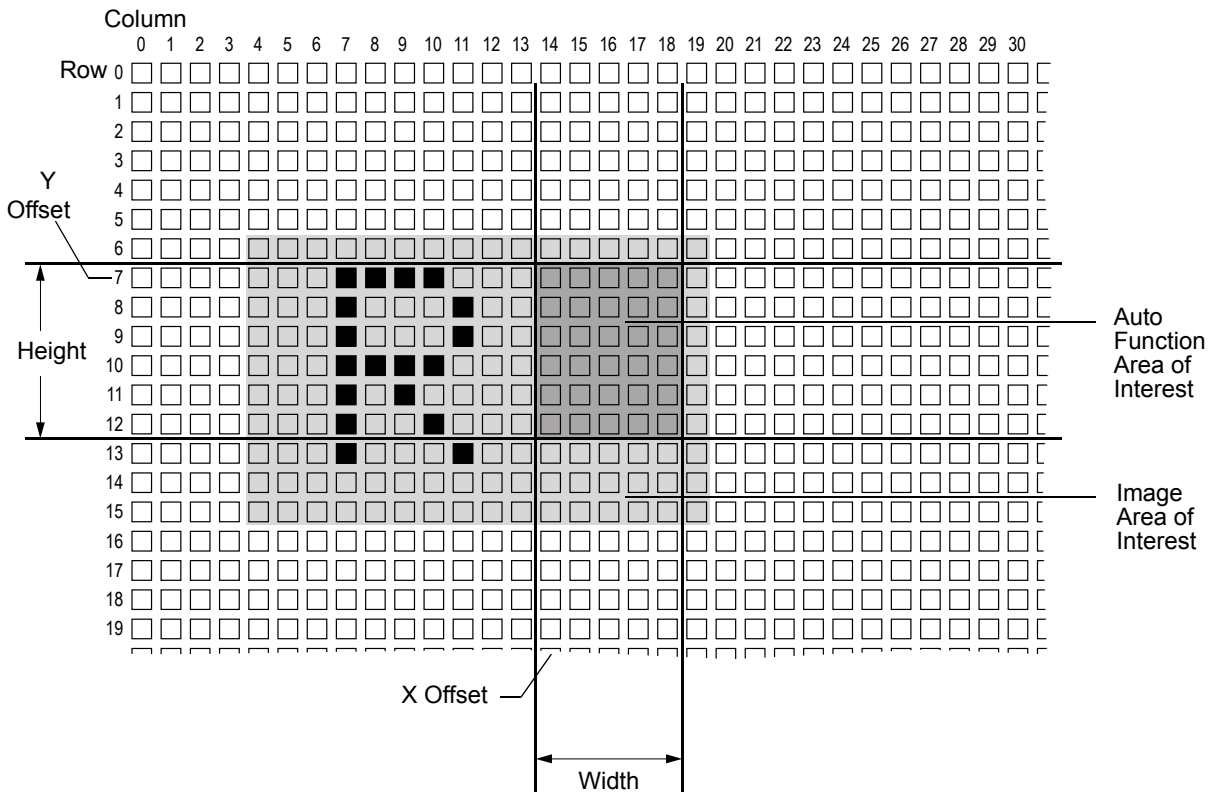


Fig. 69: Auto Function Area of Interest and Image Area of Interest

10.11.2.1 Assignment of an Auto Function to an Auto Function AOI

By default, the Gain Auto and the Exposure Auto auto functions are assigned to Auto Function AOI 1 and the Balance White Auto auto function is assigned to Auto Function AOI 2. The assignments can, however, be set as desired. For example, the Balance White Auto auto function can be assigned to Auto Function AOI 1 or all auto functions can be assigned to the same Auto Function AOI.



We strongly recommend that you do not assign an auto function to more than one Auto Function AOI even though this can be done.

One limitation must be kept in mind: For the purpose of making assignments, the Gain Auto and the Exposure Auto auto functions are always considered as a single "Intensity" auto function and therefore the Auto Function AOI assignment is always identical for both auto functions. For example, if you assign the "Intensity" auto function to Auto Function AOI 2 the Gain Auto and the Exposure Auto auto functions are both assigned to Auto Function AOI 2. This does not imply, however, that the Gain Auto and the Exposure Auto auto functions must always be used at the same time.

Assigning an Auto Function to an Auto Function AOI Using Basler pylon

You can assign auto functions to Auto Function AOIs from within your application software by using the pylon API.

As an example, the following code snippet illustrates using the API to assign the Gain Auto and Exposure Auto auto function – considered as a single "Intensity" auto function – and the Exposure Auto auto function to Auto Function AOI 1.

The snippet also illustrates disabling the unused Auto Function AOI 2 to avoid assigning any auto function to more than one Auto Function AOI.

```
// Select Auto Function AOI 1
// Assign auto functions to the selected Auto Function AOI
Camera.AutoFunctionAOISelector.SetValue(AutoFunctionAOISelector_AOI1);
Camera.AutoFunctionAOIUsageIntensity.SetValue(true);
Camera.AutoFunctionAOIUsageWhiteBalance.SetValue(true);

// Select the unused Auto Function AOI 2
// Disable the unused Auto Function AOI
Camera.AutoFunctionAOISelector.SetValue(AutoFunctionAOISelector_AOI2);
Camera.AutoFunctionAOIUsageIntensity.SetValue(false);
Camera.AutoFunctionAOIUsageWhiteBalance.SetValue(false);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Assigning an Auto Function to an Auto Function AOI Using Direct Register Access

To assign an auto function to Auto Function AOI 1:

1. Set the value of the Auto AOI 1 Usage register.

To assign an auto function to Auto Function AOI 2:

1. Set the value of the Auto AOI 2 Usage register.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.11.2.2 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 an Auto Function AOI and the 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 of an Auto Function AOI and the Image AOI will be used by the auto function to control the image property of the entire image.

Different degrees of overlap are illustrated in [Figure 70](#). 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 70](#)), all 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 70](#)), 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 70](#)), 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 70](#)), the Auto Function will **not** control the image property. For details, see the sections below, describing the individual auto functions.



We strongly recommend completely including the Auto Function AOI within the Image AOI, or, depending on your needs, setting identical positions and sizes for the Auto Function AOIs and the Image AOI.



You can use auto functions when also using the reverse X and reverse Y mirroring features. For information about the behavior of Auto Function AOIs when also using the reverse X or reverse Y mirroring feature, see the "Mirror Image" section.

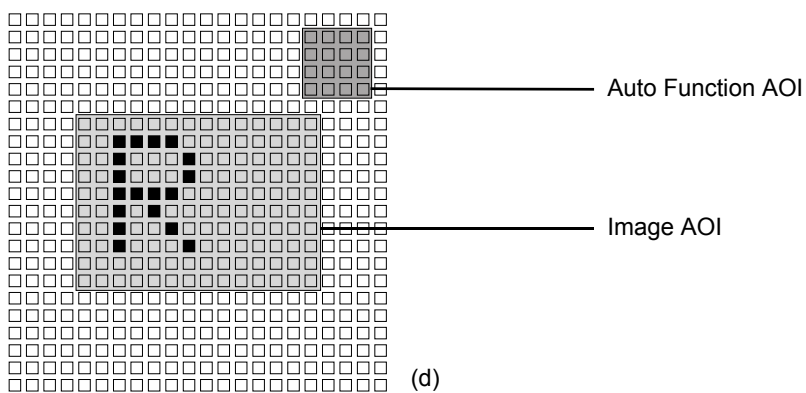
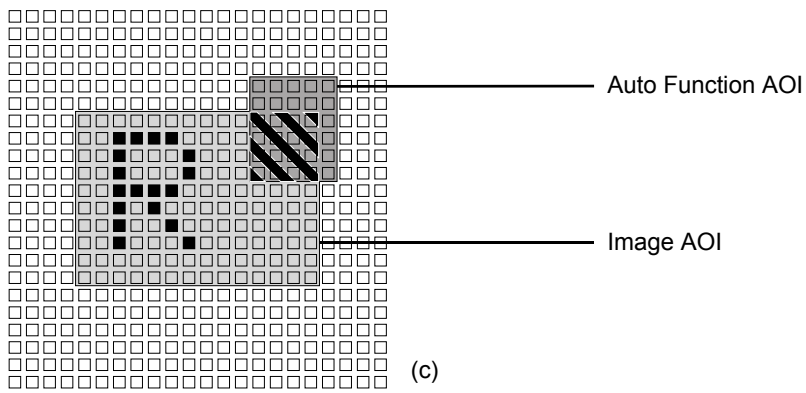
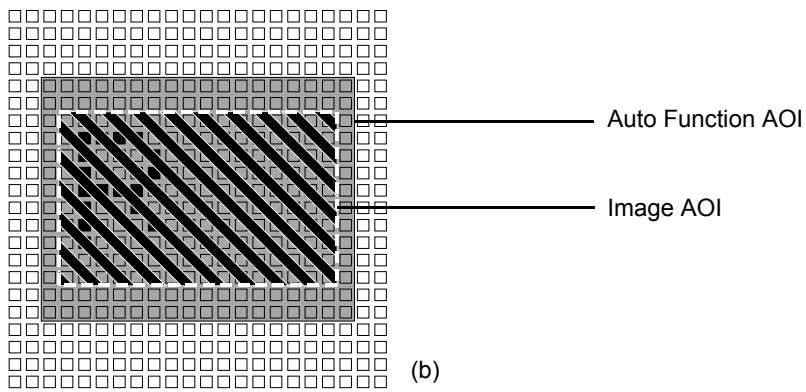
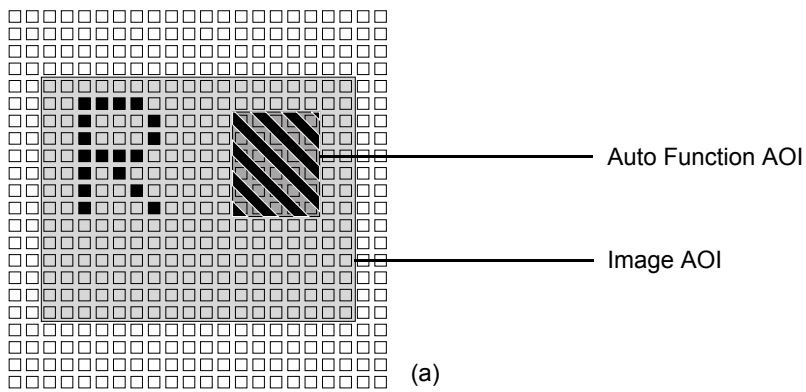


Fig. 70: Various Degrees of Overlap Between the Auto Function AOI and the Image AOI

10.11.2.3 Setting an Auto Function AOI Position and Size

Setting an Auto Function AOI position and size involves two steps.

To set the position and size of an Auto Function AOI:

1. Select the Auto Function AOI related to the auto function that you want to use.
2. Set the position and the size 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 position and the size 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 acA2000-340km, the sum of the X Offset setting plus the Width setting must not exceed 2048.
- 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 acA2000-340km, the sum of the X Offset setting plus the Width setting must not exceed 1088.



Normally, the X Offset, Y Offset, Width, and Height parameter settings for an Auto Function AOI refer to the physical columns and lines in the sensor. But if binning is enabled (monochrome cameras only), these parameters are set in terms of "virtual" columns and lines, i.e., the settings for an Auto Function AOI will refer to the binned lines and columns in the sensor and not to the physical lines in the sensor as they normally would.

For more information about the concept of a "virtual" sensor, see Section 10.10 on [page 214](#).

Setting an Auto Function AOI Position and Size Using Basler pylon

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 snippet illustrates using the API to select Auto Function AOI one and to get the maximum allowed settings for the Width and Height parameters. The snippet also illustrates setting the X Offset, Y Offset, Width, and Height parameter values.

```
// Select auto function AOI 1
// Set position and size of the selected auto function AOI
Camera.AutoFunctionAOISelector.SetValue(AutoFunctionAOISelector_AOI1);
Camera.AutoFunctionAOIOffsetX.SetValue(0);
Camera.AutoFunctionAOIOffsetY.SetValue(0);
Camera.AutoFunctionAOIWidth.SetValue(Camera.AutoFunctionAOIWidth.GetMax());
Camera.AutoFunctionAOIHeight.SetValue(Camera.AutoFunctionAOIHeight.GetMax());
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting an Auto Function AOI Position and Size Using Direct Register Access

To set the X Offset, Y Offset, Width, and Height for Auto Function 1:

1. Set the value of the Auto AOI 1 Left register to set the X offset.
2. Set the value of the Auto AOI 1 Top register to set the Y offset.
3. Set the value of the Auto AOI 1 Width register to set the width.
4. Set the value of the Auto AOI 1 Height register to set the height.

To set the X Offset, Y Offset, Width, and Height for Auto Function 2:

1. Set the value of the Auto AOI 2 Left register to set the X offset.
2. Set the value of the Auto AOI 2 Top register to set the Y offset.
3. Set the value of the Auto AOI 2 Width register to set the width.
4. Set the value of the Auto AOI 2 Height register to set the height.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.11.3 Using an Auto Function

To use an auto function:

1. Select an Auto Function AOI.
2. Assign the auto function you want to use to the selected Auto Function AOI.
3. Unassign the auto function you want to use from the other Auto Function AOI.
4. Set the position and size of the Auto Function AOI.
5. If necessary, set the lower and upper limits for the auto functions' parameter value.
6. If necessary, set the target value.
7. If necessary, set the auto function profile to define priorities between auto functions.
8. Enable the auto function by setting it to "once" or "continuous".

For more information about the individual settings, see the next sections that describe the individual auto functions.

10.11.4 Gain Auto

Gain Auto is the "automatic" counterpart to manually setting the Gain Raw parameter. When the gain auto function is operational, the camera will automatically adjust the Gain Raw parameter value within set limits until a target average gray value for the pixel data from the related Auto Function AOI is reached.

The gain auto function can be operated in the "once" and "continuous" modes of operation.

If the related Auto Function AOI does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from the Auto Function AOI will not be used to control the gain. Instead, the current manual setting for the Gain Raw parameter value will control the gain.

The gain auto function and the exposure auto function can be used at the same time. In this case, however, you must also set the auto function profile feature.

For more information about the "manual" gain settings, see Section 10.1 on [page 152](#).

For more information about the auto function profile, see Section 10.11.6 on [page 235](#).

The limits within which the camera will adjust the Gain Raw parameter are defined by the Auto Gain Raw Upper Limit and the Auto Gain Raw Lower Limit parameters. The minimum and maximum allowed settings for the Auto Gain Raw Upper Limit and Auto Gain Raw Lower Limit parameters depend on the current pixel data format, on the current settings for binning, and on whether or not the parameter limits for manually setting the gain feature are disabled.

The Auto Target Value parameter defines the target average gray value that the gain auto function will attempt to achieve when it is automatically adjusting the Gain Raw value. The target average gray value can range from 0 (black) to 255 (white) when the camera is set for an 8-bit pixel format, from 0 (black) to 1023 (white) when the camera is set for a 10-bit pixel format, or from 0 (black) to 4095 (white) when the camera is set for a 12-bit pixel format.

Setting the Gain Auto Function Using Basler pylon

Setting the gain auto function using Basler pylon involves several steps.

To set the gain auto function:

1. Select the Auto Function AOI to which the Gain Auto function is assigned.
2. Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
3. Set the Gain Selector to All.
4. Set the value of the Auto Gain Raw Lower Limit and Auto Gain Raw Upper Limit parameters.
5. Set the value of the Auto Target Value parameter.
6. Set the value of the Gain Auto parameter for the "once" or the "continuous" mode of operation.

You can set the gain auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the exposure auto functionality:

```
// Select Auto Function AOI to which the Gain Auto function is assigned
// For this example, assume that the Gain Auto function is assigned
// to Auto AOI 1
Camera.AutoFunctionAOISelector.SetValue(AutoFunctionAOISelector_AOI1);

// Set the position and size of selected auto function AOI. In this example, we set
// auto function AOI to cover the entire sensor.
Camera.AutoFunctionAOIOffsetX.SetValue(0);
Camera.AutoFunctionAOIOffsetY.SetValue(0);
Camera.AutoFunctionAOIWidth.SetValue(Camera.AutoFunctionAOIWidth.GetMax());
Camera.AutoFunctionAOIHeight.SetValue(Camera.AutoFunctionAOIHeight.GetMax());

// Select gain all and set the upper and lower gain limits for the gain
// auto function.
Camera.GainSelector.SetValue(GainSelector_All);
Camera.AutoGainRawLowerLimit.SetValue(Camera.GainRaw.GetMin());
Camera.AutoGainRawUpperLimit.SetValue(Camera.GainRaw.GetMax());

// Set target gray value for the gain auto function.
Camera.AutoTargetValue.SetValue(16);

// Set the mode of operation for gain auto function.
Camera.GainAuto.SetValue(GainAuto_Once);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Gain Auto Function Using Direct Register Access

Setting the gain auto function via direct register access involves several steps.

To set the gain auto function:

1. Set the position and size of Auto Function AOI 1 by setting the following values:
 - Auto AOI 1 Left register
 - Auto AOI 1 Top register
 - Auto AOI 1 Width register
 - Auto AOI 1 Height register

(**Note:** This step assumes that the Gain Auto function is assigned to Auto Function AOI 1. If the Gain Auto function is assigned to Auto Function AOI 2, you would set the Auto Function AOI 2 registers.)
2. Set the value of the Auto Gain Lower Limit register and the Auto Gain Upper Limit register.
3. Set the value of the Auto Target Value register.
4. Enable the gain auto function by selecting the value of the Gain Auto register to the Once or the Continuous mode of operation as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.11.5 Exposure Auto



The exposure auto function will not work if the camera's exposure mode is set to trigger width. For more information about the trigger width exposure mode, see Section 7.3.3.2 on [page 83](#).

Exposure Auto is the "automatic" counterpart to manually setting the Exposure Time Abs parameter. The exposure auto function automatically adjusts the Exposure Time Abs parameter within set limits until a target average gray value for the pixel data from the related Auto Function AOI is reached.

The exposure auto function can be operated in the "once" and "continuous" modes of operation.

If the related Auto Function AOI does not overlap the Image AOI (see the "Auto Function AOI" section), the pixel data from the Auto Function AOI will not be used to control the exposure time. Instead, the current manual setting for the Exposure Time Abs parameter will control the exposure time.

The exposure auto function and the gain auto function can be used at the same time. In this case, the auto function profile feature also takes effect. By default, the auto function profile feature minimizes gain.

When trigger width exposure mode is selected, the exposure auto function is not available.

For more information about the "manual" exposure time setting, see Section 7.4 on [page 89](#).

For more information about the trigger width exposure mode, see Section 7.3.3.2 on [page 83](#).

For more information about the auto function profile Section 10.11.6 on [page 235](#).

The limits within which the camera will adjust the Auto Exposure Time Abs parameter are defined by the Auto Exposure Time Abs Upper Limit and the Auto Exposure Time Abs Lower Limit parameters. The current minimum and the maximum allowed settings for the Auto Exposure Time Abs Upper Limit parameter and the Auto Exposure Time Abs Lower Limit parameters depend on the minimum allowed and maximum possible exposure time for your camera model.

The Auto Target Value parameter defines the target average gray value that the exposure auto function will attempt to achieve when it is automatically adjusting the Exposure Time Abs value. The target average gray value can range from 0 (black) to 255 (white) when the camera is set for an 8-bit pixel format, from 0 to 1023 when the camera is set for a 10-bit pixel format, and from 0 to 4095 when the camera is set for a 12-bit pixel format.



If the Exposure Time Abs Upper Limit parameter is set to a sufficiently high value, the camera's maximum allowed frame rate may be decreased.

Setting the Exposure Auto Function Using Basler pylon

Setting the exposure auto function using Basler pylon involves several steps.

To set the exposure auto function:

1. Select the Auto Function AOI to which the Exposure Auto function is assigned.
2. Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
3. Set the value of the Auto Exposure Time Abs Lower Limit and Auto Exposure Time Abs Upper Limit parameters.
4. Set the value of the Auto Target Value parameter.
5. Set the value of the Exposure Auto parameter for the "once" or the "continuous" mode of operation.

You can set the exposure auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the exposure auto functionality:

```
// Select Auto Function AOI to which the Exposure Auto function is assigned
// For this example, assume that the Exposure Auto function is
// assigned to Auto AOI 1
Camera.AutoFunctionAOISelector.SetValue(AutoFunctionAOISelector_AOI1);

// Set the position and size of selected auto function AOI. In this example, we set
// auto function AOI to cover the entire sensor.
Camera.AutoFunctionAOIOffsetX.SetValue(0);
Camera.AutoFunctionAOIOffsetY.SetValue(0);
Camera.AutoFunctionAOIWidth.SetValue(Camera.AutoFunctionAOIWidth.GetMax());
Camera.AutoFunctionAOIHeight.SetValue(Camera.AutoFunctionAOIHeight.GetMax());

// Set the exposure time limits for the exposure auto function
Camera.AutoExposureTimeAbsLowerLimit.SetValue(1000);
Camera.AutoExposureTimeAbsUpperLimit.SetValue(1.0E6);

// Set target gray value for the exposure auto function
// (If gain auto is enabled, this target is also used for gain auto control)
Camera.AutoTargetValue.SetValue(32);

// Set the mode of operation for the exposure auto function
Camera.ExposureAuto.SetValue(ExposureAuto_Continuous);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Exposure Auto Function Using Direct Register Access

Setting the exposure auto function via direct register access involves several steps.

To set the exposure auto function:

1. Set the position and size of Auto Function AOI 1 by setting the following values:
 - Auto AOI 1 Left register
 - Auto AOI 1 Top register
 - Auto AOI 1 Width register
 - Auto AOI 1 Height register

(**Note:** This step assumes that the Exposure Auto function has been assigned to Auto Function AOI 1. If the Exposure Auto function is assigned to Auto Function AOI 2, you would set the Auto Function AOI 2 registers.)
2. Set the value of the Auto Exposure Lower Limit register and the Auto Exposure Upper Limit register.
3. Set the value of the Auto Target Value register.
4. Enable the gain auto function by selecting the value of the Exposure Auto register to the Once or the Continuous mode of operation as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.11.6 Auto Function Profile

If you want to use the gain auto function and the exposure auto function at the same time, the auto function profile feature also takes effect. The auto function profile specifies whether the gain or the exposure time will be kept as low as possible when the camera is making automatic adjustments to achieve a target average gray value for the pixel data from the Auto Function AOI that was related to the gain auto and the exposure auto function. By default, the auto function profile feature minimizes gain.

If you want to use the gain auto and the exposure auto functions at the same time, you should set both functions for the continuous mode of operation.

Setting the Auto Function Profile Using Basler pylon

Setting the camera with Basler pylon to use the gain auto function and the exposure auto function at the same time involves several steps.

To use the gain auto function and the exposure auto function at the same time:

1. Set the value of the Auto Function Profile parameter to specify whether gain or exposure time will be minimized during automatic adjustments.
2. Set the value of the Gain Auto parameter to the "continuous" mode of operation.
3. Set the value of the Exposure Auto parameter to the "continuous" mode of operation.

You can set the auto function profile from within your application software by using the pylon API. The following code snippet illustrates using the API to set the auto function profile. As an example, Gain Auto is set to be minimized during adjustments:

```
// Use GainAuto and ExposureAuto simultaneously
Camera.AutoFunctionProfile.SetValue(AutoFunctionProfile_GainMinimum);
Camera.GainAuto.SetValue(GainAuto_Continuous);
Camera.ExposureAuto.SetValue(ExposureAuto_Continuous);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Auto Function Profile Using Direct Register Access

Setting the camera to use the gain auto function and the exposure auto function at the same time via direct register access involves several steps.

To use the gain auto function and the exposure auto function at the same time:

1. Set the value of the Auto Function Profile register to specify whether gain or exposure time will be minimized during automatic adjustments.
2. Set the value of the Gain Auto register for the Continuous mode of operation.
3. Set the value of the Exposure Auto register for the Continuous mode of operation.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.11.7 Balance White Auto

Balance White Auto is the "automatic" counterpart to manually setting the white balance. The balance white auto function is only available on color models.

Automatic white balancing is a two-step process. First, the Balance Ratio Abs parameter values for red, green, and blue are each set to 1.0. Next, the Balance Ratio Abs parameter values are automatically adjusted such that the average values for the "red", "green", and "blue" pixels are all the same. During this process, the color that needs the smallest amount of gain will be adjusted to a value of 1.0.

The balance white auto function uses the Auto Function AOI related to the Balance White Auto function. The balance white auto function can be operated in the "once" mode of operation and in the "continuous" mode of operation. For information about the "once" mode of operation and the "continuous" mode of operation, see Section 10.11.1 on [page 220](#).

If the related Auto Function AOI does not overlap the Image AOI (see the "Auto Function AOIs" section) the pixel data from the Auto Function AOI will not be used to control the white balance of the image. However, as soon as the Balance White Auto function is set to the "once" operation mode, the Balance Ratio parameter values for red, green, and blue are each set to 1.0. These settings will then control the white balance of the image.

For more information about the "manual" white balance settings, see Section 8.3.1 on [page 130](#).

Setting the Balance White Auto Function Using Basler pylon

Setting the balance white auto function using Basler pylon involves several steps.

To set the balance white auto function:

1. Select the Auto Function AOI to which the Balance White Auto is assigned.
2. Set the value of the Offset X, Offset Y, Width, and Height parameters for the AOI.
3. Set the value of the Exposure Auto parameter to the "once" or the "continuous" mode of operation.

You can set the white balance auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the balance auto functionality:

```
// Select Auto Function AOI to which the Balance White Auto function is assigned
// For this example, assume that the Balance White Auto function is
// assigned to Auto AOI 2
Camera.AutoFunctionAOISelector.SetValue(AutoFunctionAOISelector_AOI2);

// Set the position and size of selected auto function AOI. In this example, we set
// auto function AOI to cover the entire sensor.
Camera.AutoFunctionAOIOffsetX.SetValue(0);
Camera.AutoFunctionAOIOffsetY.SetValue(0);
Camera.AutoFunctionAOIWidth.SetValue(Camera.AutoFunctionAOIWidth.GetMax());
Camera.AutoFunctionAOIHeight.SetValue(Camera.AutoFunctionAOIHeight.GetMax());

// Set mode of operation for balance white auto function
```

```
Camera.BalanceWhiteAuto.SetValue(BalanceWhiteAuto_Once);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting the Balance White Auto Function Using Direct Register Access

Setting the balance white auto function via direct register access involves several steps.

To set the balance white auto function:

1. Set the position and size of Auto Function AOI 2 by setting the following values:
 - Auto AOI 2 Left register
 - Auto AOI 2 Top register
 - Auto AOI 2 Width register
 - Auto AOI 2 Height register

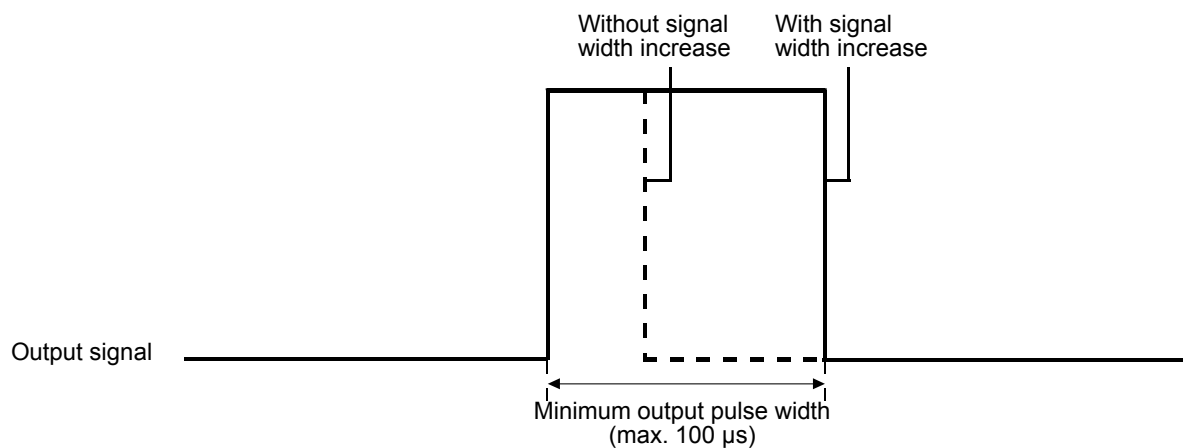
(**Note:** This step assumes that the Balance White Auto function has been assigned to Auto Function AOI 2. If the Exposure Auto function is assigned to Auto Function AOI 1, you would set the Auto Function AOI 1 registers.)
2. Enable the balance white auto function in the once mode of operation by selecting the value of the Balance White Auto register to Once.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.12 Minimum Output Pulse Width

An output signal sent by the camera may be too narrow for some receivers to be detected. To ensure reliable detection, the Minimum Output Pulse Width feature allows you to increase the signal width to a set minimum width:

- If the signal width of the original output signal is narrower than the set minimum the Minimum Output Pulse Width feature will increase the signal width to the set minimum before the signal is sent out of the camera (see the figure below).
- If the signal width of the original output signal is equal to or wider than the set minimum the Minimum Output Pulse Width feature will have no effect. The signal will be sent out of the camera with unmodified signal width.



Not to Scale

Fig. 71: Increasing the Signal Width of an Output Signal

Setting the Minimum Output Pulse Width

The Line Output Pulse Min. Width parameter sets the output signals of the GPIO line (assuming it is set as an output) or the CL Spare line to a minimum width. The parameter is set in microseconds and can be set in a range from 0 to 100 μ s.

To set the minimum output pulse width parameter value:

- Use the Line Selector to select the GPIO line or the CL Spare line.
- Set the value of the Line Output Pulse Min. Width parameter.

You can set the Line Selector and the value of the Line Output Pulse Min. Width parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to select an output line and to set the parameter value for the line:

```
// Select the GPIO line
Camera.LineSelector.SetValue(LineSelector_Line1);
// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);

// Select the CL Spare line
Camera.LineSelector.SetValue(LineSelector_ClSpare);
// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);
```

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

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

For more information about selecting the source signal for an output line of the camera, see Section 6.2.3 on [page 49](#).

10.13 Error Detection

10.13.1 LED Indicator

The LED indicator on the back of the camera can show a red and a green light. The LED indicator signals the camera's current condition as shown in Table 14.

LED State	Status Indication
Off	No power to the camera.
Solid Green	The camera has booted up successfully and is OK.
Flashing Green	The camera is set to expect an external trigger signal on an input, but no trigger signal is present.
Flashing Green/Red	An error condition has been detected that users may be able to solve by themselves. (See the next section for details.)
Flashing Red	Internal error. Contact Basler technical support.

Table 14: LED Indications



During the camera boot process, both the red and the green lights will light up.

10.13.2 Error Codes

The camera can detect several errors that can be corrected by the users themselves. If one of these errors is present, the camera will set an error code and the LED indicator will be flashing red and green alternately.

The following table indicates the available error codes:

Code	Condition	Meaning
0	No Error	The camera has not detected any errors since the last time that the error memory has been cleared.
1	Overtrigger	An overtrigger has occurred. The user has applied an acquisition start trigger to the camera when the camera was not in a waiting for acquisition start condition. The user has applied a frame start trigger to the camera when the camera was not in a waiting for frame start condition.
2	User Set	An error occurred when attempting to load a user set. Typically, this means that the user set contains an invalid value. Try loading a different user set.
3	Invalid Parameter	A parameter is set out of range or in an otherwise invalid manner. (Typically, this error only occurs when the user is setting parameters via direct register access.)

Table 15: Error Codes

When the camera detects such a user-correctable error, it records the appropriate error code in an error memory. If two or three different errors have occurred, the camera will store the code for each type of error that it has detected (it will store one occurrence of each code no matter how many times it has detected the corresponding error).

To handle error codes:

1. Read the value of the Last Error parameter.
The Last Error parameter will indicate the last error code stored in the memory.
2. Execute the Clear Last Error Command to clear the last error code from the memory.
3. Continue reading and clearing the last error until the parameter indicates a No Error code.

Reading and Clearing the Error Codes Using Basler pylon

You can use the pylon API to read the value of the Last Error parameter and to execute a Clear Last Error command from within your application software. The following code snippets illustrate using the API to read the parameter value and execute the command:

```
// Read the value of the last error code in the memory
LastErrorEnums lasterror = Camera.LastError.GetValue();

// Clear the value of the last error code in the memory
Camera.ClearLastError.Execute();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Reading and Clearing the Error Codes Using Direct Register Access

To get the value of the last error code in the memory:

1. Read the value of the Last User Error register.

To clear the value of the last error code in the memory:

1. Set the value of the Clear Last User Error register to 1.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.14 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 ADCs. Five 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, and 3, the camera's 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 camera's 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 Using Basler pylon

With Basler pylon, 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 enable one of the test images or to "test image off".

You can use the pylon API to set the Test Image Selector from within your application software. The following code snippets illustrate using the API to set the selector:

```
// Set for no test image
Camera.TestImageSelector.SetValue(TestImageSelector_Off);

// Set for test image 1
Camera.TestImageSelector.SetValue(TestImageSelector_Testimage1);
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Enabling a Test Image Using Direct Register Access

To enable a test image:

1. Set the value of the Test image Selector Register to Test Image 1, 2, 3, 4, or 5 as desired.

To disable test images:

1. Set the value of the Test image Selector Register to Off.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.14.1 Test Image Descriptions

Test Image 1: Fixed Diagonal Gray Gradient (8 Bit)

The 8-bit fixed diagonal gray gradient test image is best suited for use when the camera is set for monochrome 8-bit output. The test image consists of fixed diagonal gray gradients ranging from 0 to 255.

If the camera is set for 8-bit output and is operating at full resolution, test image 1 will look similar to Figure 72.

The mathematical expression for this test image:

$$\text{Gray Value} = [\text{column number} + \text{row number}] \text{ MOD } 256$$

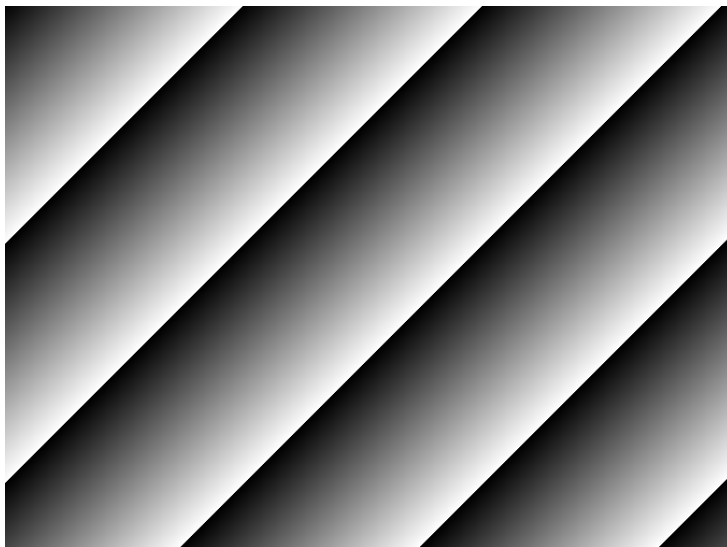


Fig. 72: Test Image One

Test Image 2: Moving Diagonal Gray Gradient (8 Bit)

The 8-bit moving diagonal gray gradient test image is similar to test image 1, apart from that it is moving. The image moves by one pixel from right to left whenever a new image acquisition is initiated. The test pattern uses a counter that increases by one for each new image acquisition.

The mathematical expression for this test image is:

$$\text{Gray Value} = [\text{column number} + \text{row number} + \text{counter}] \text{ MOD } 256$$

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 increases by one for each new image acquisition.

The mathematical expression for this test image is:

$$\text{Gray Value} = [\text{column number} + \text{row number} + \text{counter}] \text{ MOD } 4096$$

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.

10.15 Device Information Parameters

Each camera includes a set of "device information" parameters. These parameters provide 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, acA2000-140km.
- 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. This is usually the material number of the device.
- Device ID (read-only): Typically 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. The name will also be visible in the "friendly name" field of the device information objects returned by pylon's device enumeration procedure.
- Device Firmware Version (read-only): Contains the version of the firmware the camera.
- Device Scan Type (read-only): Contains the scan type of the camera, for example, line scan or area scan. The ace will always indicate 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.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Working with Device Information Parameters Using Basler pylon

You can use the pylon API to 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. 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 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 Firmware Version parameter
Pylon::String_t firmwareVersion = Camera.DeviceFirmwareVersion.GetValue();

// Read the Sensor Width parameter
int64_t sensorWidth = Camera.SensorWidth.GetValue();

// Read the Sensor Height parameter
int64_t sensorHeight = Camera.SensorHeight.GetValue();

// Read the Max Width parameter
int64_t maxWidth = Camera.WidthMax.GetValue();

// Read the Max Height parameter
int64_t maxHeight = Camera.HeightMax.GetValue();
```

You can also use the Basler pylon Viewer application to easily read the parameters and to read or write the Device User ID.

For more information about the pylon API and the pylon Viewer, see Section 3.2 on [page 20](#).

Working with Device Information Parameters Using Direct Register Access

When working with the camera via direct register access, you can do the following:

- Read the value in the Device Vendor Name register.
- Read the value in the Device Model Name register.
- Read the value in the Device Manufacturer Info register.
- Read the value in the Device Version register.
- Read the value in the Device ID register.
- Read the value in the Device User ID register
- Write the value of the Device User ID register.
- Read the value of the Device Firmware Version register.
- Read the value in the Device Scan Type register.
- Read the value in the Sensor Width register.
- Read the value in the Sensor Height register.
- Read the value in the Width Max register.
- Read the value in the Height Max register.

For more information about via direct register access, see Section 3.2 on [page 20](#).

10.16 User-Defined Values

The camera can store five "user-defined values". These five values are 32-bit signed integer values that you can set and read as desired. They simply serve as convenient storage locations for the camera user and have no impact on the operation of the camera.

The five values are designated as Value 1, Value 2, Value 3, Value 4, and Value 5.

Setting User-Defined Values Using Basler pylon

Setting a user-defined value using Basler pylon involves two steps.

To set a user-defined value:

1. Set the User Defined Value Selector to Value 1, Value 2, Value 3, Value 4, or Value 5.
2. Set the User Defined Value parameter to the desired value for the selected value.

You can use the pylon API to set the User Defined Value Selector and the User Defined Value parameter value from within your application software. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set user defined value 1
Camera.UserDefinedValueSelector.SetValue(UserDefinedValueSelector_Value1);
Camera.UserDefinedValue.SetValue(1000);

// Set user defined value 2
Camera.UserDefinedValueSelector.SetValue(UserDefinedValueSelector_Value2);
Camera.UserDefinedValue.SetValue(2000);

// Get the value of user defined value 1
Camera.UserDefinedValueSelector.SetValue(UserDefinedValueSelector_Value1);
int64_t UserValue1 = Camera.UserDefinedValue.GetValue();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Setting User-Defined Values Using Direct Register Access

To set a user-defined value:

1. Set the value of the User Defined Value 1, 2, 3, 4, or 5 register as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.17 Configuration Sets

A configuration set is a group of values that contains all of the parameter settings needed to control the camera. There are three basic types of configuration sets: the active set, the default set, and user sets.

The Active Set

The active set contains the camera's current parameter settings and thus determines the camera's performance, that is, what your image currently looks like. When you change parameter settings using the pylon API or direct register access, you are making changes to the active set. The active set is located in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off.

The Default Set

When a camera is manufactured, numerous tests are performed on the camera and four factory setups optimized for different use cases are determined. The four factory setups are:

- **Standard Factory Setup:** This 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.
- **High Gain Factory Setup:** This is similar to the standard factory setup, but the gain is set to + 6 dB.
- **Auto Functions Factory Setup:** This 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.
- **Color Factory Setup:** This is optimized to yield the best color fidelity with daylight lighting.

The factory setups are saved in permanent files in the camera's non-volatile memory. They are not lost when the camera is reset or switched off and they can't be changed.

You can select one of the four factory setups to be the camera's "default set". Instructions for selecting which factory setup will be used as the default set appear later in the "Configuration Sets" section. Note that your selection of which factory setup will serve as the default set will not be lost when the camera is reset or switched off.

When the camera is running, the default set can be loaded into the active set. The default set can also be designated as the "startup" set, i.e., the set that will be loaded into the active set whenever the camera is powered on or reset. Instructions for loading the default set into the active set and for designating which set will be the startup set appear later in this section.

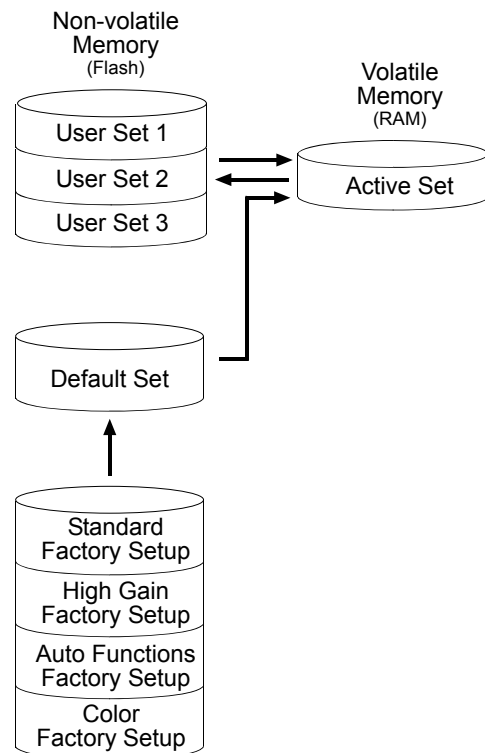


Fig. 73: Configuration Sets

User Sets

As mentioned above, the active configuration set is stored in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The camera can save most of the settings from the current active set to a reserved area in the camera's non-volatile memory. A configuration set that has been saved in the non-volatile memory is not lost when the camera is reset or switched off. There are three reserved areas in the camera's non-volatile memory available for saving configuration sets. A configuration set saved in a reserved area is commonly referred to as a "user set".

The three available user sets are called User Set 1, User Set 2, and User Set 3.

When the camera is running, a saved user set can be loaded into the active set. A saved user set can also be designated as the "startup" set, i.e., the set that will be loaded into the active set whenever the camera is powered on or reset. Instructions for loading a saved user set into the active set and for designating which set will be the startup set appear later in this section.



The values for the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If you are using the lookup table feature, you must reenter the lookup table values after each camera startup or reset.

Designating a Startup Set

You can designate the default set or one of the user sets as the "startup" set. The designated startup set will automatically be loaded into the active set whenever the camera starts up at power on or after a reset. Instructions for designating the startup set can be found in the following section.

10.17.1 Selecting a Factory Setup as the Default Set

When the camera is delivered, the Auto Functions Factory Setup will be selected as the default set. You can, however, select any one of the three factory setups to serve as the default set.



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.

Selecting a Factory Setup Using pylon

To select a factory setup as the default set:

1. 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);
```

If you want to select the Color Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_Color);
```

Selecting a Factory Setup Using Direct Register Access

To select a factory setup as the default set:

1. Set the Default Set Selector register to the Standard Factory Setup, High Gain Factory Setup, Auto Functions Factory Setup, or Color setup.

10.17.2 Saving User Sets

You can save the current parameter set being used by the camera (i.e., the "active" set in the camera's volatile memory) to user set 1, user set 2, or user set 3. The user sets are stored in the camera's non-volatile memory and will be retained when the camera power is switched off or the camera is reset. When you save the active set to a user set, any parameter data already in that user set will be overwritten.

Saving User Sets Using Basler pylon

Using Basler pylon to save the current active set to a user set in the camera's non-volatile memory involves several steps.

To save the current active set to a user set:

1. Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
2. Set the User Set Selector to User Set 1, User Set 2, or User Set 3 as desired.
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 use the pylon API to set the User Set Selector and to execute the User Set Save command from within your application software. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue(UserSetSelector_UserSet1);  
Camera.UserSetSave.Execute();
```

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

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Saving User Sets Using Direct Register Access

To save the current active set to a user set:

1. Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
2. Set the value of the User Set Selector register to User Set 1, 2, or 3 as desired.
3. Set the value of the User Set Save register to 1.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.17.3 Loading a Saved User Set or the Default Set into the Active Set

If you have saved a configuration set into one of the user sets in the camera's non-volatile memory, you can load the saved user set into the camera's active set. When you do this, the parameters stored in the user set overwrite 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 user set will now be controlling the camera.

You can also load the default set into the camera's active set.



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 an image.

Assuming that you have selected the standard factory setup as the default set, 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.

Loading a Set Using Basler pylon

Loading a saved user set or the default set from the camera's non-volatile memory into the active set using Basler pylon involves two steps.

To load a saved user set into the active set:

1. Set the User Set Selector to User Set 1, User Set 2, User Set 3, or Default as desired.
2. Execute a User Set Load command to load the selected set into the active set.

You can use the pylon API to set the User Set Selector and to execute the User Set Load command from within your application software. The following code snippet illustrates using the API to set the selector and execute the command:

```
// Load user set 2 into the active set
Camera.UserSetSelector.SetValue(UserSetSelector_UserSet2);
Camera.UserSetLoad.Execute();

// Load the default set into the active set
Camera.UserSetSelector.SetValue(UserSetSelector_Default);
Camera.UserSetLoad.Execute();
```

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Loading a Set Using Direct Register Access

To load a saved user set into the active set:

1. Set the value of the User Set Selector register to User Set 1, 2, or 3, or to the Default set as desired.
2. Set the value of the User Set Load register to 1.

For more information about direct register access, see Section 3.2 on [page 20](#).

10.17.4 Selecting a "Startup" Set

You can select the default set or one of the user sets stored in the camera's non-volatile memory to be the "startup" set. The configuration set that you select as the startup set will be loaded into the active set whenever the camera starts up at power on or after a reset.

Selecting the Startup Set Using Basler pylon

With Basler pylon, the User Set Default Selector parameter is used to select User Set 1, User Set 2, User Set 3, or the Default Set as the startup set.

You can use the pylon API to set the User Set Default Selector parameter from within your application software. The following code snippet illustrates using the API to set the selector:

```
// Designate user set 1 as the startup set
Camera.UserSetDefaultSelector.SetValue(UserSetDefaultSelector_UserSet1);

// Designate the default set as the startup set
Camera.UserSetDefaultSelector.SetValue(UserSetDefaultSelector_Default);
```

For more information about the pylon API and the pylon Viewer, see Section 3.1 on [page 18](#).

Selecting the Startup Set Using Direct Register Access

When using direct register access, the User Set Default Selector register is used to select the startup set.

To select a startup set:

1. Set the value of the User Set Default Selector register for User Set 1, User Set 2, User Set 3, or Default as desired.

For more information about direct register access, see Section 3.2 on [page 20](#).

Revision History

Doc. ID Number	Date	Changes
AW00098501000	16 Feb 2011	Preliminary release of this document. Applies to prototype cameras only.
AW00098502000	1 Apr 2011	<p>Second preliminary release.</p> <p>Added information for the acA2000-340km/kc, acA2040-70km/kc, and acA2040-180km/kc models.</p> <p>Added information for the new stacked zone imaging, sequencer, and error detection features.</p> <p>Applies to prototype cameras only.</p>
AW00098503000	6 Jun 2011	<p>Third preliminary release. Applies to prototype cameras only.</p> <p>Updated all specifications and feature descriptions.</p>
AW00098504000	7 Feb 2013	<p>Fourth release: Information relating to acA NIR cameras describes the cameras in their prototype stage.</p> <p>Indicated Basler AG as bearer of the copyright on the back of the front page.</p> <p>Updated e-mail addresses for Basler technical support.</p> <p>Removed the following cameras from the manual: acA2000-140km/kc, acA2040-70km/kc.</p> <p>Included the following cameras into the manual: acA2000-340kmNIR, acA2040-180kmNIR.</p> <p>Corrected the focal length distance in Section 1.4.1 on page 8.</p> <p>Added LZ4 licensing information in Section 1.5.1 on page 10.</p> <p>Modified the statement of camera power in Table 4 on page 25 and replaced it in Table 5 on page 26.</p> <p>Added the full config. entries in Table 5.2.2 on page 24.</p> <p>Removed absolute max. voltages from Table 6 on page 35, Table 7 on page 36, and Section 5.6.2.2 on page 35 (inputs) and from Table 6 on page 37 and Section 5.6.4.2 on page 37 (output).</p> <p>Replaced absolute max. input voltage in Figure 10 on page 35 by the recommended voltage range for standard power and I/O cable.</p>

Doc. ID Number	Date	Changes
AW00098505000	24 Mar 2015	<p>Minor corrections throughout the manual.</p> <p>Removed the note about NIR prototype cameras from the front page.</p> <p>Corrected subpart J to subpart B in the FCC section at the back of the front page.</p> <p>Updated support contact addresses and phone numbers.</p> <p>Added M5 receptacle in Camera Link Connectors section in Table 1 on page 2 and Table 2 on page 3.</p> <p>Added input and output lines in I/O Lines section in Table 1 on page 2 and Table 2 on page 3.</p> <p>Added information about SELV and LPS in Table 1 on page 2, Table 2 on page 3, on page 13, page 14, page 29, page 31, and page 33.</p> <p>Updated LZ4 licensing information in Section 1.5 on page 10.</p> <p>Added warnings about avoiding dust on the sensor in Figure 1.8 on page 13.</p> <p>Updated Chapter 3 on page 18 to reflect software update to Basler pylon Camera Software Suite.</p> <p>Added connector type information (previously Section 5.3) to Section 5.2 on page 24.</p> <p>Re-organized information in Section 5.3 on page 27.</p> <p>Updated information about Camera Link cables in Section 5.3.1 on page 27.</p> <p>Updated information about using the GPIO line as an input line in Section 5.6.2 on page 34.</p> <p>Updated Figure 9 on page 35 and Figure 10 on page 35.</p> <p>Deleted the "Schematic with GPIO Set as an Output" graphic.</p> <p>Updated information about using the GPIO line as an output line in Section 5.6.4 on page 37.</p> <p>Updated Figure 12 on page 38 and Figure 13 on page 38.</p> <p>Added a note about negative line logic of CC lines in Section 5.7.1 on page 40 and Section 6.1.1 on page 42.</p> <p>Added two standard output signals in Section 5.7.2 on page 41 and Section 6.2.3 on page 49.</p> <p>Added use case information for working with user settable output lines in Section 6.2.4 on page 51.</p> <p>Added Section 6.2.5 on page 52, Section 6.2.6 on page 53, and Section 6.2.7 on page 55.</p> <p>Rephrased information about applying trigger signals in Section 7.1 on page 62.</p> <p>Added note about trigger limitations in Section 7.3.1.2 on page 75.</p> <p>Added a note about using CC lines as a hardware trigger source in Section 7.2.1.2 on page 66, Section 7.3.1.2 on page 75, and Section 7.3.3.4 on page 86.</p> <p>(continued on next page)</p>

Doc. ID Number	Date	Changes
		<p>(continued from previous page)</p> <p>Corrected the maximum value of the acquisition start trigger delay in Section 7.2.5.2 on page 72.</p> <p>Corrected the maximum value of the frame start trigger delay in Section 7.3.3.3 on page 86.</p> <p>Added information about exposure time offsets in CMOSIS V2 sensors under "Exposure Time Offset in CMOSIS V2 Sensors" on page 89.</p> <p>Added information about using hexadecimal figures when setting gamma correction in Section 8.3.2 on page 131.</p> <p>Added Section 10.3 on page 155.</p> <p>Updated the sequencer feature information in Section 10.6 on page 165.</p> <p>Updated the binning feature information in Section 10.7.2 on page 203.</p> <p>Added Section 10.8 on page 206.</p> <p>Updated the Balance White Auto function information in Section 10.11.7 on page 236.</p> <p>Added Section 10.12 on page 238.</p>

Index

A

acquisition frame count parameter68, 70
 acquisition start trigger
 details66
 introduction.....62
 acquisition start trigger mode parameter..66
 acquisition status indicator98
 acquisition status parameter98, 99
 acquisition trigger wait signal100
 active configuration set.....249
 active set165
 see active configuration set
 advance
 asynchronous168
 AOI, see area of interest156
 API.....19
 area of interest
 auto functions AOI.....219, 221
 image204
 image AOI156
 asynchronous advance168
 asynchronous restart.....168
 auto function profile235
 auto functions
 area of interest221
 assignment to an auto function
 AOI222
 introduction.....219
 modes of operation220
 using with binning.....219
 auto functions AOI
 explained.....219
 relating to an auto function222
 setting.....226
 auto sequence set advance mode170

B

balance white auto236
 Basler binary protocol library20
 Bayer filter127
 binary protocol library20

binning201
 AOI settings.....204
 reduced resolution.....203
 stacked zone imaging205
 bit depth.....2, 3, 137
 black level.....154

C

cables
 PLC power and I/O cable28
 standard I/O cable27
 camera link pixel clock speed, setting150
 camera power.....30, 31
 CC1, CC2, and CC3.....40, 42
 CL spare.....41, 48
 cleaning the camera and sensor16
 clock speed, camera link pixel clock150
 code snippets, proper use15
 color factory setup249
 color filter127
 color transformation mode parameter134
 configuration set loaded at startup254
 configuration sets249–254
 conformity2, 4
 connector types24
 connectors23
 controlled sequence set advance
 mode175
 current set165

D

debouncer
 and exposure start delay113
 explained44
 decimation
 AOI settings.....208
 image distortion207
 reduced resolution.....207
 setting.....207
 vertical206
 default configuration set249
 device ID parameter246
 device manufacturer info parameter.....246
 device model name parameter246
 device scan type parameter246
 device user ID parameter246

device vendor name parameter
 parameters
 device vendor name 246
 device version parameter 246
 dimensions 2, 3
 dust 15

E

electromagnetic interference 11
 electrostatic discharge 11
 EMI 11
 environmental requirements 12
 error detection 240
 ESD 11
 exposure active signal 108
 exposure auto 232
 exposure modes
 timed 83
 trigger width 85
 exposure overlap time max abs
 parameter 85, 105
 exposure start delay 113
 exposure time
 controlling with an external trigger
 signal 82
 offset 89
 setting 89
 exposure time offset 89

F

factory setup
 color factory setup 249
 high gain factory setup 249
 standard factory setup 249
 firmware version parameter 246
 frame cycle signal 111
 frame rate
 and AOI size 115
 controlling with an external trigger
 signal 82
 increasing the max allowed 117
 maximum allowed 115
 frame readout time 113
 frame start trigger 167
 details 74
 introduction 63
 frame start trigger delay 86

frame start trigger mode parameter 74
 frame transmission time 113
 frame trigger wait signal 103
 free run 119
 free selection sequence set advance
 mode 194
 functional description 21

G

gain auto 229
 gamma correction 132

H

hardware trigger
 acquisition start 71
 frame start 82
 heat dissipation 12
 high gain factory setup 249
 horizontal binning 201
 humidity 12

I

image area of interest 156
 image distortion 207
 increasing the max allowed frame rate .. 117
 input line
 electrical characteristics 35
 invert 47
 propagation delay 36
 using as a source signal 44
 voltage requirements 34
 integrate enabled signal 108, 111
 invert
 input line 47
 output line 57
 IP30 8
 IR-cut filter 7, 9, 129

L

LED indicator	32, 240
lens adapter	2, 3
lens thread length	9
line inverter parameter	47, 58
line source	
parameter	50, 102, 106, 109, 112
line status	180
line status parameter	59, 61
LUT (luminance lookup table)	214

M

max height parameter	246
max lens thread length	9
max width parameter	246
maximum allowed frame rate	115
minimum output pulse width	238
mirror imaging	209
models	1
modes of operation (of auto functions)	220

N

non-sequence parameter	165
------------------------------	-----

O

optical size, sensor	2, 3
output line	
electrical characteristics	37
invert	57
minimum output pulse width	238
response time	39
selecting a source signal	49
voltage requirements	37
overlapped exposure	95
overtrigger error	98, 100
overtriggering	83

P

parameter limits, removing	155
parameter sets	
explained	249
saving	252

parameters

acquisition frame count	68, 70
acquisition start trigger mode	66
acquisition status	98, 99
color transformation mode	134
device ID	246
device manufacturer info	246
device model name	246
device scan type	246
device user ID	246
device version	246
exposure overlap time max	
abs	85, 105
firmware version	246
frame start trigger mode	74
line inverter	47, 58
line source	102, 106, 109, 112
line status	59, 61
max height	246
max width	246
readout time abs	114
remove limits	155
sensor height	246
sensor width	246
setting via register access	20
sync user output value	54
parameters loaded at startup	254
pin assignments	24
pin numbering	24
pixel clock speed	150
pixel size	2, 3
PLC power and I/O cable	27
voltage requirements	34
PoCL	30
power over Camera Link	30
power requirements, camera	30, 31
power supply	
SELV	13, 17, 29, 33
powered hub	
SELV	13, 17, 29, 33
precautions	13
protection class	8
pylon API	19
pylon SDK	18, 19
pylon Viewer	18

R

readout time abs parameter	114
reduced resolution	203, 207
registers	20
remove limits parameter	155
removing parameter limits	155
restart	
asynchronous	168
synchronous	168
reverse X	209
reverse Y	211

S

saving parameter sets	249, 252
sensor	
optical size	2, 3
pixel size	2, 3
size in pixels	2, 3
type	2, 3
sensor bit depth	137
sensor height parameter	246
sensor width parameter	246
sequence advance mode	165
sequence parameter	165
sequence set	165
address	194, 197
configuration	173, 190, 198
cycle	168, 170
index number	167
load	169
store	173, 191, 198
sequence set advance mode	
auto	170
free selection	194
sequence set index number	167
sequencer	
standard operation	168
serial number	3
sets of parameters, saving	252
setting parameters	
via register access	20
size, camera	2, 3
software development kit	19
software trigger	
acquisition start	70
frame start	79

spectral response

color cameras	7
mono cameras	5
sRGB gamma	131
stacked zone imaging	160
standard factory setup	249
standard power and I/O cable	27
voltage requirements	34
startup parameter set	250, 254
sync user output selector	54
sync user output value parameter	54
synchronous advance	168
synchronous restart	168

T

tap geometries	141
1X10-1Y	148
1X2-1Y	143
1X3-1Y	144
1X4-1Y	145
1X6-1Y	146
1X8-1Y	147
target value	219
temperature, housing	12
test images	243
timed exposure mode	83
transition threshold	34
transmission start delay	113
trigger delay	
acquisition start	72
frame start	86
trigger wait signals	100
trigger width exposure mode	85
troubleshooting with the LED indicator ..	240

U

use case diagrams	119, 170
user configuration set	250
user output selector	51
user output value parameter	
parameters	
user output value	51
user-defined values	248

V

ventilation	12
vertical binning	201
vertical decimation.....	206

W

weight	2, 4
white balance	130
white balance auto see balance white auto	

Z

zone imaging	160
--------------------	-----