

## Basler scout light



### **USER'S MANUAL**

**(for scout light Cameras Used with Basler's Pylon API)**

Document Number: AW000753

Version: 02 Language: 000 (English)

Release Date: 17 June 2009

**BASLER** 

**For customers in the U.S.A.**

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

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

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

**For customers in Canada**

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

**Pour utilisateurs au Canada**

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

**Life Support Applications**

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

**Warranty Note**

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

**All material in this publication is subject to change without notice and is copyright Basler Vision Technologies.**

## **Contacting Basler Support Worldwide**

### **Europe:**

Basler AG  
An der Strusbek 60 - 62  
22926 Ahrensburg  
Germany

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

[bc.support.europe@baslerweb.com](mailto:bc.support.europe@baslerweb.com)

### **Americas:**

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

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

[bc.support.usa@baslerweb.com](mailto:bc.support.usa@baslerweb.com)

### **Asia:**

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

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

[bc.support.asia@baslerweb.com](mailto:bc.support.asia@baslerweb.com)

**[www.baslerweb.com](http://www.baslerweb.com)**



---

## Table of Contents

<b>1</b>	<b>Specifications, Requirements, and Precautions</b>	<b>1</b>
1.1	Models	1
1.2	General Specifications	2
1.3	Spectral Response for Mono Cameras	4
1.4	Mechanical Specifications	7
1.4.1	Camera Dimensions and Mounting Points	7
1.4.2	Mechanical Stress Test Results	9
1.5	Software Licensing Information	10
1.6	Avoiding EMI and ESD Problems	11
1.7	Environmental Requirements	12
1.7.1	Temperature and Humidity	12
1.7.2	Ventilation	12
1.8	Precautions	13
<b>2</b>	<b>Software and Hardware Installation</b>	<b>15</b>
<b>3</b>	<b>Tools for Changing Camera Parameters</b>	<b>17</b>
3.1	The pylon Viewer	17
3.2	The pylon API	17
<b>4</b>	<b>Functional Description</b>	<b>19</b>
4.1	Overview (All Models Except slA750-60fm)	19
4.2	Overview (slA750-60fm Only)	21
<b>5</b>	<b>Physical Interface</b>	<b>23</b>
5.1	General Description of the Connections	23
5.2	Connector Pin Assignments and Numbering	24
5.2.1	IEEE 1394b Socket Pin Assignments	24
5.2.2	12-pin Receptacle Pin Assignments	25
5.2.3	Pin Numbering	26
5.3	Connector Types	27
5.3.1	IEEE 1394b Connector	27
5.3.2	12-pin Connector	27
5.4	Cabling Requirements	28
5.4.1	IEEE 1394b Cable	28
5.4.2	I/O Cable	28
5.5	IEEE 1394b Device Information	30
5.6	Camera Power	31

---

5.7	Input and Output Lines . . . . .	32
5.7.1	I/O Schematic . . . . .	32
5.7.2	Input Line Description . . . . .	32
5.7.2.1	Voltage Requirements. . . . .	32
5.7.2.2	Input Line Schematic. . . . .	33
5.7.3	Output Line Description . . . . .	34
5.7.3.1	Voltage Requirements. . . . .	34
5.7.3.2	Output Line Schematic . . . . .	34
<b>6</b>	<b>Image Acquisition Control . . . . .</b>	<b>37</b>
6.1	Controlling Image Acquisition with Parameters Only (No Triggering) . . . . .	37
6.1.1	Switching Off Triggering. . . . .	37
6.1.2	Acquiring One Image at a Time . . . . .	38
6.1.3	Acquiring Images Continuously (Free-run) . . . . .	38
6.2	Controlling Image Acquisition with a Software Trigger . . . . .	40
6.2.1	Enabling the Software Trigger Feature . . . . .	40
6.2.2	Acquiring a Single Image by Applying One Software Trigger . . . . .	41
6.2.3	Acquiring Images by Applying a Series of Software Triggers . . . . .	42
6.3	Controlling Image Acquisition with a Hardware Trigger . . . . .	44
6.3.1	Exposure Modes . . . . .	45
6.3.2	Setting the Camera for Hardware Triggering . . . . .	47
6.3.3	Acquiring a Single Image by Applying One Hardware Trigger Transition . . . . .	48
6.3.4	Acquiring Images by Applying a Series of Hardware Trigger Transitions . . . . .	49
6.4	Exposure Time Parameters . . . . .	51
6.4.1	Setting the Exposure Time Using "Raw" Settings . . . . .	51
6.4.2	Setting the Exposure Time Using "Absolute" Settings. . . . .	53
6.5	Overlapping Exposure and Sensor Readout (All Models Except sIA750-60fm) . . . . .	54
6.5.1	Guidelines for Overlapped Operation . . . . .	55
6.6	Exposure Must Not Overlap Sensor Readout (sIA750-60fm Only) . . . . .	56
6.7	Trigger Ready Signal . . . . .	57
6.7.1	Trigger Ready Signal (All Models Except sIA750-60fm) . . . . .	57
6.7.2	Trigger Ready Signal (sIA750-60fm Only). . . . .	59
6.8	Exposure Active Signal. . . . .	61
6.9	Acquisition Timing Chart. . . . .	62
6.10	Maximum Allowed Acquisition Frame Rate (All Models Except sIA750-60fm). . . . .	65
6.10.1	Effect of the Packet Size Setting on the Maximum Allowed Frame Rate . . . . .	68
6.11	Maximum Allowed Acquisition Frame Rate (sIA750-60fm Only) . . . . .	70
6.11.1	Effect of the Packet Size Setting on the Maximum Allowed Frame Rate . . . . .	73

---

<b>7</b>	<b>Pixel Data Formats</b>	<b>75</b>
7.1	Setting the Pixel Data Format	75
7.2	Pixel Data Formats	76
7.2.1	Mono 8 Format (Equivalent to DCAM Mono 8)	76
7.2.2	Mono 16 Format (Equivalent to DCAM Mono 16)	78
7.2.3	Mono 12 Packed Format	80
7.2.4	YUV 4:2:2 Packed Format (Equivalent to DCAM YUV 4:2:2)	82
7.2.5	YUV 4:2:2 (YUYV) Packed Format	83
7.3	Pixel Transmission Sequence	86
<b>8</b>	<b>I/O Control</b>	<b>87</b>
8.1	Configuring the Input Line	87
8.1.1	Assigning the Input Line to Receive a Hardware Trigger Signal	87
8.2	Configuring the Output Line	88
8.2.1	Assigning a Camera Output Signal to the Physical Output Line	88
8.2.2	Setting the State of a User Settable Output Line	89
8.2.3	Setting the Output Line for Invert	89
8.2.4	Working with the Timer Signal	91
8.2.4.1	Setting the Trigger Source for the Timer	91
8.2.4.2	Setting the Timer Delay Time	92
8.2.4.3	Setting the Timer Duration Time	94
8.3	Checking the State of the I/O Lines	96
8.3.1	Checking the State of the Output Line	96
8.3.2	Checking the State of All Lines	96
<b>9</b>	<b>Standard Features</b>	<b>99</b>
9.1	Gain	99
9.2	Black Level	104
9.3	Digital Shift	105
9.3.1	Digital Shift with 12 Bit Pixel Formats	105
9.3.2	Digital Shift with 8 Bit Pixel Formats	107
9.3.3	Precautions When Using Digital Shift	109
9.3.4	Enabling and Setting Digital Shift	109
9.4	Area of Interest (AOI)	111
9.4.1	Changing AOI Parameters "On-the-Fly"	114
9.5	Reverse X	115
9.6	Disable Parameter Limits	117
9.7	Debouncer	118
9.8	Trigger Delay	120
9.9	Acquisition Status	121
9.10	Test Images	122
9.11	Device Information Parameters	125

9.12	Configuration Sets	127
9.12.1	Saving User Sets	128
9.12.2	Selecting a Factory Setup as the Default Set	129
9.12.3	Loading a Saved Set or the Default Set into the Active Set.	130
9.12.4	Selecting the Startup Set	131
<b>10</b>	<b>Using Multiple Cameras on a Single Bus and Managing Bandwidth</b>	<b>133</b>
10.1	Using Multiple Cameras Where All Devices are 1394b	133
10.2	Using Multiple Cameras Where 1394a and 1394b Devices are Mixed	135
10.2.1	Recommended Packet Size	138
<b>11</b>	<b>Troubleshooting and Support</b>	<b>139</b>
11.1	Tech Support Resources	139
11.2	Obtaining an RMA Number	139
11.3	Troubleshooting with the Camera LED	140
11.4	Troubleshooting Charts	141
11.4.1	My Camera Is Not Being Recognized	142
11.4.2	I Do Not Get an Image	143
11.4.3	I Can Not Get the Full Frame Rate	144
11.4.4	I Get Poor Image Quality	146
11.5	Before Contacting Basler Technical Support	148
	<b>Revision History</b>	<b>151</b>
	<b>Feedback</b>	<b>153</b>
	<b>Index</b>	<b>155</b>



# 1 Specifications, Requirements, and Precautions

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

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

## 1.1 Models

The current Basler scout light camera models are listed in the top row of the specification tables on the next pages of this manual. The camera models are differentiated by their sensor size and their maximum frame rate at full resolution.

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

## 1.2 General Specifications

Specification	sIA750-60fm	sIA1000-30fm
Sensor Size (H x V pixels)	752 x 480	1034 x 779
Sensor Type	Aptina MT9V022 (formerly known as the Micron MT9V022) Progressive Scan CMOS	Sony ICX204 AL Progressive Scan CCD
Optical Size	1/3"	1/3"
Pixel Size	6.0 $\mu\text{m}$ x 6.0 $\mu\text{m}$	4.65 $\mu\text{m}$ x 4.65 $\mu\text{m}$
Max. Frame Rate (at full resolution)	64.9 fps	30 fps
Mono/Color	Mono	
Data Output Type	IEEE 1394b	
Pixel Data Formats	Mono 8 (= DCAM Mono 8) YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed	Mono 8 (= DCAM Mono 8) Mono 16 (= DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	10 bits	12 bits
Synchronization	Via external trigger signal or via software	
Exposure Control	Programmable via the camera API	
Camera Power Requirements	+8 to +36 VDC supplied via the IEEE 1394 cable, < 1% ripple	
	1.7 W @ 12 V	2.5 W @ 12 V
I/O Lines	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount	
Size (L x W x H)	73.7 mm x 44 mm x 29 mm (without lens adapter or connectors)	
	85.5 mm x 44 mm x 29 mm (with lens adapter and connectors)	
Weight	160 g (typical)	
Conformity	CE, FCC, GenICam, IP30	

Table 1: General Specifications

Specification	sIA1390-17fm	sIA1600-14fm
Sensor Size (H x V pixels)	1392 x 1040	1626 x 1236
Sensor Type	Sony ICX267 AL Progressive Scan CCD	Sony ICX274 AL Progressive Scan CCD
Optical Size	1/2"	1/1.8"
Pixel Size	4.65 $\mu\text{m}$ x 4.65 $\mu\text{m}$	4.4 $\mu\text{m}$ x 4.4 $\mu\text{m}$
Max. Frame Rate (at full resolution)	17 fps	14 fps
Mono/Color	Mono	
Data Output Type	IEEE 1394b	
Pixel Data Formats	Mono 8 (= DCAM Mono 8) Mono 16 (= DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed	
ADC Bit Depth	12 bits	
Synchronization	Via external trigger signal or via software	
Exposure Control	Programmable via the camera API	
Camera Power Requirements	+8 to +36 VDC supplied via the IEEE 1394 cable, < 1% ripple	
	2.75 W @ 12 V	2.75 W @ 12 V
I/O Ports	1 opto-isolated input line and 1 opto-isolated output line	
Lens Adapter	C-mount	
Size (L x W x H)	73.7 mm x 44 mm x 29 mm (without lens adapter or connectors)	
	85.5 mm x 44 mm x 29 mm (with lens adapter and connectors)	
Weight	160 g (typical)	
Conformity	CE, FCC, GenICam, IP30	

Table 2: General Specifications

## 1.3 Spectral Response for Mono Cameras

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



**Note**

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

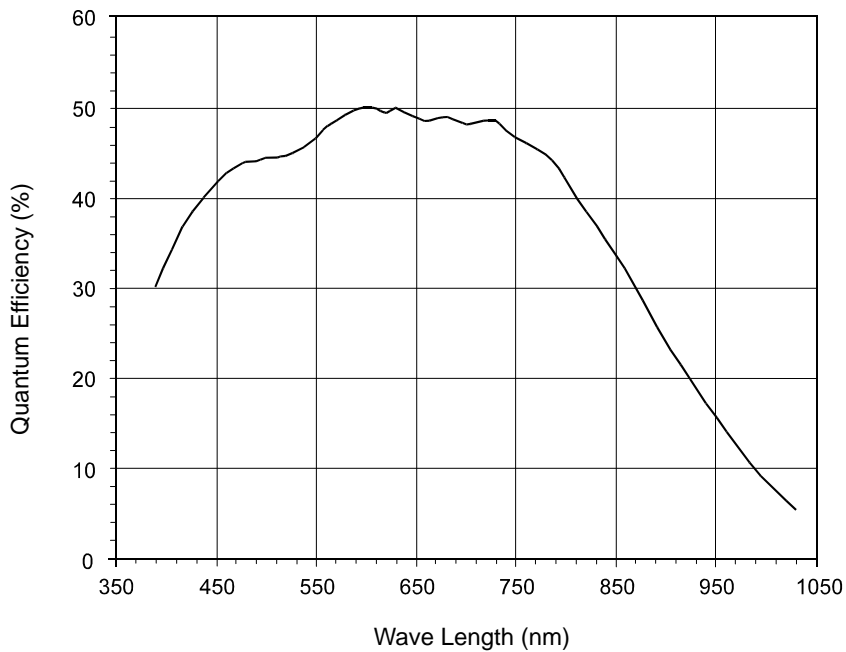


Fig. 1: sIA750-60fm Spectral Response

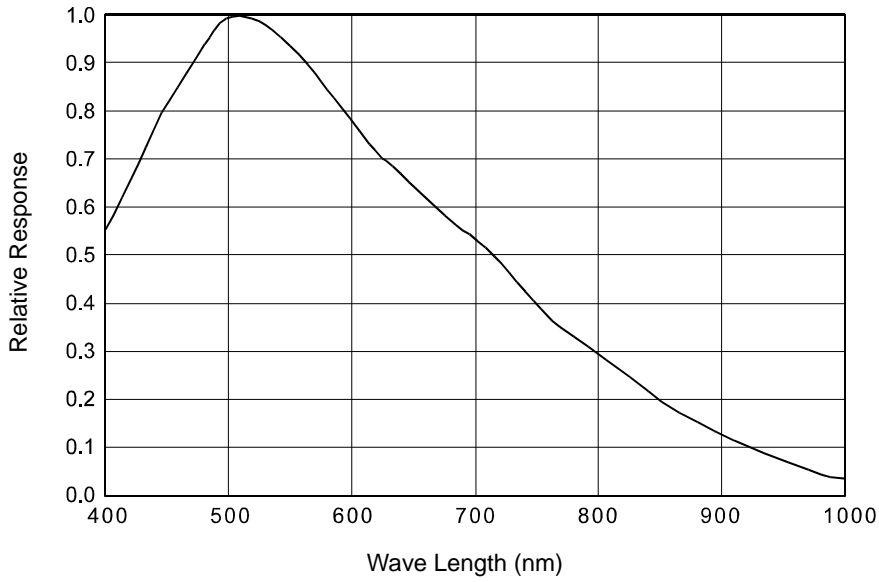


Fig. 2: slA1000-30fm Spectral Response

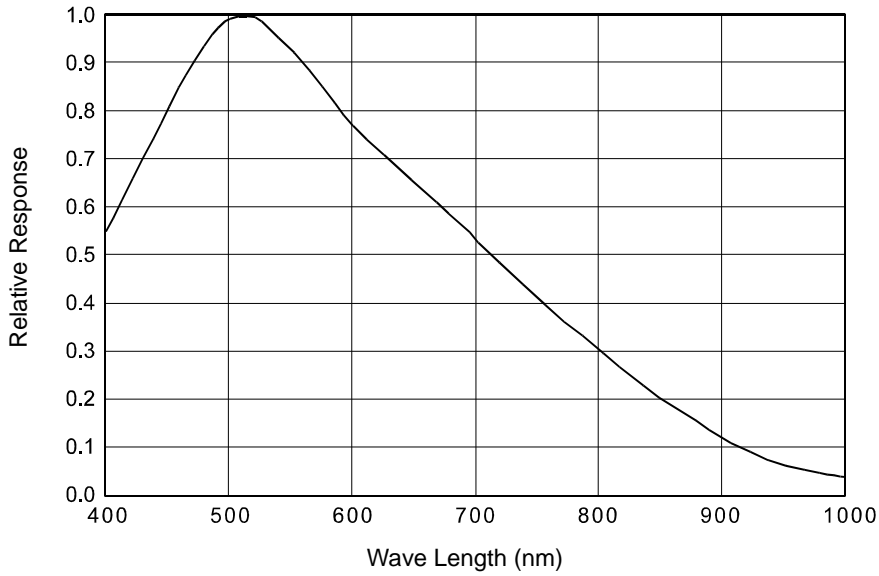


Fig. 3: slA1390-17fm Spectral Response

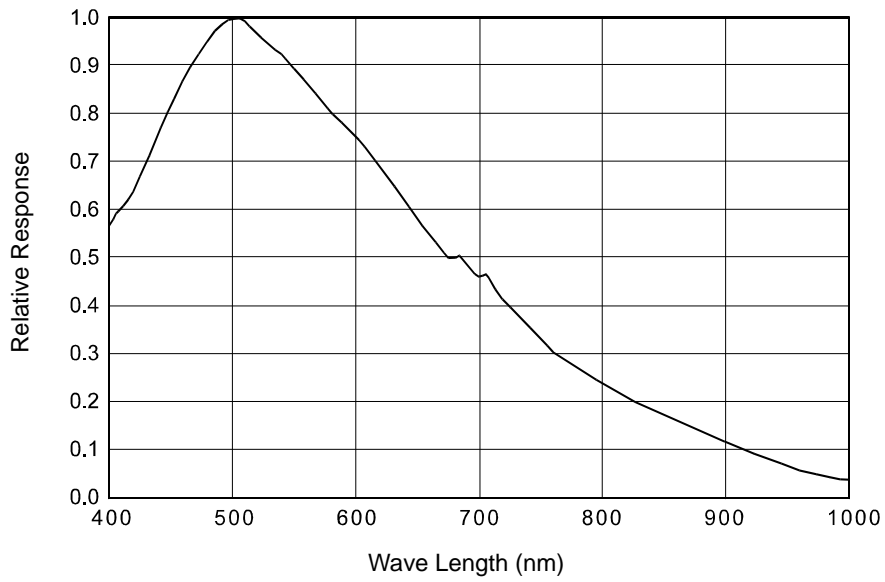


Fig. 4: sIA1600-14fm Spectral Response

## 1.4 Mechanical Specifications

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

### 1.4.1 Camera Dimensions and Mounting Points

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

The camera dimensions in millimeters are as shown in Figure 5.

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

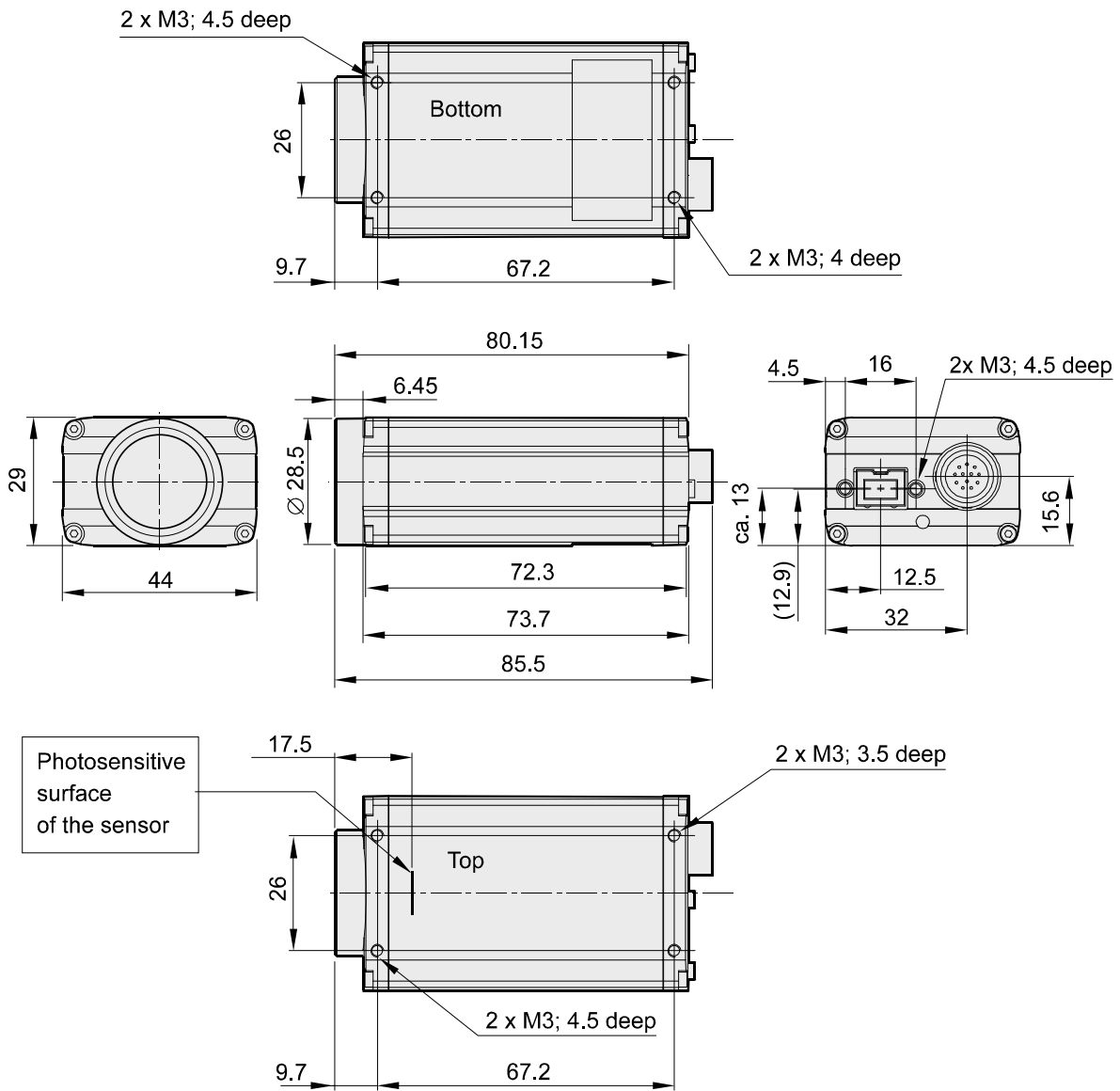


Fig. 5: Mechanical Dimensions (in mm) for Cameras with the Standard C-mount Lens Adapter



## 1.4.2 Mechanical Stress Test Results

Scout cameras were submitted to an independent mechanical testing laboratory and subjected to the stress tests listed below. The mechanical stress tests were performed on selected camera models with standard housings. After mechanical testing, the cameras exhibited no detectable physical damage and produced normal images during standard operational testing.

Test	Standard	Conditions
Vibration (sinusoidal, each axis)	DIN EN 60068-2-6	10-58 Hz / 1.5 mm_58-500 Hz / 20 g_1 Octave/Minute 10 repetitions
Shock (each axis)	DIN EN 60068-2-27	20 g / 11 ms / 10 shocks positive 20 g / 11 ms / 10 shocks negative
Bump (each axis)	DIN EN 60068-2-29	20 g / 11 ms / 100 shocks positive 20 g / 11 ms / 100 shocks negative
Vibration (broad-band random, digital control, each axis)	DIN EN 60068-2-64	15-500 Hz / 0.05 PSD (ESS standard profile) / 00:30 h

Table 3: Mechanical Stress Tests

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

## 1.5 Software Licensing Information

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

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

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

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

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

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

## 1.6 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.



### Note

The Basler application note called *Avoiding EMI and ESD in Basler Camera Installations* provides much more detail about avoiding EMI and ESD.

The application note can be downloaded at:

[www.baslerweb.com/indizes/download\\_index\\_en\\_31412.html](http://www.baslerweb.com/indizes/download_index_en_31412.html)

## 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 Ventilation

Allow sufficient air circulation around the camera to prevent internal heat build-up in your system and to keep the camera's housing temperature below 50 °C. Additional cooling devices such as fans or heat sinks are not normally required, but should be provided if necessary.

## 1.8 Precautions



### CAUTION

#### Avoid Dust on the Sensor

The camera is shipped with a cap on the lens mount. To avoid collecting dust on the camera's sensor, make sure that you always put the cap in place when there is no lens mounted on the camera.

To further enhance dust protection, the internal space in the camera that contains the imaging sensor is sealed off from the camera's other internal spaces.



### CAUTION

#### Incorrect Power Can Cause Damage

The polarity of the power on the camera's IEEE 1394b socket must be as shown in the pin assignment table. **Do not** reverse the power polarity. Reversing the polarity will damage the camera.

If the voltage to the camera is greater than +36 VDC, damage to the camera can result. If the voltage is less than +8 VDC, the camera may operate erratically.



### CAUTION

#### Inappropriate Code May Cause Unexpected Camera Behavior

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

To ensure that the snippets will work properly in your application, you must adjust them to meet your specific needs and must test them thoroughly prior to use.

## **Warranty Precautions**

**To ensure that your warranty remains in force:**

### **Do not remove the camera's serial number label**

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

### **Do not open the camera housing**

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

### **Keep foreign matter outside of the camera**

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

### **Avoid Electromagnetic fields**

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

### **Transport Properly**

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

### **Clean Properly**

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

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

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

### **Read the manual**

Read the manual carefully before using the camera!

## 2 Software and Hardware Installation

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

You can download the guide from the Basler website:

[www.baslerweb.com/indizes/download\\_index\\_en\\_19627.html](http://www.baslerweb.com/indizes/download_index_en_19627.html).

The guide includes information about both hardware and software and describes how to begin capturing images.





# 3 Tools for Changing Camera Parameters

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

## 3.1 The pylon Viewer

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

The pylon Viewer is included in Basler's pylon Driver Package. You can download the pylon package from the Basler website: [www.baslerweb.com/beitraege/beitrag\\_en\\_71708.html](http://www.baslerweb.com/beitraege/beitrag_en_71708.html).

For more information about using the viewer, see the Installation and Setup Guide for Cameras Used with Basler's pylon API, (AW000611xx000). You can download the guide from the Basler website: [www.baslerweb.com/indizes/download\\_index\\_en\\_19627.html](http://www.baslerweb.com/indizes/download_index_en_19627.html).

## 3.2 The pylon API

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

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

The SDK is included in Basler's pylon Driver Package. You can download the pylon package from the Basler website: [www.baslerweb.com/beitraege/beitrag\\_en\\_71708.html](http://www.baslerweb.com/beitraege/beitrag_en_71708.html).



# 4 Functional Description

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

## 4.1 Overview (All Models Except slA750-60fm)



### Note

The information in this section applies to all camera models except the slA750-60fm. For information about slA750-60fm cameras, see Section 4.2 on [page 21](#).

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

Exposure start, exposure time, and charge readout can be controlled by parameters transmitted to the camera via the Basler pylon API and the IEEE 1394b interface. There are also parameters available to set the camera for single frame acquisition or continuous frame acquisition.

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

Accumulated charges are read out of the sensor when exposure ends. At readout, accumulated charges are transported from the sensor's light-sensitive elements (pixels) to the vertical shift registers (see Figure 6 on [page 20](#)). The charges from the bottom line of pixels in the array are then moved into a horizontal shift register. Next, the charges are shifted out of the horizontal register. As the charges move out of the horizontal shift register, they are converted to voltages proportional to the size of each charge. Each voltage is then amplified by a Variable Gain Control (VGC) and digitized by an Analog-to-Digital converter (ADC). After each voltage has been amplified and digitized, it passes through an FPGA and into an image buffer. All shifting is clocked according to the camera's internal data rate. Shifting continues in a linewise fashion until all image data has been read out of the sensor.

The pixel data leaves the image buffer and passes back through the FPGA to an IEEE1394b link layer controller where it is assembled into data packets. The packets are passed to a 1394b physical layer controller which transmits them isochronously to an interface board in the host PC. The physical and link layer controllers also handle transmission and receipt of asynchronous control data such as changes to the camera's parameters.

The image buffer between the sensor and the link layer controller allows data to be read out of the sensor at a rate that is independent of the data transmission rate between the camera and the host computer. This ensures that the data transmission rate has no influence on image quality.

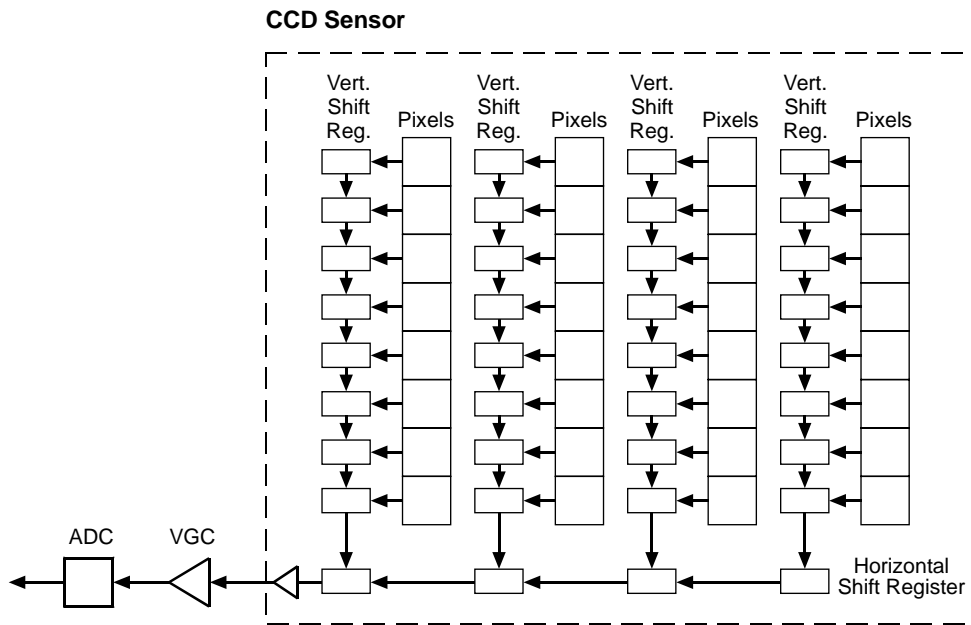


Fig. 6: CCD Sensor Architecture

## 4.2 Overview (sIA750-60fm Only)



### Note

The information in this section only applies to sIA750-60fm cameras. For information about the other camera models, see Section 4.1 on [page 19](#).

Each camera provides features such as a full frame shutter and electronic exposure time control. The sensor chip includes gain controls, ADCs, and other digital devices.

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

Exposure start can also be controlled via an externally generated hardware trigger (ExTrig) signal. The ExTrig signal facilitates periodic or non-periodic acquisition start. Exposure can be set for a preprogrammed period of time.

Accumulated charges are read out when the programmed exposure time ends. At readout, the accumulated charges are transported from the sensor's light-sensitive elements (pixels) to the sensor's column buses (see Figure 7 on [page 22](#)). The charges from the bottom line of pixels in the array are then moved into the analog processing section of the sensor. As the charges move from the pixels to the analog processing section, they are converted to voltages proportional to the size of each charge. The voltages from the analog processing section are next passed to a bank of Analog-to-Digital converters (ADCs).

Finally, the voltages pass through a section of the sensor where they receive additional digital processing and then they are moved out of the sensor. As each voltage leaves the sensor, it passes through an FPGA and into an image buffer. All shifting is clocked according to the camera's internal data rate. Shifting continues in a linewise fashion until all image data has been read out of the sensor.

The pixel data leaves the image buffer and passes back through the FPGA to an IEEE1394b link layer controller where it is assembled into data packets. The packets are passed to a 1394b physical layer controller which transmits them isochronously to an interface board in the host PC. The physical and link layer controllers also handle transmission and receipt of asynchronous control data such as changes to the camera's parameters.

The image buffer between the sensor and the link layer controller allows data to be read out of the sensor at a rate that is independent of the data transmission rate between the camera and the host computer. This ensures that the data transmission rate has no influence on image quality.

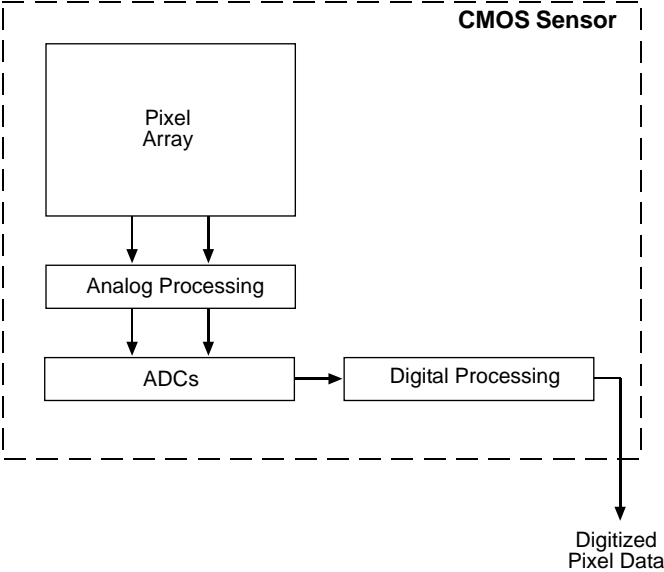


Fig. 7: CMOS Sensor Architecture

# 5 Physical Interface

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

## 5.1 General Description of the Connections

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

- an IEEE 1394b socket used to provide power and a bus connection to the camera.
- a 12-pin receptacle used to provide access to the camera's I/O ports.

There is also an LED indicator on the back.

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

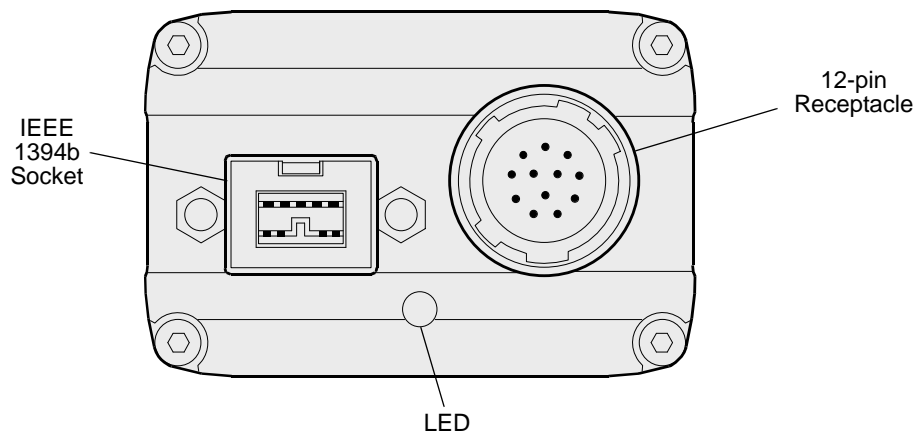


Fig. 8: Camera Connectors and LED

## 5.2 Connector Pin Assignments and Numbering

### 5.2.1 IEEE 1394b Socket Pin Assignments

The IEEE 1394b socket is used to supply power to the camera and to interface video data and control signals. The pin assignments for the socket are as shown in Table 4. Note that these are the standard pin assignments for IEEE 1394b sockets.

Pin	Signal
1	TPB - (twisted pair B minus)
2	TPB + (twisted pair B plus)
3	TPA - (twisted pair A minus)
4	TPA + (twisted pair A plus)
5	TPA R (twisted pair A ground)
6	VG (power ground)
7	Not connected
8	VP (+8 to +36 VDC power)
9	TPB R (twisted pair B ground)

Table 4: Pin Assignments for the IEEE 1394b Socket

Pin numbering for the IEEE 1394b socket is as shown in Section 5.2.3 on [page 26](#).



## 5.2.2 12-pin Receptacle Pin Assignments

The 12 pin receptacle is used to access the one physical input line and one physical output line available on the camera. The pin assignments for the receptacle are shown in Table 5.

Pin	Designation
1	Non-functional
2	Non-functional
3	I/O Input 1
4	Non-functional
5	I/O Input Gnd
6	I/O Output 1
7	Non-functional
8	Non-functional
9	Non-functional
10	I/O Output VCC
11	Non-functional
12	Non-functional

Table 5: Pin Assignments for the 12-pin Receptacle



### CAUTION

#### Avoid Applying Voltage to the Non-functional Pins

Applying incorrect voltages to the non-functional pins in the 12 pin connector may damage the electronic components in the camera. We recommend that you do not apply signals of any kind to the non-functional pins.

Pin numbering for the 12-pin receptacle is as shown in Section 5.2.3 on [page 26](#).

## 5.2.3 Pin Numbering

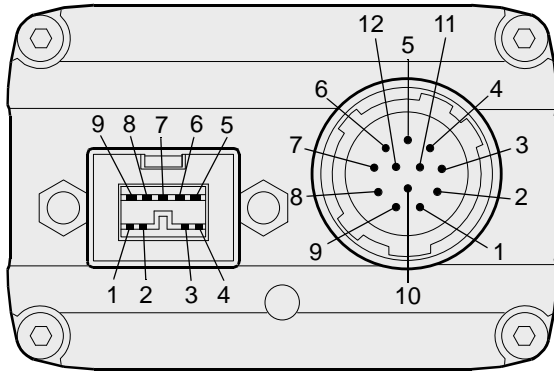


Fig. 9: Pin Numbering for the IEEE 1394b Socket and the 12-pin Receptacle

## 5.3 Connector Types

### 5.3.1 IEEE 1394b Connector

The 1394b socket on the camera is a standard, 9-pin IEEE 1394b bilingual socket.

The recommended mating connector is any standard, 9-pin IEEE 1394b plug.

### 5.3.2 12-pin Connector

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

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

## 5.4 Cabling Requirements

### 5.4.1 IEEE 1394b Cable

The maximum length of the IEEE 1394b cable used between the camera and the adapter in your PC or between the camera and a 1394b hub is 4.5 meters as specified in the IEEE 1394 standard. Standard, 9-pin, shielded 1394b to 1394b cables should be used.

**Note**

The camera is backward compatible with IEEE 1394a devices. If you will be connecting the camera to an IEEE 1394a device, you must use a conversion cable. The cable should have a 9-pin IEEE 1394b plug on the end that connects to the camera and a 6-pin IEEE 1394a plug on the end that connects to the device.

### 5.4.2 I/O Cable

The end of the I/O cable that connects to the camera must be terminated with a Hirose micro plug (part number HR10A-10P-12S) or the equivalent. The cable must be wired as shown in Figure 10.

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

Close proximity to strong magnetic fields should be avoided.

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

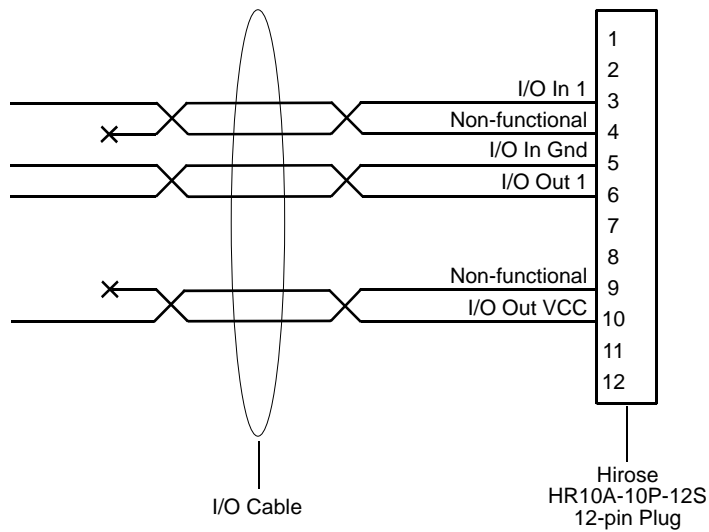


Fig. 10: I/O Cable

**CAUTION****Avoid Applying Voltage to the Non-functional Pins**

Applying incorrect voltages to the non-functional pins in the 12 pin connector may damage the electronic components in the camera. We recommend that you do not apply signals of any kind to the non-functional pins.

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

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

## 5.5 IEEE 1394b Device Information

The camera uses an IEEE1394b - 2002 compliant physical layer device that can transmit at speeds up to 800 Mbit/s (S800). The device is backward compatible with IEEE 1394a - 2000 devices.

Detailed spec sheets for IEEE 1394b - 2002 compliant physical layer devices of the type used in the camera are available at the Texas Instruments website: [www.ti.com](http://www.ti.com).

## 5.6 Camera Power

Camera power must be supplied to the camera via the IEEE 1394b cable. Power consumption is as shown in the specification tables in Section 1 of this manual.

If your camera is connected to an IEEE 1394b adapter in a desktop computer, consult the instructions for the adapter and make sure that the adapter is properly configured to supply power to the camera.

If your camera is connected to a powered hub, consult the instructions for the hub and make sure that it is properly configured to supply power to the camera.

Many laptop computers have a connector for an IEEE 1394 device. In most cases, laptops **do not** supply power to the connected IEEE 1394 device. In this situation, you must use a powered hub between the laptop and the camera or you must install a PCMCIA IEEE 1394 adapter card that connects to an external power supply.



### CAUTION

#### Incorrect Power Can Cause Damage

The polarity of the power on the camera's IEEE 1394b socket must be as shown in the pin assignment table. **Do not** reverse the power polarity. Reversing the polarity will damage the camera.

If the voltage supplied to the camera is greater than +36 VDC, damage to the camera can result. If the voltage is less than +8 VDC, the camera may operate erratically.

The following voltage requirements apply to the camera power (supplied via the IEEE 1394b cable):

Voltage	Significance
< +8 VDC	The camera may operate erratically.
+12 VDC	Recommended operating voltage; < 1 % ripple required.
+36 VDC	Absolute maximum; the camera may be damaged when the absolute maximum is exceeded.

Table 6: Voltage Requirements for the Camera Power

# 5.7 Input and Output Lines

## 5.7.1 I/O Schematic

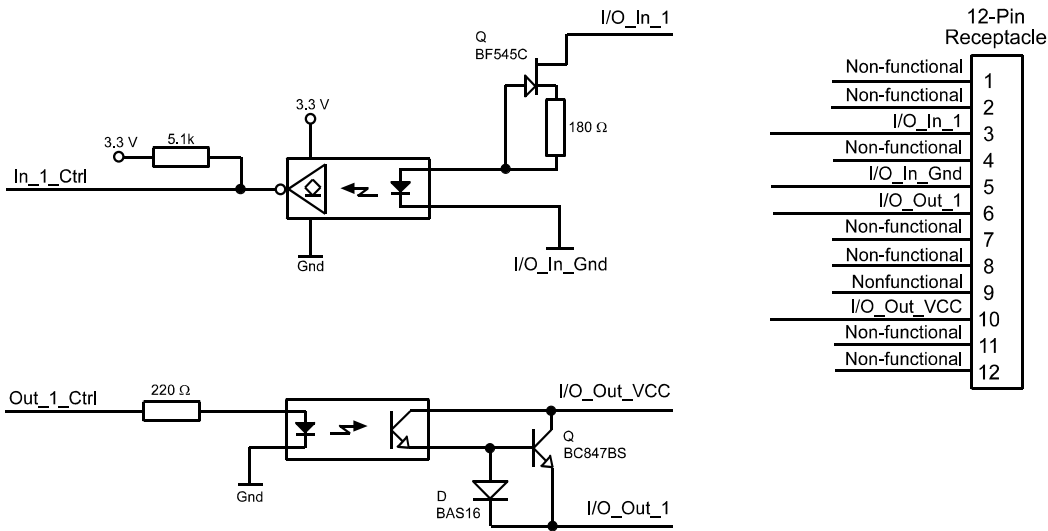


Fig. 11: I/O Line Schematic

## 5.7.2 Input Line Description

### 5.7.2.1 Voltage Requirements

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

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

Table 7: Voltage Requirements for the I/O Input



## 5.7.2.2 Input Line Schematic

The camera is equipped with one physical input line designated as input line 1. The input line is accessed via the 12-pin receptacle on the back of the camera.

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

Figure 12 shows an example of a typical circuit you can use to input a signal into the camera.

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

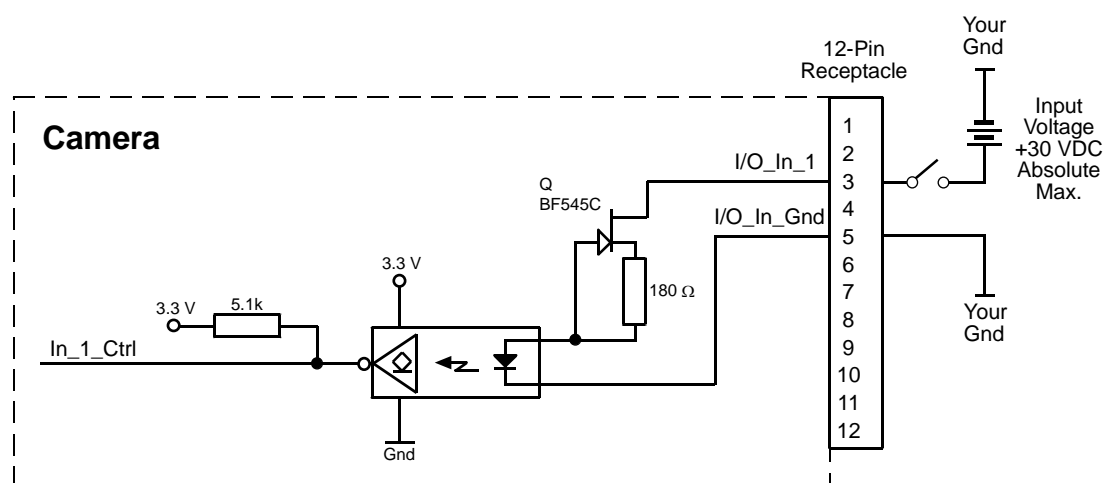


Fig. 12: Typical Input Circuit

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

For more information about how to use an ExTrig signal to control acquisition start, see Section 6.3 on [page 44](#).

For more information about configuring the input line, see Section 8.1 on [page 87](#).

## 5.7.3 Output Line Description

### 5.7.3.1 Voltage Requirements

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

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

Table 8: Voltage Requirements for the I/O Output VCC

### 5.7.3.2 Output Line Schematic

The camera is equipped with one physical output line designated as output line 1. The output line is accessed via the 12-pin receptacle on the back of the camera.

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

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

Figure 13 shows a typical circuit you can use to monitor the output line with a voltage signal.

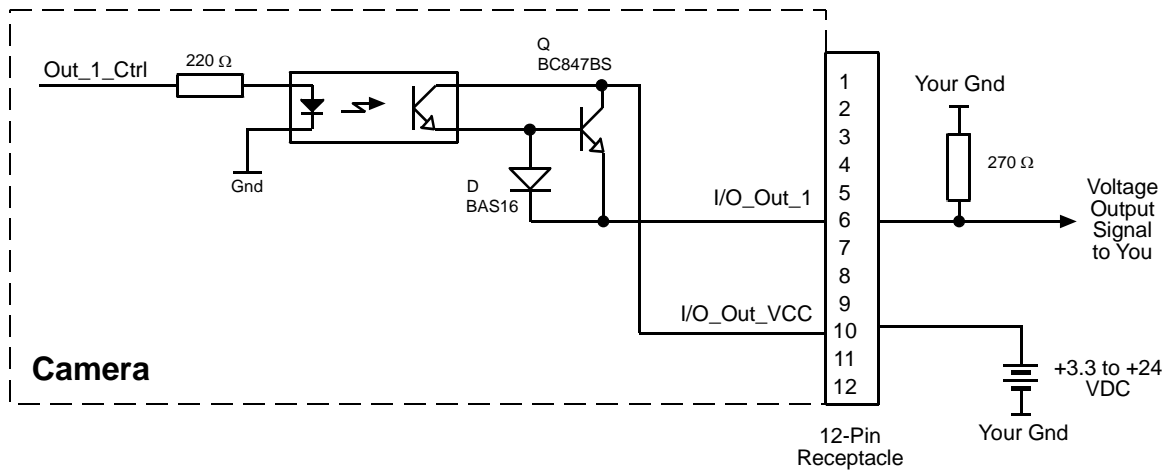


Fig. 13: Typical Voltage Output Circuit

Figure 14 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.

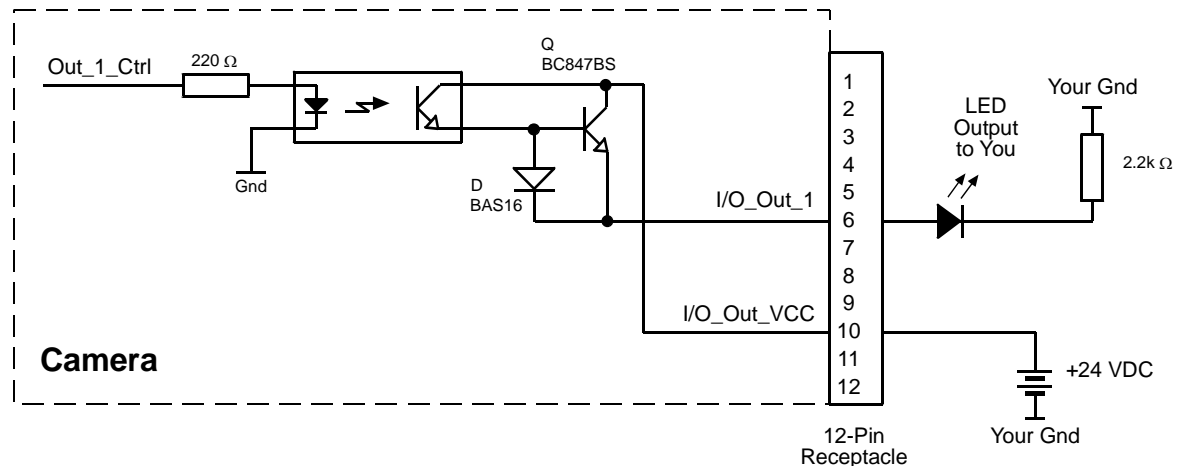


Fig. 14: Typical LED Output Signal at +24 VDC for the External Circuit (Example)

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

The assignment of a camera output signal to the physical output line can be changed by the user.

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

For more information about the exposure active signal, see Section 6.8 on [page 61](#).

For more information about assigning camera output signals to the physical output line, see Section 8.2 on [page 88](#).

# 6 Image Acquisition Control

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

## 6.1 Controlling Image Acquisition with Parameters Only (No Triggering)

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

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

### 6.1.1 Switching Off Triggering

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 6.1.2 Acquiring One Image at a Time

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the camera’s exposure time parameter, see Section 6.4 on [page 51](#).

## 6.1.3 Acquiring Images Continuously (Free-run)

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

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

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the camera's exposure time parameter, see Section 6.4 on [page 51](#).

For more information about determining the maximum allowed acquisition frame rate, see Section 6.10 on [page 65](#).

**Note**

The explanations in Section 6.1.2 and Section 6.1.3 are intended to give you a basic idea of how parameters alone can be used to control image acquisition. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

## 6.2 Controlling Image Acquisition with a Software Trigger

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

Image acquisition starts when the Trigger Software command is executed. The exposure time for each image is determined by the value of the camera's exposure time parameter. Figure 15 illustrates image acquisition with a software trigger.

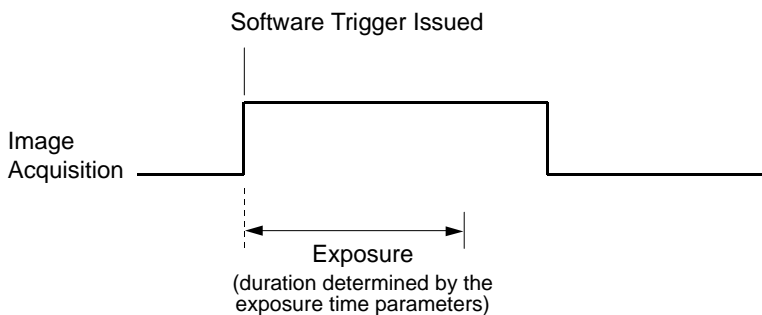


Fig. 15: Image Acquisition with a Software Trigger

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

### 6.2.1 Enabling the Software Trigger Feature

To enable the software trigger feature:

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

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

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

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



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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 6.2.2 Acquiring a Single Image by Applying One Software Trigger

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the camera's exposure time parameter, see Section 6.4 on [page 51](#).

## 6.2.3 Acquiring Images by Applying a Series of Software Triggers

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

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

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

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

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

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

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

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

```
Camera.AcquisitionFrameRateAbs.SetValue( 60.0 );  
double resultingFrameRate = Camera.ResultingFrameRateAbs.GetValue( );  
  
// how to disable the FrameRateAbs parameter  
Camera.AcquisitionFrameRateEnable.SetValue( false );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the camera's exposure time parameter, see Section 6.4 on [page 51](#).

For more information about determining the maximum allowed acquisition frame rate, see Section 6.10 on [page 65](#).

For more information about the Acquisition Status feature, see Section 9.9 on [page 121](#).

**Note**

The explanations in Section 6.2.2 and Section 6.2.3 are intended to give you a basic idea of how the use of a software trigger works. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

## 6.3 Controlling Image Acquisition with a Hardware Trigger

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

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

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

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

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

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

The minimum high time for a rising edge trigger (or low time for a falling edge trigger) is 100 nanoseconds.

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

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

For more information about setting the camera for hardware triggering and selecting the input line to receive the ExTrig signal, see Section 6.3.2 on [page 47](#).

For more information about determining the maximum allowed acquisition frame rate, see Section 6.10 on [page 65](#).

## 6.3.1 Exposure Modes

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

### Timed Exposure Mode

When timed mode is selected, the exposure time for each image is determined by the value of the camera's exposure time parameter. If the camera is set for rising edge triggering, the exposure time starts when the ExTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExTrig signal falls. Figure 16 illustrates timed exposure with the camera set for rising edge triggering.

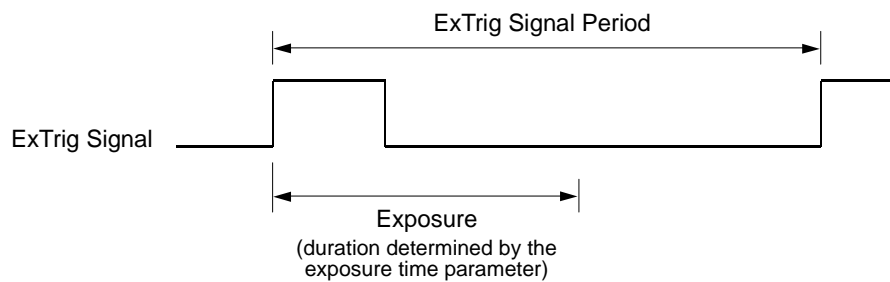


Fig. 16: Timed Exposure with Rising Edge Triggering

### Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure will be directly controlled by the ExTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExTrig signal rises and continues until the ExTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExTrig signal falls and continues until the ExTrig signal rises. Figure 17 illustrates trigger width exposure with the camera set for rising edge triggering.

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

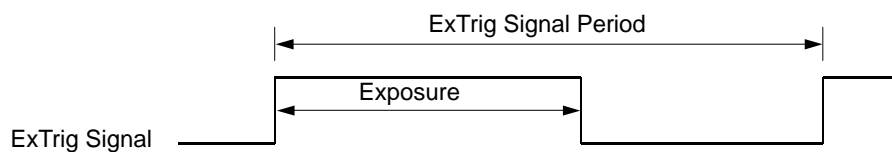


Fig. 17: Trigger Width Exposure with Rising Edge Triggering

**Note**

The trigger width exposure mode **is not** available on sIA750-60fm cameras. The trigger width exposure mode is available on all other camera models.

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

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

If you are using the trigger width exposure mode and the camera is operating with overlapped exposures, there is something you must keep in mind. If the action of the ExTrig signal would end the current exposure while readout of the previously acquired image is still taking place, the camera will automatically continue the exposure until readout of the previous image is complete. This situation is illustrated Figure 16 for rising edge operation. On the first cycle of the ExTrig signal shown in the figure, the signal rises and falls while readout is taking place. Normally you would expect exposure to take place only when the ExTrig signal is high. But since the signal falls while the previous frame is still reading out, the camera automatically extends exposure until the readout is complete. On the second cycle of the ExTrig signal shown in the figure, the signal rises during previous frame readout, but falls after the readout is complete. This is a normal situation and exposure would be determined by the high time of the ExTrig signal as you would expect.

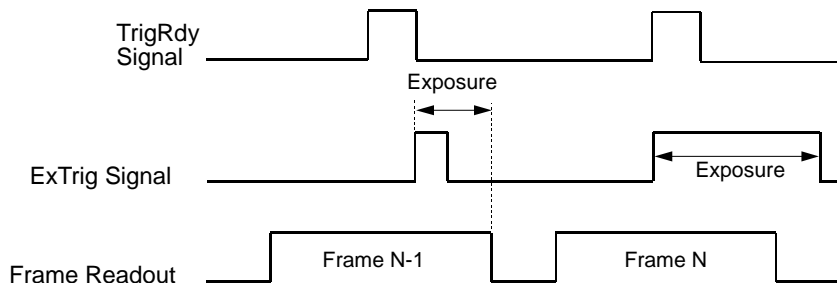


Fig. 18: Trigger Width Exposure Mode with Overlapped Exposure

### Selecting an Exposure Mode

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

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

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

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

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

For more information about the pylon viewer, see Section 3.1 on [page 17](#).

For more information about the camera's exposure time parameter, see Section 6.4 on [page 51](#).

For more information about overlapped exposure, see Section 6.5 on [page 54](#).

For more detailed information about using the trigger width exposure mode with overlapped exposure, refer to the application notes called "Using a Specific External Trigger Signal with Overlapped Exposure" (AW000565xx000). The application notes are available in the downloads section of the Basler website: [www.baslerweb.com/indizes/download\\_index\\_en\\_31412.html](http://www.baslerweb.com/indizes/download_index_en_31412.html).

## 6.3.2 Setting the Camera for Hardware Triggering

To set the camera for hardware triggering:

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 6.3.3 Acquiring a Single Image by Applying One Hardware Trigger Transition

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

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

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

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

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

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

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



For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the Trigger Ready signal, see Section 6.7 on [page 57](#).

For more information about the camera's exposure time parameter, see Section 6.4 on [page 51](#).

### 6.3.4 Acquiring Images by Applying a Series of Hardware Trigger Transitions

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

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

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

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

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the Trigger Ready signal, see Section 6.7 on [page 57](#).

For more information about the camera's exposure time parameter, see Section 6.4 on [page 51](#).



**Note**

The explanations in Section 6.3.3 and Section 6.3.4 are intended to give you a basic idea of how the use of a hardware trigger works. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

## 6.4 Exposure Time Parameters

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

The exposure time must not be set below a minimum specified value. The minimum exposure time varies by camera model as shown in Table 9.

The maximum exposure time that can be set also varies by camera model as shown in Table 9.

Camera Model	Minimum Allowed Exposure Time	Maximum Possible Exposure Time
slA750-60fm	124 $\mu$ s	126976 $\mu$ s
slA1000-30fm	32 $\mu$ s	10000000 $\mu$ s
slA1390-17fm	34 $\mu$ s	10000000 $\mu$ s
slA1600-14fm	31 $\mu$ s	10000000 $\mu$ s

Table 9: Minimum Allowed Exposure Time and Maximum Possible Exposure Time

For information on parameter settings for obtaining the maximum possible exposure time, see Section 6.4.1 on [page 51](#).

### 6.4.1 Setting the Exposure Time Using "Raw" Settings

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

$$\text{Exposure Time} = (\text{Exposure Time Raw Parameter Value}) \times (\text{Exposure Time Base})$$

By default, the Exposure Time Base is fixed at 20  $\mu$ s on all camera models except the slA750-60fm. On slA750-60fm cameras, the default Exposure Time Base is 31  $\mu$ s.

Typically, the exposure time is adjusted by setting only the Exposure Time Raw parameter. The Exposure Time Raw parameter value can range from 1 to 4095. So if the parameter value was set to 100 on an slA1000-30fm camera, for example, the exposure time will be 100 x 20  $\mu$ s or 2000  $\mu$ s.

#### Settings for Obtaining the Maximum Possible Exposure Time

On all camera models except the slA750-60fm, you can obtain the maximum possible exposure time (10000000  $\mu$ s) by setting the Exposure Time Raw parameter value to 1 and the Exposure Time Base Abs value to 10000000  $\mu$ s.

On slA750-60fm cameras, you can obtain the maximum possible exposure time (126976  $\mu$ s) by, e.g., setting the exposure time raw parameter value to 2048 and the Exposure Time Base Abs value to 62  $\mu$ s.

## Changing the Exposure Time Base

Normally, the exposure time is adjusted by setting the value of the Exposure Time Raw parameter as explained above. However, if you require an exposure time that is longer than what you can achieve by changing the value of the Exposure Time Raw parameter alone, the Exposure Time Base Abs parameter can be used to change the exposure time base.

The Exposure Time Base Abs parameter value sets the exposure time base in  $\mu\text{s}$  and this parameter can be used to change the exposure time base.

On all camera models except the sIA750-60fm, the default exposure time base is 20  $\mu\text{s}$  and the time base can be changed in increments of 1  $\mu\text{s}$ .

On sIA750-60fm cameras, the default exposure time base is 31  $\mu\text{s}$  and the time base can be changed in increments of 31  $\mu\text{s}$ .

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

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

Camera.ExposureTimeBaseAbs.SetValue( 186 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 6.4.2 Setting the Exposure Time Using "Absolute" Settings

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

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

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

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

### Setting the Absolute Exposure Time Parameter

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 6.5 Overlapping Exposure and Sensor Readout (All Models Except sIA750-60fm)



### Note

The information in this section applies to all camera models except the sIA750-60fm fm/fc. For information about sIA750-60fm cameras, see Section 6.6 on [page 56](#).

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

In regard to this image acquisition process, there are two common ways for the camera to operate: with “non-overlapped” exposure and with “overlapped” exposure. In the non-overlapped mode of operation, each time an image is acquired, the camera completes the entire exposure/readout process before acquisition of the next image is started. This situation is illustrated in Figure 19.

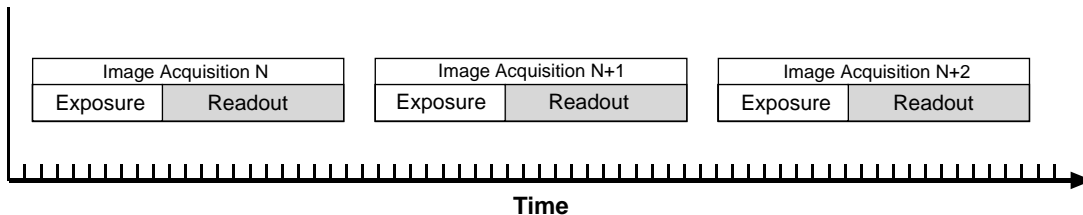


Fig. 19: Non-overlapped Exposure

While operating in a non-overlapped fashion is perfectly normal and is appropriate for many situations, it is not the most efficient way to operate the camera in terms of acquisition frame rate. On this camera, however, it is allowable to begin exposing a new image while a previously acquired image is being read out. This situation is illustrated in Figure 20 and is known as operating the camera with “overlapped” exposure.

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

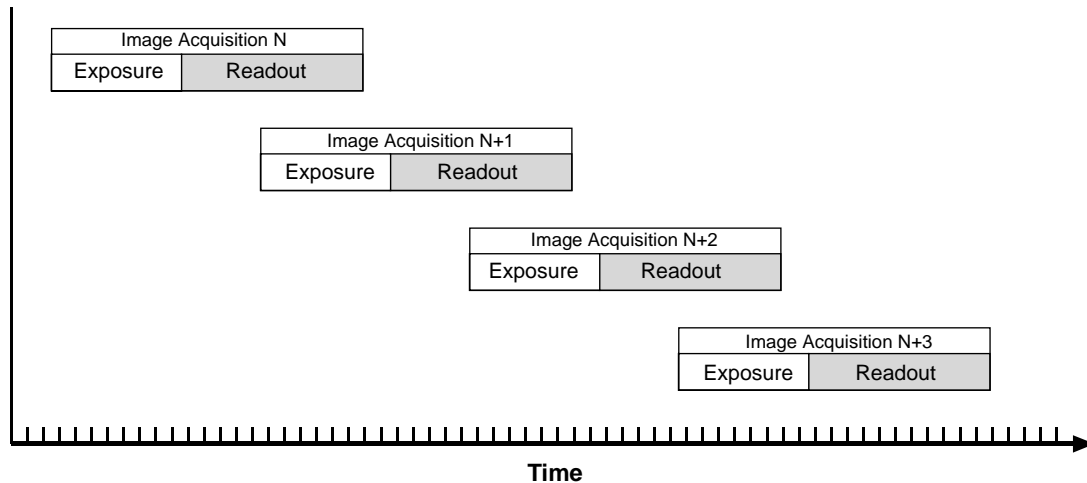


Fig. 20: Overlapped Exposure

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

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

You can calculate the readout time for a captured image by using the formula on [page 62](#).

## 6.5.1 Guidelines for Overlapped Operation

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

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

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

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

For more information about using the Trigger Ready signal with all camera models except the sIA750-60fm, see Section 6.7.1 on [page 57](#).

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

[www.baslerweb.com/indizes/download\\_index\\_en\\_31412.html](http://www.baslerweb.com/indizes/download_index_en_31412.html).

## 6.6 Exposure Must Not Overlap Sensor Readout (sIA750-60fm Only)



### Note

The information in this section only applies to sIA750-60fm cameras. For information about the other camera models, see Section 6.5 on [page 54](#).

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

On these cameras, exposure for a new acquisition must not begin until readout of the previously acquired image is complete. This situation is illustrated in Figure 19.

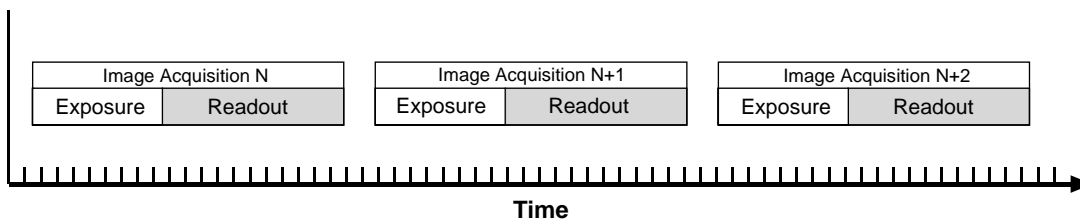


Fig. 21: Non-overlapped Readout and Exposure

A result of this characteristic is that the exposure time setting on the camera will have a direct effect on the camera's maximum allowed frame rate. At longer exposure times, the maximum allowed frame rate will be lower.

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

For more information about using the Trigger Ready signal with sIA750-60fm cameras, see Section 6.7.2 on [page 59](#).



## 6.7 Trigger Ready Signal

### 6.7.1 Trigger Ready Signal (All Models Except sIA750-60fm)

**Note**

The information in this section applies to all camera models except the sIA750-60fm fm/fc. For information about sIA750-60fm cameras, see Section 6.7.2 on [page 59](#).

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

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

The camera supplies a “Trigger Ready” (TrigRdy) output signal you can use to ensure that these conditions are met when you are using a hardware trigger signal to trigger image acquisition. When you are acquiring images, the camera automatically calculates the earliest moment that it is safe to trigger each new acquisition. The trigger ready signal will go high when it is safe to trigger an acquisition, will go low when the acquisition has started, and will go high again when it is safe to trigger the next acquisition (see Figure 22). The camera calculates the rise of the trigger ready signal based on the current exposure time parameter setting, the current size of the area of interest, and the time it will take to readout the captured pixel values from the sensor.

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

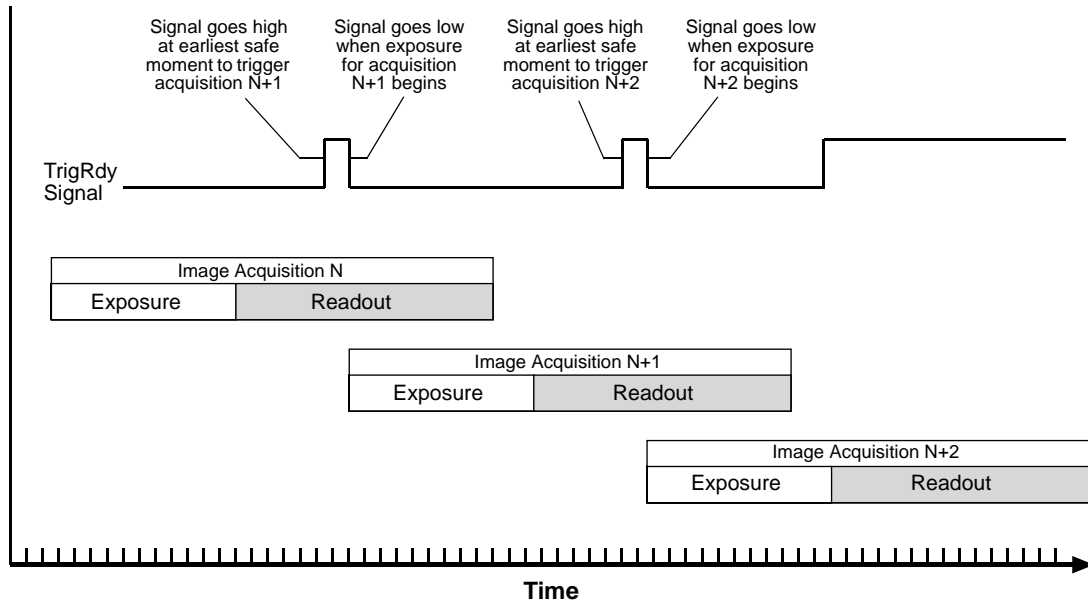


Fig. 22: Trigger Ready Signal

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

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



#### Note

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

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

The trigger ready signal is not normally assigned to the physical output line on the camera. This can be changed, however, and the trigger ready signal can be assigned to the camera's output line.

For more information about changing the output signal assigned to the camera's output line, see Section 8.2 on [page 88](#).

For more information about the electrical characteristics of the camera's output line, see Section 5.7.3 on [page 34](#).

## 6.7.2 Trigger Ready Signal (sIA750-60fm Only)



### Note

The information in this section only applies to sIA750-60fm cameras. For information about the other camera models, see Section 6.7.1 on [page 57](#).

As described in an earlier section, on these cameras the exposure for an image acquisition must not begin until readout of the previously acquired image has ended. The camera supplies a “Trigger Ready” (TrigRdy) output signal you can use to ensure that these conditions are met when you are using a hardware trigger signal to trigger image acquisition. When you are acquiring images, the camera automatically calculates the earliest moment that it is safe to trigger each new acquisition. The trigger ready signal will go high when it is safe to trigger an acquisition, will go low when the acquisition has started, and will go high again when it is safe to trigger the next acquisition (see Figure 22). The camera calculates the rise of the trigger ready signal based on the current exposure time parameter setting, the current size of the area of interest, and the time it will take to readout the captured pixel values from the sensor.

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

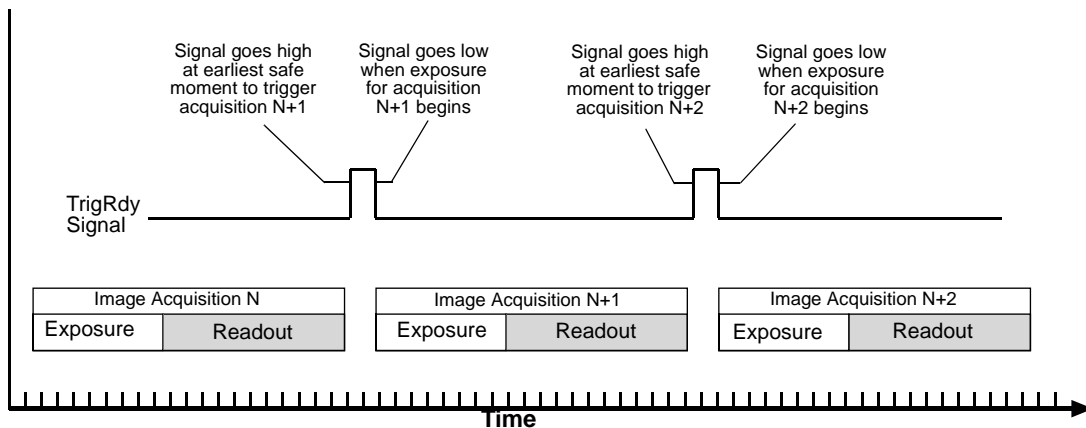


Fig. 23: Trigger Ready Signal

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

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



**Note**

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

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

The trigger ready signal is not normally assigned to the physical output line on the camera. This can be changed, however, and the trigger ready signal can be assigned to the camera's output line.

For more information about changing the output signal assigned to the camera's output line, see Section 8.2 on [page 88](#).

For more information about the electrical characteristics of the camera's output line, see Section 5.7.3 on [page 34](#).

## 6.8 Exposure Active Signal

The camera's "exposure active" (ExpAc) signal goes high when the exposure time for each image acquisition begins and goes low when the exposure time ends as shown in Figure 24. 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.

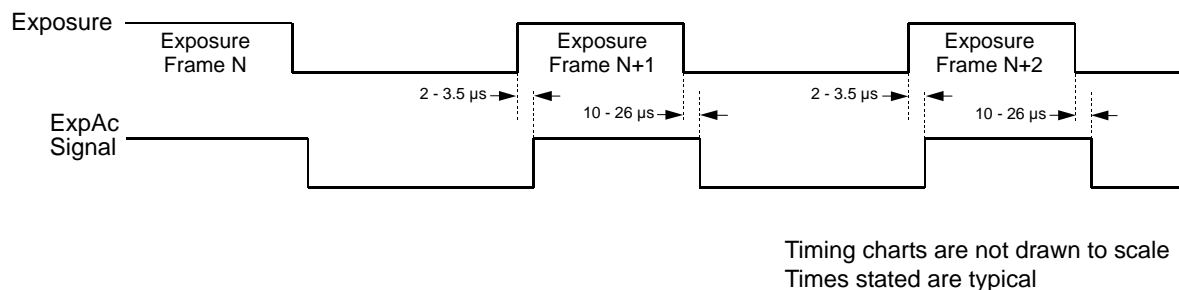


Fig. 24: Exposure Active Signal



### Note

When you use the exposure active signal, be aware that there is a delay in the rise and the fall of the signal in relation to the start and the end of exposure. See Figure 24 for details.

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

For more information about changing the output signal assigned to the camera's output line, see Section 8.2 on [page 88](#).

For more information about the electrical characteristics of the camera's output line, see Section 5.7.3 on [page 34](#).

## 6.9 Acquisition Timing Chart

Figure 25 shows a timing chart for image acquisition and transmission. The chart assumes that exposure is triggered with an ExTrig signal with rising edge activation and that the camera is set for programmable exposure mode.

As Figure 25 shows, there is a slight delay between the rise of the ExTrig signal and the start of exposure. After the exposure time for an image capture is complete, the camera begins reading out the captured image data from the imaging sensor into a buffer in the camera. When the camera has determined that a sufficient amount of image data has accumulated in the buffer, it will begin transmitting the data from the camera to the host PC.

This buffering technique avoids the need to exactly synchronize the clock used for sensor readout with the clock used for data transmission over the IEEE 1394b bus. The camera will begin transmitting data when it has determined that it can safely do so without over-running or under-running the buffer. This buffering technique is also an important element in achieving the highest possible frame rate with the best image quality.

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

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

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

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

Camera Model	Exposure Start Delay
slA750-60fm	180.0 $\mu$ s
slA1000-30fm	45.33 $\mu$ s
slA1390-17fm	58.90 $\mu$ s
slA1600-14fm	60.52 $\mu$ s

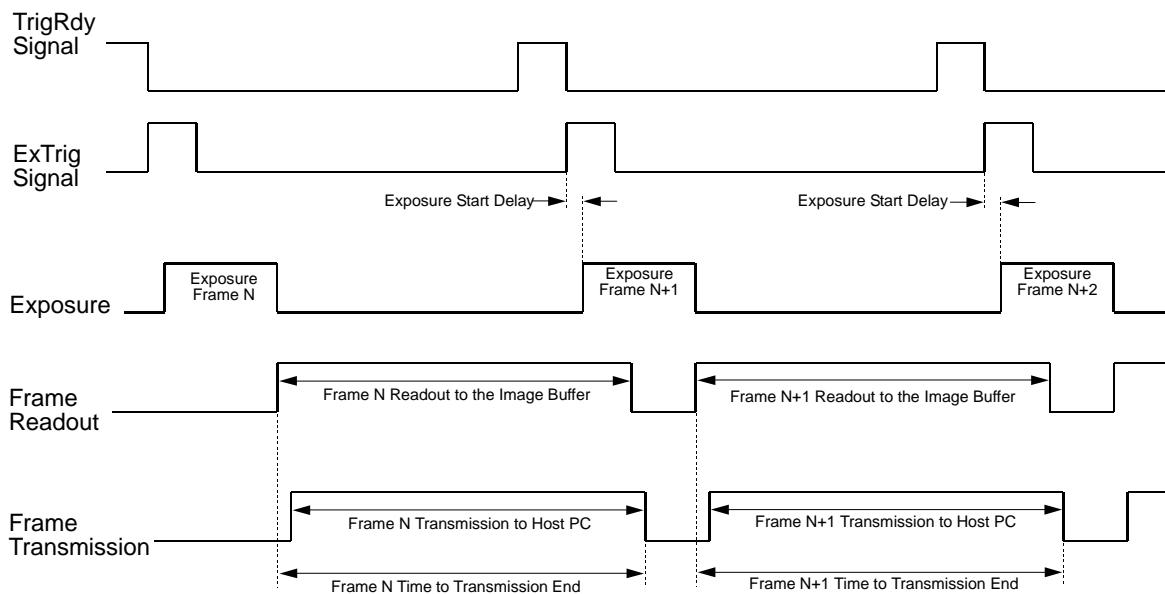
Table 10: Exposure Start Delays

Note that, if the debouncer feature is used, the debouncer setting for the input line must be added to the exposure start delays shown in Table 10 to determine the total start delay. For example, assume that you are using an slA1000-30fm camera and that you have set the cameras for hardware triggering. Also assume that you have selected input line 1 to accept the hardware trigger signal and that you have set the Line Debouncer Time Abs parameter for input line 1 to 5  $\mu$ s. In this case:

Total Start Delay = Start Delay from Table 10+ Debouncer Setting

Total Start Delay = 45.33  $\mu$ s+ 5  $\mu$ s

Total Start Delay = 50.33  $\mu$ s



Timing charts are not drawn to scale

Fig. 25: Exposure Start Controlled with an ExTrig Signal

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

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

Where the values for the constants  $C_1$  and  $C_2$  are from the table in Section 6.10 on [page 65](#) for all camera models except the sIA750-60fm or from the table in Section 6.11 on [page 70](#) for sIA750-60fm cameras.

For more information about the frame height, see Section 9.4 on [page 111](#).

You can calculate the time to transmission end ( $T_e$ ) using these three steps:

1. Calculate the frame readout time ( $T_r$ ) using the formula above.
2. Calculate the base transmission time ( $T_b$ ) using these formulas:

$$\text{Packets per frame} = \frac{\text{Value of the Payload Size Parameter}}{\text{Value of the Packet Size Parameter}}$$

(round the result up to the nearest integer)

$$T_b = \text{Packets per frame} \times 125 \mu\text{s}$$

3. Compare the results:

If  $T_b \leq T_r$ , then  $T_e = T_r + 250 \mu\text{s}$

If  $T_b > T_r$ , then  $T_e = T_b + 250 \mu\text{s}$

You can determine the value of the Payload Size and Packet Size parameters from within your application software by using the pylon API. The following code snippet illustrates using the API to work with the parameter values:

```
// Get payload size
int64_t payloadSize = Camera.PayloadSize.GetValue();

// Set packet size
Camera.PacketSizeSize.SetValue( 4096 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).



## 6.10 Maximum Allowed Acquisition Frame Rate (All Models Except sIA750-60fm)



### Note

The information in this section applies to all camera models except the sIA750-60fm. For information about sIA750-60fm cameras, see Section 6.11 on [page 70](#).

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

- The amount of time it takes to read an acquired image out of the imaging sensor and into the camera's frame buffer (an acquired image is also known as a frame). This time varies depending on the height of the frame. Shorter frames take less time to read out of the sensor. The frame height is determined by the camera's AOI settings.
- The exposure time for acquired frames. If you use very long exposure times, you can acquire fewer frames per second.
- The number of packets needed to transfer an acquired frame from the camera to your PC.

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

You can read the current value of the Resulting Frame RateAbs parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to get the parameter values:

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the AOI settings, see Section 9.4 on [page 111](#).

## Increasing the Maximum Allowed Frame Rate

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

- You will often find that the sensor readout time is most restrictive factor. Decreasing the height of the acquired frames will decrease the sensor readout time and will make this factor less restrictive.
- If you find that the number of packets needed to transmit an image is restricting the frame rate, you may be able to decrease the number of packets needed to transmit a frame. The next section in this manual explains more about the effect of changing the packets per frame.
- If you are using normal exposure times and you are using the camera at its maximum resolution, your exposure time will not normally be the most restrictive factor on the frame rate. However, if you are using long exposure times or small areas of interest, it is quite possible to find that your exposure time is the most restrictive factor on the frame rate. In this case, you should lower your exposure time. (You may need to compensate for a lower exposure time by using a brighter light source or increasing the opening of your lens aperture.)

For more information about the AOI settings, see Section 9.4 on [page 111](#).

### Formula 1:

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

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

Where:

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

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

	sIA1000-30fm	sIA1390-17fm	sIA1600-14fm
$C_1$	36.57 $\mu\text{s}$	44.71 $\mu\text{s}$	52.37 $\mu\text{s}$
$C_2$	4699 $\mu\text{s}$	12215 $\mu\text{s}$	6896 $\mu\text{s}$

**Formula 2:**

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

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

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

	sIA1000-30fm	sIA1390-17fm	sIA1600-14fm
$C_3$	136.47 $\mu\text{s}$	176.76 $\mu\text{s}$	181.64 $\mu\text{s}$

**Formula 3:**

Calculates the maximum frame rate based on the number of packets needed to transmit a captured frame from the camera to your host PC via the IEEE 1394 bus:

$$\text{Packets per frame} = \frac{\text{Value of the Payload Size Parameter}}{\text{Value of the Packet Size Parameter}}$$

(round the result up to the nearest integer)

$$\text{Max. Frames/s} = \frac{1}{\text{Packets per frame} \times 125 \mu\text{s}}$$

**Example**

Assume that you are using a monochrome sIA1000-30fm camera set for an exposure time of 2000  $\mu\text{s}$  and for 600 x 400 resolution. Also assume that you have checked the value of the Payload Size parameter and the Packet Size parameters and found them to be 327100 and 8192 respectively.

**Formula 1:**

$$\text{Max Frames/s} = \frac{1}{(400 \times 36.57 \mu\text{s}) + 4699 \mu\text{s}}$$

$$\text{Max Frames/s} = 51.7 \text{ frames/s}$$

**Formula 2:**

$$\text{Max Frames/s} = \frac{1}{2000 \mu\text{s} + 36.47 \mu\text{s}}$$

$$\text{Max Frames/s} = 491.0 \text{ frames/s}$$

**Formula 3:**

$$\text{Packets per frame} = \frac{327100}{8192}$$

$$\text{Packets per frame} = 39.9$$

(Round the result up to 40.)

$$\text{Max. Frames/s} = \frac{1}{40 \times 125 \mu\text{s}}$$

$$\text{Max Frames/s} = 200 \text{ frames/s}$$

Formula one returns the lowest value. So in this case, the limiting factor is the sensor readout time and the maximum allowed acquisition frame rate would be 51.7 frames per second.

## 6.10.1 Effect of the Packet Size Setting on the Maximum Allowed Frame Rate

After a camera acquires a frame, the image data is read out from the sensor into a buffer. Once the frame has been read out to the buffer, the data is packetized and transmitted across the IEEE 1394b bus to your host PC.

A parameter called Packet Size determines the number of bytes of data that will be included in each packet transferred across the bus. The minimum value for the Packet Size parameter is 1 and the maximum value is 8192. Normally, the value of the Packet Size parameter is set to the maximum and at maximum, the Packet Size parameter has no noticeable effect on the operation of the camera.

If you lower the value of the Packet Size parameter, the amount of image data included in each packet transmitted across the bus will be lower. This means that it will take more packets to transmit each frame and since the cycle time of the IEEE 1394b bus is fixed, it also means that it will take more time to transmit each frame. If you lower the Packet Size parameter enough, the slower data transfer rate can begin to affect the maximum allowed frame capture rate of your camera. If you look at the formulas the previous section, you will notice that one of the factors that can limit the maximum allowed frame rate is the number of packets needed to transmit a frame. The number of packets per frame is directly related to the Packet Size parameter setting.

You can see the effect of changing the Packet Size parameter by looking at the read only parameter called Resulting Frame Rate Abs. The Resulting Frame Rate Abs parameter indicates the

maximum frame rate with the current camera settings. If you gradually decrease the setting for the Packet Size parameter, you will eventually find that the value of the Resulting Frame Rate Abs parameter will also decrease.

If you are operating a single camera on your IEEE 1394b bus, you would ordinarily leave the Packet Size parameter set at the maximum. However, if you are operating multiple cameras on a single IEEE 1394b bus, you will probably need to change the Packet Size parameter so that the cameras can effectively share the available bus bandwidth.

You can set the value of the Packet Size parameter and read the current value of the resulting Frame Rate parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to work with the parameter values:

```
// Set packet size
Camera.PacketSize.SetValue( 4096 );

// Get resulting framerate
double resultingFps = Camera.ResultingFrameRateAbs.GetValue();
```

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

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

For more information about using multiple cameras on a single bus, see Section 10 on [page 133](#).

## 6.11 Maximum Allowed Acquisition Frame Rate (sIA750-60fm Only)



### Note

The information in this section only applies to sIA750-60fm cameras. For information about the other camera models, see Section 6.10 on [page 65](#).

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

- The sum of the exposure time plus the amount of time it takes to read the acquired image out of the imaging sensor and into the camera's frame buffer. (An acquired image is also known as a frame.)

The exposure time is set by the user. If you use very long exposure times, you can acquire fewer frames per second.

The readout time varies depending on the height of the frame. Shorter frames take less time to read out of the sensor. The frame height is determined by the camera's AOI Height settings.

- The number of packets needed to transfer an acquired frame from the camera to your PC.

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

You can read the current value of the Resulting Frame RateAbs parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to get the parameter values:

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the AOI settings, see Section 9.4 on [page 111](#).

## Increasing the Maximum Allowed Frame Rate

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

- You will often find that the sum of the exposure time plus the sensor readout time is the most restrictive factor.

Decreasing the AOI height for the acquired frames will decrease the sensor readout time and will make this factor less restrictive.

If you are using long exposure times, it is quite possible to find that your exposure time is making this factor the most restrictive. In this case, you should lower your exposure time. (You may need to compensate for a lower exposure time by using a brighter light source or increasing the opening of your lens aperture.)

- If you find that the number of packets needed to transmit an image is restricting the frame rate, you may be able to decrease the number of packets needed to transmit a frame. The next section in this manual explains more about the effect of changing the packets per frame.

For more information about the AOI settings, see Section 9.4 on [page 111](#).

### Formula 1:

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

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

Where:

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

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

	sIA750-60 fm
$C_1$	31.0 $\mu\text{s}$
$C_2$	397.0 $\mu\text{s}$

**Formula 2:**

Calculates the maximum frame rate based on the number of packets needed to transmit a captured frame from the camera to your host PC via the IEEE 1394 bus:

$$\text{Packets per frame} = \frac{\text{Value of the Payload Size Parameter}}{\text{Value of the Packet Size Parameter}}$$

(round the result up to the nearest integer)

$$\text{Max. Frames/s} = \frac{1}{\text{Packets per frame} \times 125 \mu\text{s}}$$

**Example**

Assume that you are using a monochrome sIA750-60fm camera set for an exposure time of 2000  $\mu\text{s}$  and for 600 x 400 resolution. Also assume that you have checked the value of the Payload Size parameter and the Packet Size parameters and found them to be 327100 and 8192 respectively.

**Formula 1:**

$$\text{Max. Frames/s} = \frac{1}{2000 \mu\text{s} + (400 \times 31.0 \mu\text{s}) + 397.0 \mu\text{s}}$$

$$\text{Max. Frames/s} = 67.6 \text{ frames/s}$$

**Formula 2:**

$$\text{Packets per frame} = \frac{327100}{8192}$$

$$\text{Packets per frame} = 39.9$$

(Round the result up to 40.)

$$\text{Max. Frames/s} = \frac{1}{40 \times 125 \mu\text{s}}$$

$$\text{Max Frames/s} = 200 \text{ frames/s}$$

Formula one returns the lowest value. So in this case, the limiting factor is the sum of the exposure time plus the sensor readout time and the maximum allowed acquisition frame rate would be 67.6 frames per second.



## 6.11.1 Effect of the Packet Size Setting on the Maximum Allowed Frame Rate

After a camera acquires a frame, the image data is read out from the sensor into a buffer. Once the frame has been read out to the buffer, the data is packetized and transmitted across the IEEE 1394b bus to your host PC.

A parameter called Packet Size determines the number of bytes of data that will be included in each packet transferred across the bus. The minimum value for the Packet Size parameter is 1 and the maximum value is 8192. Normally, the value of the Packet Size parameter is set to the maximum and at maximum, the Packet Size parameter has no noticeable effect on the operation of the camera.

If you lower the value of the Packet Size parameter, the amount of image data included in each packet transmitted across the bus will be lower. This means that it will take more packets to transmit each frame and since the cycle time of the IEEE 1394b bus is fixed, it also means that it will take more time to transmit each frame. If you lower the Packet Size parameter enough, the slower data transfer rate can begin to affect the maximum allowed frame capture rate of your camera. If you look at the formulas the previous section, you will notice that one of the factors that can limit the maximum allowed frame rate is the number of packets needed to transmit a frame. The number of packets per frame is directly related to the Packet Size parameter setting.

You can see the effect of changing the Packet Size parameter by looking at the read only parameter called Resulting Frame Rate Abs. The Resulting Frame Rate Abs parameter indicates the maximum frame rate with the current camera settings. If you gradually decrease the setting for the Packet Size parameter, you will eventually find that the value of the Resulting Frame Rate Abs parameter will also decrease.

If you are operating a single camera on your IEEE 1394b bus, you would ordinarily leave the Packet Size parameter set at the maximum. However, if you are operating multiple cameras on a single IEEE 1394b bus, you will probably need to change the Packet Size parameter so that the cameras can effectively share the available bus bandwidth.

You can set the value of the Packet Size parameter and read the current value of the resulting Frame Rate parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to work with the parameter values:

```
// Set packet size
Camera.PacketSize.SetValue( 4096 );

// Get resulting framerate
double resultingFps = Camera.ResultingFrameRateAbs.GetValue();
```

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

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

For more information about using multiple cameras on a single bus, see Section 10 on [page 133](#).



# 7 Pixel Data Formats

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

## 7.1 Setting the Pixel Data Format

The setting for the camera's Pixel Format parameter determines the format of the pixel data that will be output from the camera. The available pixel formats depend on the camera model. Table 11 lists the pixel formats available on each camera model.

Mono Camera Model	Mono 8	Mono 16	Mono 12 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
slA750-60fm	•			•	•
slA1000-30fm	•	•	•	•	•
slA1390-17fm	•	•	•	•	•
slA1600-14fm	•	•	•	•	•

Table 11: Pixel Formats Available on Each Camera Model (• = format available)

Details of the formats are described in Section 7.2 on [page 76](#).

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 7.2 Pixel Data Formats

### 7.2.1 Mono 8 Format (Equivalent to DCAM Mono 8)

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

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono8 output.

The following standards are used in the table:

$P_0$  = the first pixel transmitted by the camera

$P_n$  = the last pixel transmitted by the camera

$B_0$  = the first byte in the buffer

$B_m$  = the last byte in the buffer

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

With the camera set for Mono8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

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

## 7.2.2 Mono 16 Format (Equivalent to DCAM Mono 16)

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

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

The following standards are used in the table:

$P_0$  = the first pixel transmitted by the camera

$P_n$  = the last pixel transmitted by the camera

$B_0$  = the first byte in the buffer

$B_m$  = the last byte in the buffer

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

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

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

**Note**

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

## 7.2.3 Mono 12 Packed Format

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

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

The following standards are used in the table:

$P_0$  = the first pixel transmitted by the camera

$P_n$  = the last pixel transmitted by the camera

$B_0$  = the first byte in the buffer

$B_m$  = the last byte in the buffer

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



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

<b>This Data Value (Hexadecimal)</b>	<b>Indicates This Signal Level (Decimal)</b>
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

## 7.2.4 YUV 4:2:2 Packed Format (Equivalent to DCAM YUV 4:2:2)

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

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

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

The following standards are used in the table:

$P_0$  = the first pixel transmitted by the camera

$P_n$  = the last pixel transmitted by the camera

$B_0$  = the first byte in the buffer

$B_m$  = the last byte in the buffer

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

B <sub>m-3</sub>	U value for P <sub>n-1</sub>
B <sub>m-2</sub>	Y value for P <sub>n-1</sub>
B <sub>m-1</sub>	V Value for P <sub>n-1</sub>
B <sub>m</sub>	Y value for P <sub>n</sub>

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

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

The pixel data output for the U component or the V component is 8 bit data of the “straight binary” type. The data values for a U or a V component will always be zero.

## 7.2.5 YUV 4:2:2 (YUYV) Packed Format

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

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

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

The following standards are used in the table:

P<sub>0</sub> = the first pixel transmitted by the camera

$P_n$  = the last pixel transmitted by the camera

$B_0$  = the first byte in the buffer

$B_m$  = the last byte in the buffer

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

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

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

The pixel data output for the U component or the V component is 8 bit data of the “straight binary” type. The data values for a U or a V component will always be zero.

## 7.3 Pixel Transmission Sequence

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

Row <sub>0</sub> Col <sub>0</sub> ,	Row <sub>0</sub> Col <sub>1</sub> ,	Row <sub>0</sub> Col <sub>2</sub>	.. ..	Row <sub>0</sub> Col <sub>m-2</sub> ,	Row <sub>0</sub> Col <sub>m-1</sub> ,	Row <sub>0</sub> Col <sub>m</sub>
Row <sub>1</sub> Col <sub>0</sub> ,	Row <sub>1</sub> Col <sub>1</sub> ,	Row <sub>1</sub> Col <sub>2</sub>	.. ..	Row <sub>1</sub> Col <sub>m-2</sub> ,	Row <sub>1</sub> Col <sub>m-1</sub> ,	Row <sub>1</sub> Col <sub>m</sub>
Row <sub>2</sub> Col <sub>0</sub> ,	Row <sub>2</sub> Col <sub>1</sub> ,	Row <sub>2</sub> Col <sub>2</sub>	.. ..	Row <sub>2</sub> Col <sub>m-2</sub> ,	Row <sub>2</sub> Col <sub>m-1</sub> ,	Row <sub>2</sub> Col <sub>m</sub>
:	:	:		:	:	:
:	:	:		:	:	:
Row <sub>n-2</sub> Col <sub>0</sub> ,	Row <sub>n-2</sub> Col <sub>1</sub> ,	Row <sub>n-2</sub> Col <sub>2</sub>	.. ..	Row <sub>n-2</sub> Col <sub>m-2</sub> ,	Row <sub>n-2</sub> Col <sub>m-1</sub> ,	Row <sub>n-2</sub> Col <sub>m</sub>
Row <sub>n-1</sub> Col <sub>0</sub> ,	Row <sub>n-1</sub> Col <sub>1</sub> ,	Row <sub>n-1</sub> Col <sub>2</sub>	.. ..	Row <sub>n-1</sub> Col <sub>m-2</sub> ,	Row <sub>n-1</sub> Col <sub>m-1</sub> ,	Row <sub>n-1</sub> Col <sub>m</sub>
Row <sub>n</sub> Col <sub>0</sub> ,	Row <sub>n</sub> Col <sub>1</sub> ,	Row <sub>n</sub> Col <sub>2</sub>	.. ..	Row <sub>n</sub> Col <sub>m-2</sub> ,	Row <sub>n</sub> Col <sub>m-1</sub> ,	Row <sub>n</sub> Col <sub>m</sub>

Where Row<sub>0</sub> Col<sub>0</sub> is the upper left corner of the sensor

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

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

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

# 8 I/O Control

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

For more detailed information about the physical and electrical characteristics of the input and output lines, see Section 5.4 on [page 28](#).

## 8.1 Configuring the Input Line

### 8.1.1 Assigning the Input Line to Receive a Hardware Trigger Signal

The camera is equipped with one physical input line designated as input line 1. You can assign the camera's input line to receive an external hardware trigger (ExTrig) signal. The incoming ExTrig signal can then be used to control image acquisition.

Section 6.3.2 on [page 47](#) explains how to configure the camera to react to a hardware trigger signal and how to assign the input line to receive the hardware trigger signal.

**Note**

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

## 8.2 Configuring the Output Line

### 8.2.1 Assigning a Camera Output Signal to the Physical Output Line

The camera is equipped with one physical output line designated as output line 1. You can use the camera's output signal assignment capability to assign one of the camera's standard output signals as the source signal for physical output line 1. The camera has a variety of standard output signals available including:

- Exposure Active
- Trigger Ready
- Timer 1

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

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

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

**Note**

By default, the Exposure Active signal is assigned to output line 1.

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about setting the state of a user settable output line, see Section 8.2.2 on [page 89](#).

For more information about working with a timer output signal, see Section 8.2.4 on [page 91](#)

For more information about the exposure active signal, see Section 6.8 on [page 61](#).

For more information about the trigger ready signal, see Section 6.7 on [page 57](#).



## 8.2.2 Setting the State of a User Settable Output Line

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

### Setting the State of a User Settable Output Line

To set the state of a user settable output line:

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

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

```
Camera.LineSelector.SetValue( LineSelector_Out1 );  
Camera.LineSource.SetValue( LineSource_UserOutput );  
Camera.UserOutputSelector.SetValue( UserOutputSelector_UserOutput1 );  
Camera.UserOutputValue.SetValue( true );  
bool currentUserOutput1State = Camera.UserOutputValue.GetValue( );
```

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

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



#### Note

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.

## 8.2.3 Setting the Output Line for Invert

You can set the output line to invert or not to invert the outgoing signal. To set the invert function on the output line:

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 8.2.4 Working with the Timer Signal

The camera a timer output signal available called timer 1. The timer works as follows:

- A trigger source event occurs that starts the timer.
- A delay period begins to expire.
- When the delay expires, the timer signal goes high and a duration period begins to expire.
- When the duration period expires, the timer signal goes low.

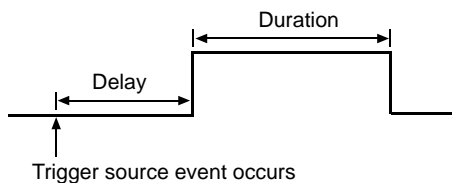


Fig. 26: Timer Signal

Currently, the only trigger source event available to start the timer is "exposure active". In other words, you can use exposure start to trigger the start of the timer.

If you require the timer signal to be high when the timer is triggered and to go low when the delay expires, simply set the output line to invert.

### 8.2.4.1 Setting the Trigger Source for the Timer

To set the trigger source for a timer:

- Use the Timer Selector to select timer 1.
- Set the value of the Timer Trigger Source parameter to exposure active. This will set the selected timer to use the start of exposure to begin the timer.

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 8.2.4.2 Setting the Timer Delay Time

There are two ways to set the delay time for timer 1: by setting "raw" values or by setting an "absolute value". You can use whichever method you prefer to set the delay time.

### Setting the Delay Time with Raw Values

When the delay time for timer 1 is set using "raw" values, the delay time will be determined by a combination of two elements. The first element is the value of the Timer Delay Raw parameter, and the second element is the Timer Delay Time Base. The delay time is the product of these two elements:

Delay Time = (Timer Delay Raw Parameter Value) x (Timer Delay Time Base)

By default, the Timer Delay Time Base is fixed at 1  $\mu$ s. Typically, the delay time is adjusted by setting the Timer Delay Raw parameter value.

The Timer Delay Raw parameter value can range from 0 to 4095. So if the value is set to 100, for example, the timer delay will be 100 x 1  $\mu$ s or 100  $\mu$ s.

To set the delay for timer 1:

- Use the Timer Selector to select timer 1.
- Set the value of the Timer Delay Raw parameter.

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );  
Camera.TimerDelayRaw.SetValue( 100 );
```

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

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

### Changing the Delay Time Base

By default, the Timer Delay Time Base is fixed at 1  $\mu$ s (minimum value), and the timer delay is normally adjusted by setting the value of the Timer Delay Raw parameter. However, if you require a delay time that is longer than what you can achieve by changing the value of the Timer Delay Raw parameter alone, the Timer Delay Time Base Abs parameter can be used to change the delay time base.

The Timer Delay Time Base Abs parameter value sets the delay time base in  $\mu$ s. The default is 1  $\mu$ s and it can be changed in 1  $\mu$ s increments.

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

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

---

## Setting the Delay Time with an Absolute Value

You can also set the timer 1 delay by using an "absolute" value. This is accomplished by setting the Timer Delay Abs parameter. The units for setting this parameter are  $\mu\text{s}$  and the value can be set in increments of 1  $\mu\text{s}$ .

To set the delay for timer 1 using an absolute value:

- Use the Timer Selector to select timer 1.
- Set the value of the Timer Delay Abs parameter.

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );  
Camera.TimerDelayAbs.SetValue( 100 );
```

When you use the Timer Delay Abs parameter to set the delay time, the camera accomplishes the setting change by automatically changing the Timer Delay Raw parameter to achieve the value specified by the Timer Delay Abs setting. This leads to a limitation that you must keep in mind if you use Timer Delay Abs parameter to set the delay time. That is, you must set the Timer Delay Abs parameter to a value that is equivalent to a setting you could achieve by using the Timer Delay Raw and the current Timer Delay Base parameters. For example, if the time base was currently set to 50  $\mu\text{s}$ , you could use the Timer Delay Abs parameter to set the delay to 50  $\mu\text{s}$ , 100  $\mu\text{s}$ , 150  $\mu\text{s}$ , etc.

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

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

### 8.2.4.3 Setting the Timer Duration Time

There are two ways to set the duration time for timer 1: by setting "raw" values or by setting an "absolute value". You can use whichever method you prefer to set the duration time.

#### Setting the Duration Time with Raw Values

When the duration time for timer 1 is set using "raw" values, the duration time will be determined by a combination of two elements. The first element is the value of the Timer Duration Raw parameter, and the second element is the Timer Duration Time Base. The duration time is the product of these two elements:

Duration Time = (Timer Duration Raw Parameter Value) x (Timer Duration Time Base)

By default, the Timer Duration Time Base is fixed at 1  $\mu$ s. Typically, the duration time is adjusted by setting only the Timer Duration Raw parameter value.

The Timer Duration Raw parameter value can range from 1 to 4095. So if the value is set to 100, for example, the timer duration will be 100 x 1  $\mu$ s or 100  $\mu$ s.

To set the duration for timer 1:

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

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );  
Camera.TimerDurationRaw.SetValue( 100 );
```

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

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

#### Changing the Duration Time Base

By default, the Timer Duration Time Base is fixed at 1  $\mu$ s, and the timer duration is normally adjusted by setting the value of the Timer Duration Raw parameter. However, if you require a duration time that is longer than what you can achieve by changing the value of the Timer Duration Raw parameter alone, the Timer Duration Time Base Abs parameter can be used to change the duration time base.

The Timer Duration Time Base Abs parameter value sets the duration time base in  $\mu$ s. The default is 1  $\mu$ s and it can be changed in 1  $\mu$ s increments.

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

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

## Setting the Timer Duration with an Absolute Value

You can also set the timer 1 duration by using an "absolute" value. This is accomplished by setting the Timer Duration Abs parameter. The units for setting this parameter are  $\mu\text{s}$  and the value can be set in increments of 1  $\mu\text{s}$ .

To set the duration timer 1 using an absolute value:

- Use the Timer Selector to select timer 1.
- Set the value of the Timer Duration Abs parameter.

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

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );  
Camera.TimerDurationAbs.SetValue( 100 );
```

When you use the Timer Duration Abs parameter to set the duration time, the camera accomplishes the setting change by automatically changing the Timer Duration Raw parameter to achieve the value specified by the Timer Duration Abs setting. This leads to a limitation that you must keep in mind if you use Timer Duration Abs parameter to set the duration time. That is, you must set the Timer Duration Abs parameter to a value that is equivalent to a setting you could achieve by using the Timer Duration Raw and the current Timer Duration Base parameters. For example, if the time base was currently set to 50  $\mu\text{s}$ , you could use the Timer Duration Abs parameter to set the duration to 50  $\mu\text{s}$ , 100  $\mu\text{s}$ , 150  $\mu\text{s}$ , etc.

If you read the current value of the Timer Duration Abs parameter, the value will indicate the product of the Timer Duration Raw parameter and the Timer Duration Time Base. In other words, the Timer Duration Abs parameter will indicate the current duration time setting.

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

## 8.3 Checking the State of the I/O Lines

### 8.3.1 Checking the State of the Output Line

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

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

### 8.3.2 Checking the State of All Lines

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

- Read the value of the Line Status All parameter.

You can read the Line Status All parameter value from within your application software by using the 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 27, certain bits in the value are associated with each line and the bits will indicate the state of the lines. If a bit is 0, it indicates that



the state of the associated line is currently low. If a bit is 1, it indicates that the state of the associated line is current high.

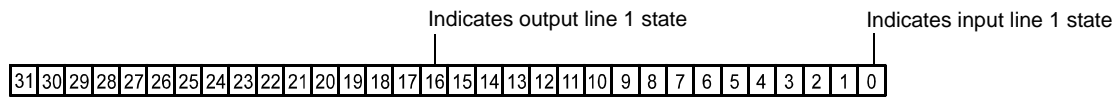


Fig. 27: Line Status All Parameter Bits



# 9 Standard Features

This section provides detailed information about the standard features available on each camera. It also includes an explanation of their operation and the parameters associated with each feature.

## 9.1 Gain

The camera's gain setting is adjustable. As shown in Figure 28, increasing the gain increases the slope of the response curve for the camera. This results in a higher gray value output from the camera for a given amount of output from the imaging sensor. Decreasing the gain decreases the slope of the response curve and results in a lower gray value for a given amount of sensor output.

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

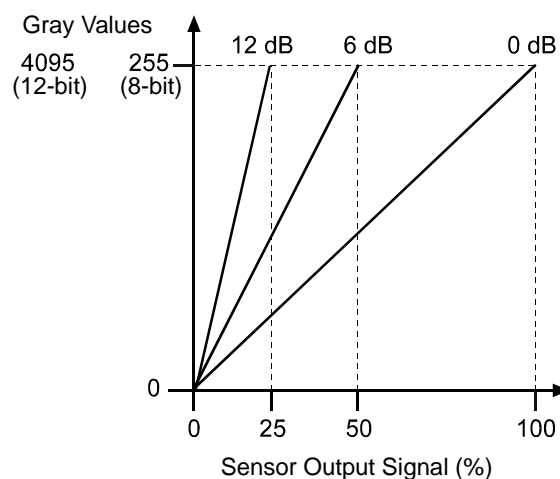


Fig. 28: Gain in dB

## Setting the Gain (All Models Except sIA750-60fm)



### Note

The information in this section applies to all camera models except the sIA750-60fm fm/fc. For information about sIA750-60fm cameras, see the next section.

The camera's gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on a decimal scale. The minimum decimal setting varies depending on the camera model (see Table 12). The maximum setting depends on whether the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed) or yields an effective pixel depth of 12 bits per pixel (Mono 16, Mono 12 Packed).

Camera Model	Min Setting	Max Setting (8 bit depth)	Max Setting (16 bit depth)
sIA1000-30 fm	360	1023	511
sIA1390-17fm	360	1023	511
sIA1600-14fm	350	1023	511

Table 12: Minimum and Maximum Allowed Gain Raw Settings

To set the Gain Raw parameter value:

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

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

```
Camera.GainSelector.SetValue( GainSelector_All );
Camera.GainRaw.SetValue( 400 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

If you know the current decimal setting for the gain raw, you can use the formulas below to calculate the dB of gain that will result from that setting.

For gain raw settings from 110 to 511:

$$\text{Gain}_{\text{dB}} = 20 \times \log_{10} \left( \frac{658 + \text{Gain Raw Setting}}{658 - \text{Gain Raw Setting}} \right) - G_c$$

For gain raw settings from 512 to 1023:

$$\text{Gain}_{\text{dB}} = (0.0354 \times \text{Gain Raw Setting}) - G_c$$

Where:

$$G_c = 20 \times \log_{10} \left( \frac{658 + \text{Min Gain Raw Setting}}{658 - \text{Min Gain Raw Setting}} \right)$$

### Example:

Assume that you are working with an sIA1390-17fm camera that is set for the Mono 8 pixel format and has a gain raw setting of 500. Calculating the gain is a two step process:

Step 1:

$$G_c = 20 \times \log_{10} \left( \frac{658 + 192}{658 - 192} \right)$$

$$G_c = 5.22 \text{ dB}$$

Step 2:

$$\text{Gain}_{\text{dB}} = 20 \times \log_{10} \left( \frac{658 + 500}{658 - 500} \right) - 5.22 \text{ db}$$

$$\text{Gain}_{\text{dB}} = 12.1 \text{ dB}$$

Table 13 shows the minimum and maximum gain in dB for each camera model.

Camera Model	dB Gain at Min Setting	dB Gain at Max Setting (8 bit depth)	dB Gain at Max Setting (16 bit depth)
sIA1000-30fm	0	25.5	7.3
sIA1390-17fm	0	25.5	7.3
sIA1600-14fm	0	25.9	7.7

Table 13: Minimum and Maximum dB of Gain

## Setting the Gain (sIA750-60fm Only)



### Note

The information in this section only applies to sIA750-60fm cameras. For information about the other camera models, see the previous section.

The camera's gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on a decimal scale. The range for the Gain Raw parameter setting is from 0 to 22.

To set the Gain Raw parameter value:

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

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

```
Camera.GainSelector.SetValue( GainSelector_All );  
Camera.GainRaw.SetValue( 20 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

If you know the current decimal setting for the gain raw, you can use the following formula to calculate the dB of gain that will result from that setting:

$$\text{Gain}_{\text{dB}} = 20 \times \log_{10} \left( 1 + \frac{\text{Gain Raw Setting}}{6} \right)$$

**Example:**

Assume that you are working with an sIA750-60fm camera that has a gain raw setting of 18. The gain is calculated as follows:

$$\text{Gain}_{\text{dB}} = 20 \times \log_{10} \left( 1 + \frac{18}{6} \right)$$

$$\text{Gain}_{\text{dB}} = 12.0 \text{ dB}$$

Table 14 shows the dB of gain that will be achieved at various Gain Raw settings.

Gain Setting	dB Gain
0	0
5	5.3
10	8.5
15	10.9
20	12.7
22	13.4

Table 14: dB of Gain at Various Settings

## 9.2 Black Level

Adjusting the camera's black level will result in an offset to the pixel values output by the camera. Increasing the black level setting will result in a positive offset in the digital values output for the pixels. Decreasing the black level setting will result in a negative offset in the digital values output for the pixels.

### Effect on All Camera Models Except the sIA750-60fm

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

If the camera is set for a pixel data format that yields an effective pixel depth of 12 bits per pixel (Mono 16, Mono 12 Packed, Bayer BG 16, Bayer RG 16, Bayer BG 12 Packed), an increase of 1 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. A decrease of 1 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

### Effect on sIA750-60fm Models

An increase of 4 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. And a decrease of 4 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

## Setting the Black Level

The black level can be adjusted by changing the value of the Black Level Raw parameter. The Black Level Raw parameter value can range from 0 to 255 on all camera models except the sIA750-60fm. On sIA750-60fm cameras, the parameter value can range from 0 to 64.

To set the Black Level Raw parameter value:

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

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

```
Camera.BlackLevelSelector.SetValue ( BlackLevelSelector_All );  
Camera.BlackLevelRaw.SetValue( 32 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).



## 9.3 Digital Shift



### Note

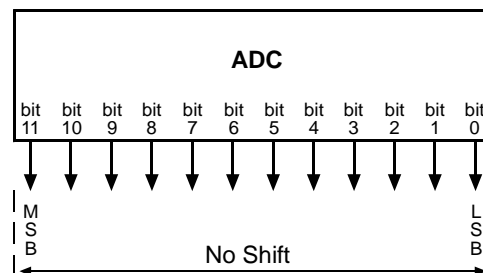
The information in this section applies to all camera models except the sIA750-60fm.

The digital shift feature lets you change the group of bits that is output from the ADC in the camera. Using the digital shift feature will effectively multiply the output of the camera by 2 times, 4 times, 8 times, or 16 times. The next two sections describe how the digital shift works when the camera is set for a 12 bit pixel format and when it is set for a 8 bit pixel format. There is also a section describing precautions that you must observe when using the digital shift feature and a section that describes enabling and setting the digital shift feature.

### 9.3.1 Digital Shift with 12 Bit Pixel Formats

#### No Shift

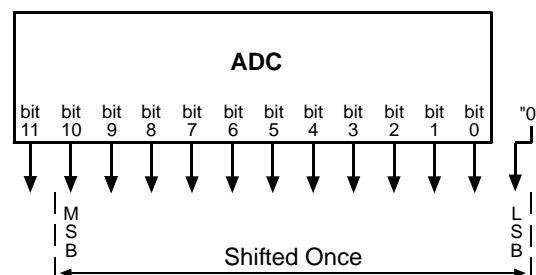
As mentioned in the Functional Description section of this manual, the camera uses a 12 bit ADC to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 12 bit effective depth, by default, the camera transmits the 12 bits that are output from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



#### Shift by 1

When the camera is set to shift by 1, the output from the camera will include bit 10 through bit 0 from the ADC along with a zero as an LSB.

The result of shifting once is that the output of the camera is effectively multiplied by 2. For example, assume that the camera is set for no shift, that it is viewing a uniform white target, and that under these conditions the reading for the brightest pixel is 100. If you changed the digital shift setting to shift by 1, the reading would increase to 200.



When the camera is set to shift by 1, the least significant bit output from the camera for each pixel value will be 0. This means that no odd gray values can be output and that the gray value scale will

only include values of 2, 4, 6, 8, 10, and so on. This absence of some gray values is commonly referred to as "missing codes".

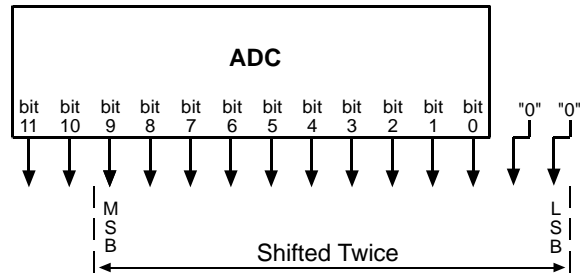
If the pixel values being output by the camera's sensor are high enough to set bit 11 to 1, we recommend not using shift by 1. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 1 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 2048.

### Shift by 2

When the camera is set to shift by 2, the output from the camera will include bit 9 through bit 0 from the ADC along with 2 zeros as LSBs.

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

When the camera is set to shift by 2, the 2 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 4th value, for example, 4, 8, 16, 20, and so on.



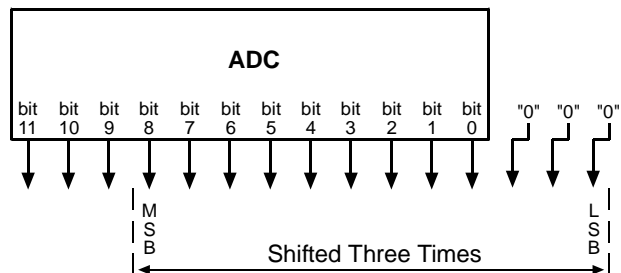
If the pixel values being output by the camera's sensor are high enough to set bit 10 or bit 11 to 1, we recommend not using shift by 2. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 2 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 1024.

### Shift By 3

When the camera is set to shift by 3, the output from the camera will include bit 8 through bit 0 from the ADC along with 3 zeros as LSBs.

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

When the camera is set to shift by 3, the 3 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 8th gray value, for example, 8, 16, 24, 32, and so on.



If the pixel values being output by the camera's sensor are high enough to set bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 3. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 3 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 512.

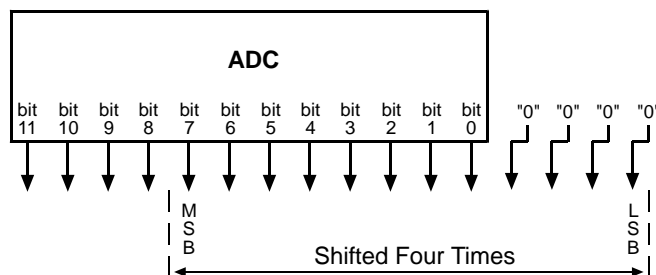
### Shift By 4

When the camera is set to shift by 4, the output from the camera will include bit 7 through bit 0 from the ADC along with 4 zeros as LSBs.

The result of shifting 4 times is that the output of the camera is effectively multiplied by 16.

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

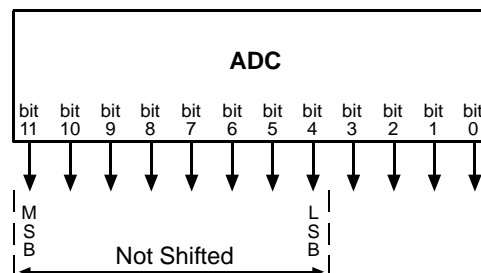
If the pixel values being output by the camera's sensor are high enough to set bit 8, bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 4. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 4 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 256.



## 9.3.2 Digital Shift with 8 Bit Pixel Formats

### No Shift

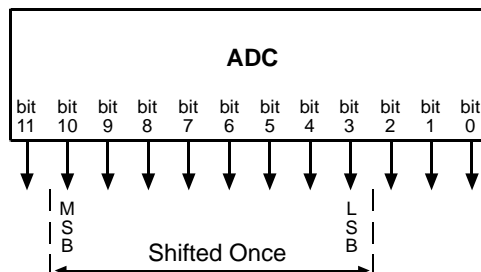
As mentioned in the Functional Description section of this manual, the camera uses a 12 bit ADC to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 8 bit effective depth, by default, the camera drops the 4 least significant bits from the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



### Shift by 1

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

The result of shifting once is that the output of the camera is effectively multiplied by 2. For example, assume that the camera is set for no shift, that it is viewing a uniform white target, and that under these conditions the reading for the brightest pixel is 10. If you changed the digital shift setting to shift by 1, the reading would increase to 20.



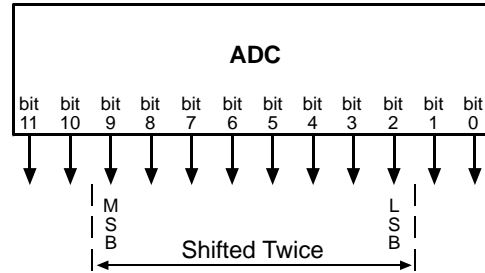
If the pixel values being output by the camera's sensor are high enough to set bit 11 to 1, we recommend not using shift by 1. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 1 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 128.

### Shift by 2

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

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

If the pixel values being output by the camera's sensor are high enough to set bit 10 or bit 11 to 1, we recommend not using shift by 2. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 2 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 64.

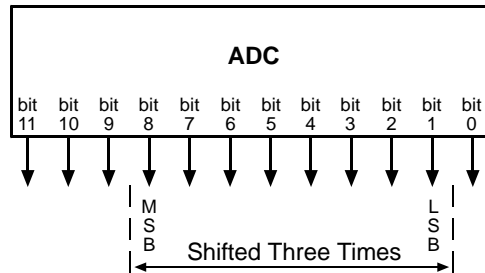


### Shift by 3

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

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

If the pixel values being output by the camera's sensor are high enough to set bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 3. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, that you should only use the shift by 3 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 32.

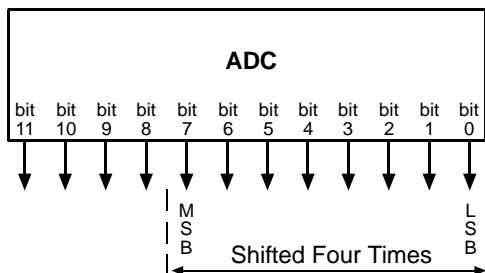


### Shift by 4

When the camera is set to shift by 4, the output from the camera will include bit 7 through bit 0 from the ADC.

The result of shifting four times is that the output of the camera is effectively multiplied by 16.

If the pixel values being output by the camera's sensor are high enough to set bit 8, bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 4. If you do nonetheless, all bits output from the camera will



automatically be set to 1. Therefore, you should only use the multiply by 4 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 16.

### 9.3.3 Precautions When Using Digital Shift

There are several checks and precautions that you must follow before using the digital shift feature. The checks and precautions differ depending on whether the camera will be set for a 12 bit pixel format or for an 8 bit pixel format in your application.

#### **If you will be using a 12 bit pixel format, make this check:**

Use the pylon Viewer or the pylon API to set the camera for a 12 bit pixel format and **no digital shift**.

Check the output of the camera under your normal lighting conditions and note the readings for the brightest pixels.

- If any of the readings are above 2048, do not use digital shift.
- If all of the readings are below 2048, you can safely use the shift by 1 setting.
- If all of the readings are below 1024, you can safely use the shift by 1 or 2 settings.
- If all of the readings are below 512, you can safely use the shift by 1, 2, or 3 settings.
- If all of the readings are below 256, you can safely use the shift by 1, 2, 3, or 4 settings.

#### **If you will be using an 8 bit format, make this check:**

Use the pylon Viewer or the pylon API to set the camera for a 8 bit pixel format and **no digital shift**.

Check the output of the camera under your normal lighting conditions and note the readings for the brightest pixels.

- If any of the readings are above 128, do not use digital shift.
- If all of the readings are below 128, you can safely use the shift by 1 setting.
- If all of the readings are below 64, you can safely use the shift by 1 or 2 settings.
- If all of the readings are below 32, you can safely use the shift by 1, 2, or 3 settings.
- If all of the readings are below 16, you can safely use the shift by 1, 2, 3, or 4 settings.

### 9.3.4 Enabling and Setting Digital Shift

You can enable or disable the digital shift feature by setting the value of the Digital Shift parameter. When the parameter is set to zero, digital shift will be disabled. When the parameter is set to 1, 2, 3, or 4, digital shift will be set to shift by 1, shift by 2, shift by 3, or shift by 4 respectively.

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

```
// Disable digital shift
Camera.DigitalShift.SetValue( 0 );
```

```
// Enable digital shift by 2  
Camera.DigitalShift.SetValue( 2 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 9.4 Area of Interest (AOI)

The area of interest (AOI) feature lets you specify a portion of the sensor array and after each image is acquired, only the pixel information from the specified portion of the array is transmitted to the host PC.

The area of interest is referenced to the top left corner of the sensor array. The top left corner is designated as column 0 and row 0 as shown in Figure 29.

The location and size of the area of interest is defined by declaring an X offset (coordinate), a width, a Y offset (coordinate), and a height. For example, suppose that you specify the x offset as 10, the width as 16, the y offset as 6, and the height as 10. The area of the array that is bounded by these settings is shown in Figure 29.

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

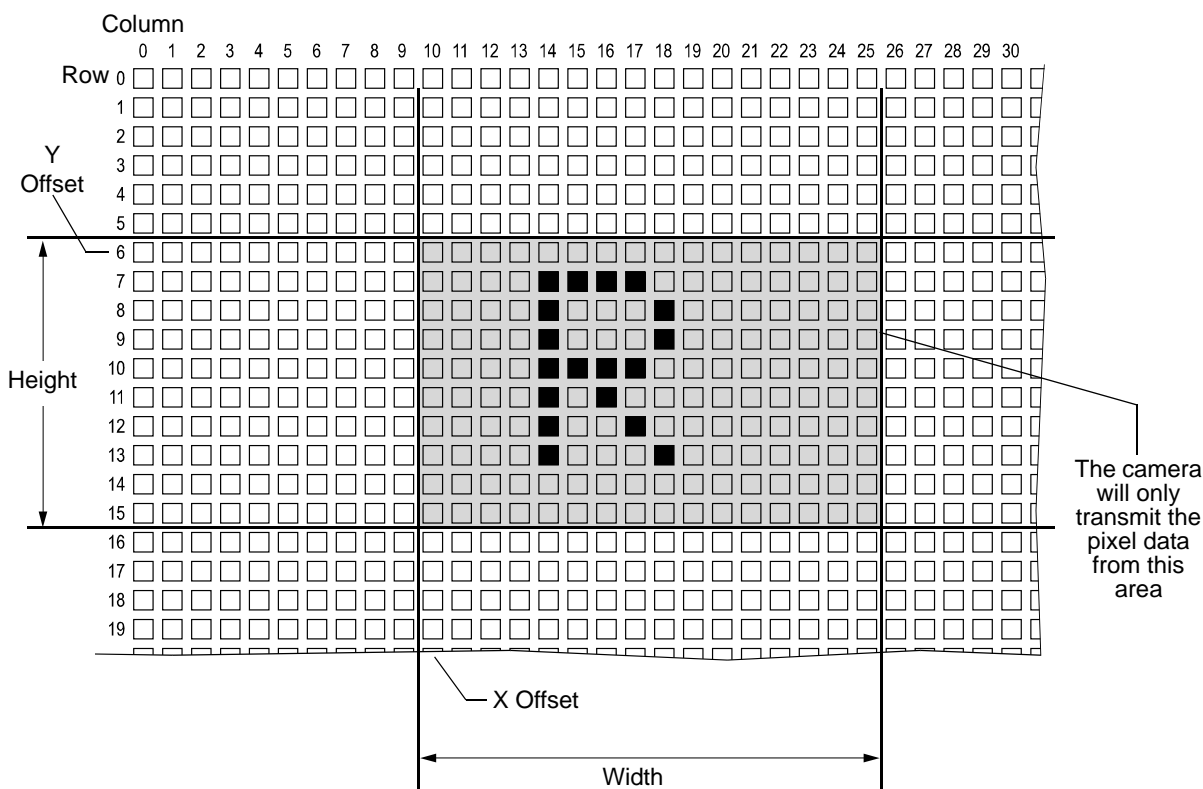


Fig. 29: Area of Interest

One of the main advantages of the AOI feature is that decreasing the height of the AOI can increase the camera's maximum allowed acquisition frame rate.

For more information about how changing the AOI height affects the maximum allowed frame rate, see Section 6.10 on [page 65](#).

## Setting the AOI

The AOI is set by default to use the full resolution of the camera's sensor.

You can change the size and the position of the AOI by changing the value of the camera's X Offset, Y Offset, Width, and Height parameters.

- The value of the X Offset parameter determines the starting column for the area of interest.
- The value of the Y Offset parameter determines the starting row for the area of interest.
- The value of the Width parameter determines the width of the area of interest.
- The value of the Height parameter determines the height of the area of interest.

When you are setting the camera's area of interest, you must follow these guidelines:

- The sum of the current X Offset setting plus the current Width setting must not exceed the width of the sensor in the camera model you are using. For example, on the sIA1000-30fm, the sum of the current X Offset setting plus the current Width setting must not exceed 1034.
- The sum of the current Y Offset setting plus the current Height setting must not exceed the height of the sensor in the camera model you are using. For example, on the the sIA1000-30fm, the sum of the current Y Offset setting plus the current Height setting must not exceed 779.
- The X Offset, Y Offset, Width, and Height parameters can be set in increments of 1.



You can set the X Offset, Y Offset, Width, and Height parameter values from within your application software by using the pylon API. The following code snippets illustrate using the API to get the maximum allowed settings and the increments for the Width and Height parameters. They also illustrate setting the X Offset, Y Offset, Width, and Height parameter values

```
int64_t widthMax = Camera.Width.GetMax( );
int64_t 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 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 9.4.1 Changing AOI Parameters "On-the-Fly"

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

## 9.5 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 30 shows a normal image on the left and an image captured with reverse X enabled on the right.

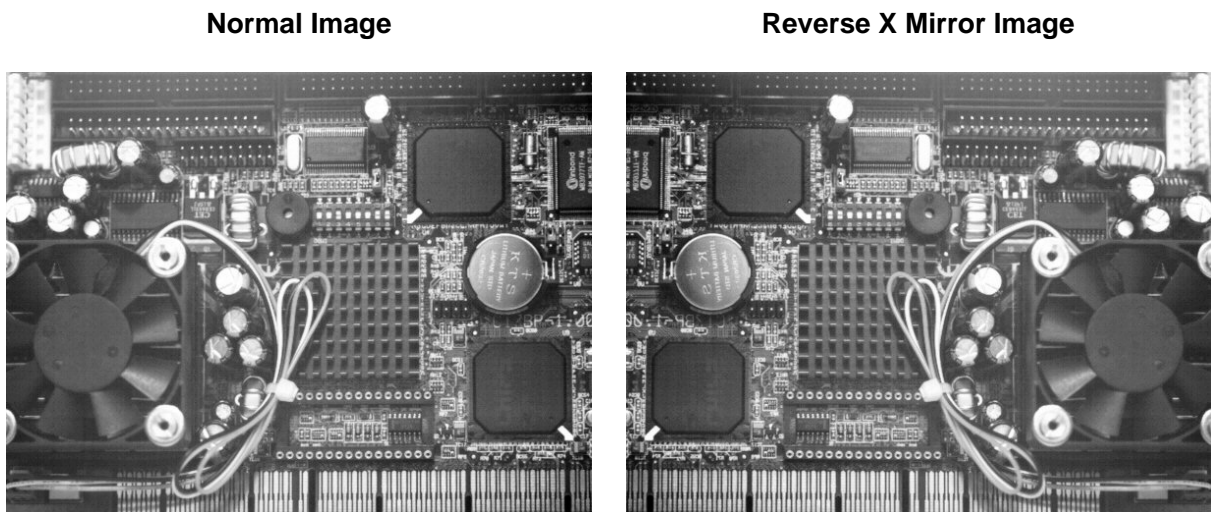


Fig. 30: Reverse X Mirror Imaging

### Using AOIs with Reverse X

You can use the AOI feature when using the reverse X feature. Note, however, that an AOI is always defined with respect to the pixels of the sensor. Therefore, the position of an AOI on the sensor remains the same regardless of whether or not the reverse X feature is enabled.

As a consequence, an AOI will display different images depending on whether or not the reverse X feature is enabled.

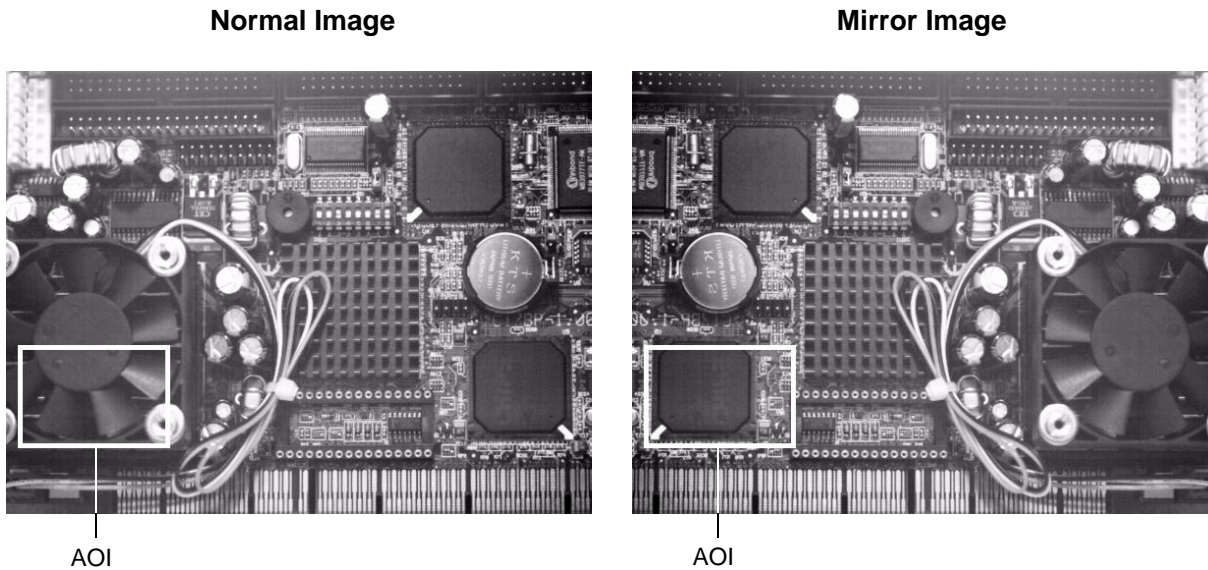


Fig. 31: Using an AOI with Reverse X Mirror Imaging

## Setting Reverse X

You can enable or disable the reverse X feature by setting the `ReverseX` parameter value. You can set the parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
// Enable reverse X
Camera.ReverseX.SetValue(true);
```

For detailed information about using the pylon API, refer to the [Basler pylon Programmer's Guide and API Reference](#).

You can also use the [Basler pylon Viewer](#) application to easily set the parameter.

## 9.6 Disable Parameter Limits

For each camera parameter, the allowed range of parameter values normally is limited. The factory limits are designed to ensure optimum camera operation and, in particular, good image quality. For special camera uses, however, it may be helpful to set parameter values outside of the factory limits.

The disable parameter limits feature lets you disable the factory parameter limits for certain parameters. When the factory parameter limits are disabled, the parameter values can be set within extended limits. Typically, the range of the extended limits is dictated by the physical restrictions of the camera's electronic devices, such as the absolute limits of the camera's variable gain control.

The values for the extended limits can be seen using the Basler pylon Viewer or from within your application via the pylon API.



### Note

Currently, the parameter limits can only be disabled on the Gain feature.

### Disabling Parameter Limits

To disable the limits for a parameter:

- Use the Parameter Selector to select the parameter whose limits you wish to disable.
- Set the value of the Remove Limits parameter.

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

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

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


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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 9.7 Debouncer

The debouncer feature aids in discriminating between valid and invalid input signals and only lets valid signals pass to the camera. The debouncer value specifies the minimum time that an input signal must remain high or remain low in order to be considered a valid input signal.

	<p>We recommend setting the debouncer value so that it is slightly greater than the longest expected duration of an invalid signal.</p> <p>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.</p>
---	---

Note that the debouncer delays a valid signal between its arrival at the camera and its transfer. The duration of the delay will be determined by the debouncer value.

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

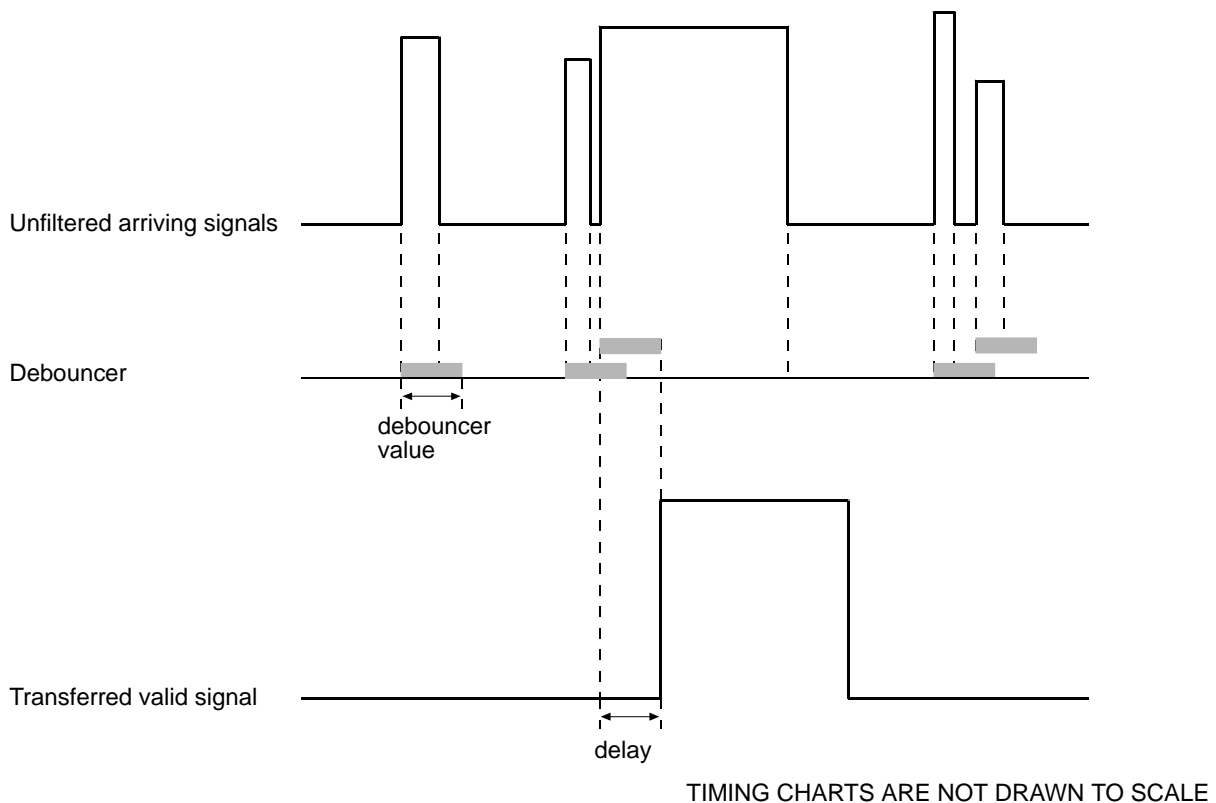


Fig. 32: Filtering of Input Signals by the Debouncer

## Setting the Debouncer

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

To set the debouncer:

- Use the Line Selector to select input line1.
- Set the value of the Line Debouncer Time Abs parameter.

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

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

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

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 9.8 Trigger Delay

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

The trigger delay may be specified in the range from 0 to 100000001  $\mu$ s (equivalent to 10 s). When the delay is set to 0  $\mu$ s, no delay will be applied.

The trigger delay will not operate when the camera is triggered by your application software and when the camera operates in continuous frame mode (free run).

### Setting the Trigger Delay

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

```
// Trigger delay
double TriggerDelay_us = 1000.0 // 1000us == 1ms == 0.001s;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );
```

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

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



## 9.9 Acquisition Status

When controlling image acquisition with a software trigger you can use the acquisition status feature to determine when the camera is ready to be triggered for an image acquisition. Using this feature, you can avoid triggering the camera at a rate that exceeds the maximum allowed with the current camera settings.



### Note

It is not possible to monitor the status of the Acquisition Start command. Therefore, you can not use the status of the Acquisition Start command to determine when the camera is ready to be triggered for an image acquisition.

### Determining the Acquisition Status

To determine the acquisition status of the camera:

- Use the Acquisition Status Selector to select the Frame Trigger Wait status.
- Read the value of the AcquisitionStatus parameter. If the value is set to "false", the camera is not ready to receive a software trigger, if the value is set to "true", the camera is ready to receive a software trigger.

You can set the Acquisition Status Selector and read the AcquisitionStatus parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set and read the parameter values:

```
// Set the Acquisition Status Selector
Camera.AcquisitionStatusSelector.SetValue(
AcquisitionStatusSelector_FrameTriggerWait );

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

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

You can also use the Basler pylon Viewer application to easily set the Acquisition Status Selector.

## 9.10 Test Images

All cameras include the ability to generate test images. Test images are used to check the camera's basic functionality and its ability to transmit an image to the host PC. Test images can be used for service purposes and for failure diagnostics. For test images, the image is generated internally by the camera's logic and does not use the optics, the imaging sensor, or the ADC. 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 cameras digital features will also have no effect on the transmitted images. But for test images 4 and 5, the cameras digital features will affect the images transmitted by the camera. This makes test images 4 and 5 a good way to check the effect of using a digital feature.

### Enabling a Test Image

The Test Image Selector is used to set the camera to output a test image. You can set the value of the Test Image Selector to one of the test images or to "test image off".

You can set the Test Image Selector from within your application software by using the pylon API. The following code snippets illustrate using the API to set the selector:

```
// set for no test image
Camera.TestImageSelector.SetValue( TestImageSelector_Off );

// set for the first test image
Camera.TestImageSelector.SetValue( TestImageSelector_Testimage1 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## Test Image 1 - Fixed Diagonal Gray Gradient (8 bit)

The 8 bit fixed diagonal gray gradient test image is best suited for use when the camera is set for monochrome 8 bit output. The test image consists of fixed diagonal gray gradients ranging from 0 to 255.

If the camera is set for 8 bit output and is operating at full resolution, test image one will look similar to Figure 33.

The mathematical expression for this test image:

$$\text{Gray Value} = [\text{column number} + \text{row number}] \text{ MOD } 256$$

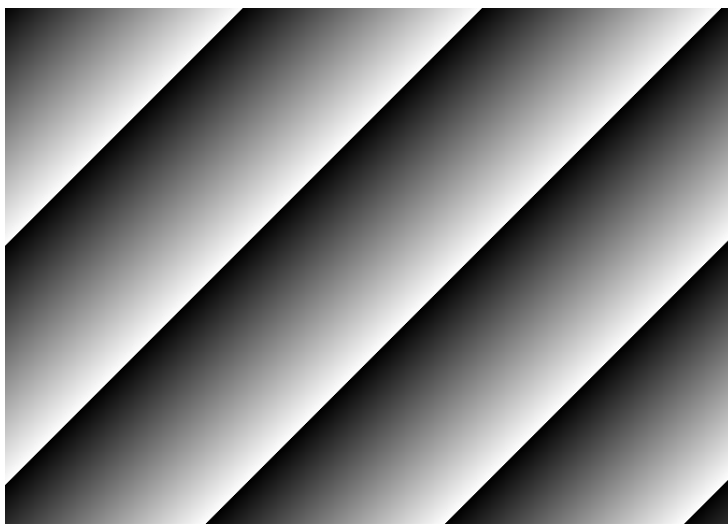


Fig. 33: Test Image One

## Test Image 2 - Moving Diagonal Gray Gradient (8 bit)

The 8 bit moving diagonal gray gradient test image is similar to test image 1, but it is not stationary. The image moves by one pixel from right to left whenever a new image acquisition is initiated. The test pattern uses a counter that increments by one for each new image acquisition.

The mathematical expression for this test image is:

$$\text{Gray Value} = [\text{column number} + \text{row number} + \text{counter}] \text{ MOD } 256$$

### 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 increments by one for each new image acquisition.

The mathematical expression for this test image is:

$$\text{Gray Value} = [\text{column number} + \text{row number} + \text{counter}] \text{ MOD } 4096$$



#### Note

On sIA750-60fm cameras, test image 3 is a 10 bit pattern. Since these cameras do not have a 10 bit output mode available, use of test image 3 on sIA750-60fm cameras is not recommended.

### Test Image 4 - Moving Diagonal Gray Gradient Feature Test (8 bit)

The basic appearance of test image 4 is similar to test image 2 (the 8 bit moving diagonal gray gradient image). The difference between test image 4 and test image 2 is this: if a camera feature that involves digital processing is enabled, test image 4 **will** show the effects of the feature while test image 2 **will not**. This makes test image 4 useful for checking the effects of digital features such as the luminance lookup table.

### Test Image 5 - Moving Diagonal Gray Gradient Feature Test (12 bit)

The basic appearance of test image 5 is similar to test image 3 (the 12 bit moving diagonal gray gradient image). The difference between test image 5 and test image 3 is this: if a camera feature that involves digital processing is enabled, test image 5 **will** show the effects of the feature while test image 3 **will not**. This makes test image 5 useful for checking the effects of digital features such as the luminance lookup table.



#### Note

On sIA750-60fm cameras, test image 5 is a 10 bit pattern. Since these cameras do not have a 10 bit output mode available, use of test image 5 on sIA750-60fm cameras is not normally recommended.

## 9.11 Device Information Parameters

Each camera includes a set of "device information" parameters. These parameters provide some basic information about the camera. The device information parameters include:

- Device Vendor Name (read only) - contains the name of the camera's vendor. For scout cameras, this string will always indicate Basler as the vendor.
- Device Model Name (read only) - contains the model name of the camera, for example, sIA1000-30fm.
- Firmware Version (read only) - contains the version of the firmware in the camera.
- Device ID (read only) - contains the serial number of the camera.
- Device Scan Type (read only) - contains the scan type of the camera, for example, area scan.
- Sensor Width (read only) - contains the physical width of the sensor in pixels.
- Sensor Height (read only) - contains the physical height of the sensor.
- Max Width (read only) - Indicates the camera's maximum area of interest (AOI) width setting.
- Max Height (read only) - Indicates the camera's maximum area of interest (AOI) height setting.

You can read the values for all of the device information parameters or set the value of the Device ID parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to read the parameters or write the Device 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 Firmware Version parameter
Pylon::String_t firmwareVersion = Camera.DeviceFirmwareVersion.GetValue();

// Write and read the Device ID
Camera.DeviceID = "custom name";
Pylon::String_t deviceID = Camera.DeviceID.GetValue();

// Read the Sensor Width parameter
int64_t sensorWidth = Camera.SensorWidth.GetValue();

// Read the Sensor Height parameter
int64_t sensorHeight = Camera.SensorHeight.GetValue();

// Read the Max Width parameter
int64_t maxWidth = Camera.WidthMax.GetValue();

// Read the Max Height parameter
int64_t maxHeight = Camera.HeightMax.GetValue();
```

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

You can also use the Basler pylon Viewer application to easily read the parameters and to read or write the Device ID.

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 9.12 Configuration Sets

A configuration set is a group of values that contains all of the parameter settings needed to control the camera. There are three basic types of configuration sets: the active configuration set, the default configuration set, and user configuration sets.

### Active Configuration Set

The active configuration set contains the camera's current parameter settings and thus determines the camera's performance, that is, what your image currently looks like. When you change parameter settings using the pylon API or the pylon Viewer, you are making changes to the active configuration set. The active configuration set is located in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The active configuration set is usually called the "active set" for short.

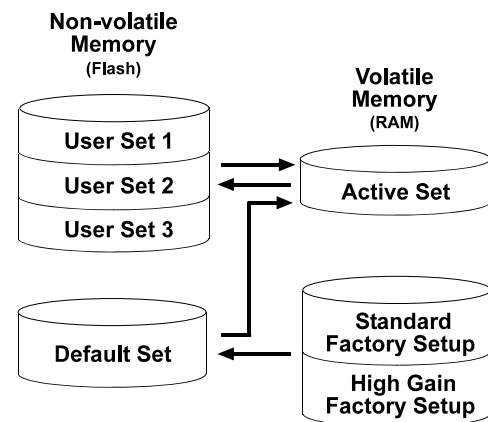


Fig. 34: Configuration Sets

### Default Configuration Set

When a camera is manufactured, numerous tests are performed on the camera and two factory optimized setups are determined. The two factory optimized setups are:

- The Standard Factory Setup - is optimized for average conditions and will provide good camera performance in many common applications. In the standard factory setup, the gain is set to a low value, and all auto functions are set to off.
- The High Gain Factory Setup - is similar to the standard factory setup, but the gain is set to + 6 dB.

The factory setups are saved in permanent files in the camera's non-volatile memory. They are not lost when the camera is reset or switched off and they cannot be changed.

You can select one of the two factory setups to be the camera's "default configuration set". Instructions for selecting which factory setup will be used as the default set appear below. Note that your selection of which factory setup will serve as the default set will not be lost when the camera is reset or switched off.

The default configuration set can be loaded into the active set. The default configuration set can also be selected as the camera's startup set. Instructions for loading the default set into the active set and for selecting the startup set appear below.

### User Configuration Sets

As mentioned above, the active configuration set is stored in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The camera can save most of the

settings from the current active set to a reserved area in the camera's non-volatile memory. A configuration set saved in the non-volatile memory is not lost when the camera is reset or switched off. There are three reserved areas in the camera's non-volatile memory available for saving configuration sets. A configuration set saved in a reserved area is commonly referred to as a "user configuration set" or "user set" for short.

The three available user sets are called User Set 1, User Set 2, and User Set 3.

**Note**

The settings for the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If used, these settings must be set again after each camera reset or restart.

## Startup Set

You can select the default configuration set or one of the user configuration sets stored in the camera's non-volatile memory to be the "startup set." The configuration set that you have selected as the startup set will automatically be loaded into the active set whenever the camera starts up at power on or after a reset. Instructions for selecting the startup set appear below.

## 9.12.1 Saving User Sets

Saving the current active set into a user set in the camera's non-volatile memory is a three step process:

- Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
- Set the User Set Selector to User Set 1, User Set 2, or User Set 3.
- Execute a User Set Save command to save the active set to the selected user set.

Saving an active set to a user set in the camera's non-volatile memory will overwrite any parameters that were previously saved in that user set.

You can set the User Set Selector and execute the User Set Save command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet1 );  
Camera.UserSetSave.Execute( );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).



## 9.12.2 Selecting a Factory Setup as the Default Set

When the camera is delivered, the Standard Factory Setup will be selected as the default configuration set. You can, however, select either one of the two factory setups to serve as the default set.

To select which factory setup to serve as the default set:

- Set the Default Set Selector to the Standard Factory Setup or the High Gain 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);
```



### Note

Selecting which factory setup will serve as the default set is only allowed when the camera is idle, i.e. when it is not acquiring images continuously or does not have a single image acquisition pending.

Selecting the Standard Factory Setup as the default set and then loading the default set into the active set is a good course of action if you have grossly misadjusted the settings in the camera and you are not sure how to recover. The standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

## 9.12.3 Loading a Saved Set or the Default Set into the Active Set

If you have saved a configuration set into the camera's non-volatile memory, you can load the saved set from the camera's non-volatile memory into the camera's active set. When you do this, the loaded set overwrites the parameters in the active set. Since the settings in the active set control the current operation of the camera, the settings from the loaded set will now be controlling the camera.

You can also load the default set into the camera's active set.

To load a saved configuration set or the default set from the camera's non-volatile memory into the active set:

- Set the User Set Selector to User Set 1, User Set 2, User Set 3 or Default.
- Execute a User Set Load command to load the selected set into the active set.

You can set the User Set Selector and execute the User Set Load command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet2 );  
Camera.UserSetLoad.Execute( );
```



### Note

Loading a user set or the default set into the active set is only allowed when the camera is idle, i.e. when it is not acquiring images continuously or does not have a single image acquisition pending.

Loading the Default Set with the Standard Factory Setup selected into the active set is a good course of action if you have grossly misadjusted the settings in the camera and you are not sure how to recover. The standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

## 9.12.4 Selecting the Startup Set

You can select the default configuration set (i.e., whichever was selected as the default configuration set, either the Standard Factory Setup or the High Gain Factory Setup) or one of the user configuration sets stored in the camera's non-volatile memory to be the "startup set". The configuration set that you designate as the startup set will be loaded into the active set whenever the camera starts up at power on or after a reset.

The User Set Default Selector is used to select the startup set:

- Set the User Set Default Selector to User Set 1, User Set 2, User Set 3 or Default.

You can set the User Set Default Selector from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector:

```
Camera.UserSetDefaultSelector.SetValue( UserSetDefaultSelector_Default );
```



# 10 Using Multiple Cameras on a Single Bus and Managing Bandwidth

This section includes information about using multiple cameras on a single IEEE 1394 bus.

## 10.1 Using Multiple Cameras Where All Devices are 1394b

Most of the information included in this manual assumes that you have a single camera attached to your IEEE 1394b bus. But is it also quite common to attach more than one camera to a single bus. One of the main advantages of the IEEE 1394 bus architecture is that it is designed to handle multiple devices (such as cameras) connected to a single bus. And the connected devices can share the available bandwidth on the bus.

One way you can manage two cameras on a single bus is to operate the cameras so that only one camera is transmitting an image at any given time. In this situation, the camera transmitting images can use 100% of the bus bandwidth. In many situations, however, you would like to have two (or more) cameras transmitting images at the same time. In this case, the cameras that are transmitting images simultaneously must share the available bus bandwidth.

To understand how bandwidth is shared on an IEEE 1394 bus, we need to look at a few bus architecture basics. The IEEE 1394b bus operates on a 125 microsecond cycle. During each cycle, the bus can carry a single packet from one device with a packet size up to 8192 bytes. As an alternative, the bus can carry several packets from different devices where the sum of the packet sizes is 8192 bytes or less. These two situations are illustrated in Figure 35.

In situation 1 shown in the figure, a single camera (camera A) is attached to the bus and we want that camera to use 100% of the bandwidth available during each bus cycle. In this case, we would set the camera so that it would put 8192 bytes into the packet that it sends on each cycle of the bus.

In situation 2, we have three cameras (cameras A, B, and C) attached to the bus and all three cameras will transmit image data simultaneously. We want camera A to use 25%, of the available bus bandwidth, camera B to use 25%, and camera 3 to use 50%. In this case, we would set camera A so that it would put 2048 bytes, i.e., 25% of the 8192 byte maximum, into each packet it sends. We would set camera B so that it would put 2048 bytes (25% of 8192) into each packet it sends. And we would set camera C so that it would put 4096 bytes (50% of the 8192) into each packet it sends. As shown in the figure, the bus carries the packets sequentially on each cycle. The total byte load in all of the packets combined is 8192 and is equal to the maximum allowed per cycle. Note

that we could have made the packet sizes smaller and thus the total byte load per cycle would be less than the maximum allowed. It is OK to make the total byte load smaller than the maximum, but not larger.

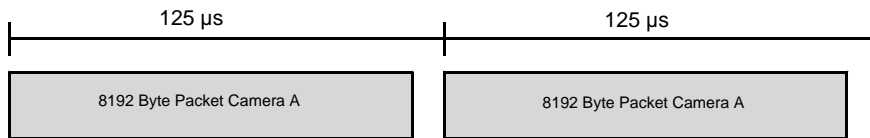
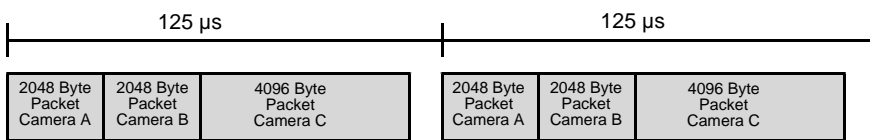
**Situation 1:****Situation 2:**

Fig. 35: Packet Transmission During Bus Cycles

A parameter called the Packet Size is used to set the size of the packet that the camera will transmit on each cycle of the bus. For an IEEE 1394b camera attached to an IEEE 1394b bus, the minimum value for this setting is 1 byte and the maximum is 8192 bytes. If you set the packet size to 8192, the camera will use 100% of the available bus bandwidth when it is transmitting images. If you set the bytes per packet to a lower value, the camera will use less of the bandwidth. For example, if you set the value to 5120 (62.5% of 8192), then the camera will send 5120 byte packets when it is transmitting image data and will use 62.5% of the available bus bandwidth.

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

```
// Set packet size  
Camera.PacketSize.SetValue( 4096 );
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

## 10.2 Using Multiple Cameras Where 1394a and 1394b Devices are Mixed

The descriptions in the previous section assume that all of the devices on the bus are IEEE 1394b devices. If the bus has mixed IEEE 1394a devices and IEEE 1394b devices, determining how to share bandwidth between devices is a bit more difficult. As a first step toward understanding the situation, consider the difference between 1394a devices and 1394b devices:

- A 1394a device can transmit at what is known as S400 speed (400 Mbit/s).
- During a single bus cycle, a device operating at S400 speed can transmit a single packet of up to 4096 bytes. Alternatively, several devices operating at S400 speed can transmit packets during a single bus cycle as long as the sum of the bytes in the packets is 4096 bytes or less.
- A 1394b device can transmit at what is known as S800 speed (800 Mbit/s).
- During a single bus cycle, a device operating at S800 speed can transmit a single packet of up to 8192 bytes. Alternatively, several devices operating at S800 speed can transmit packets during a single bus cycle as long as the sum of the bytes in the packets is 8192 bytes or less.

The next thing that we must consider in a bus that has mixed 1394a and 1394b devices is the speed at which each device will transmit:

- A 1394a device will always be capable of transmitting at S400 speed on a mixed 1394a / 1394b bus. (The 1394a device can transmit at slower speeds, but we are assuming that you always want to transmit at the fastest speed.)
- A 1394b device will transmit at S800 speed if all of the devices in its path to the host PC, including the adapter card in the PC, are 1394b devices. If the path to the PC passes through any 1394a device, then the 1394b device will transmit at S400 speed.

Figure 36 illustrates some situations where 1394a devices and 1394b devices are mixed on a single bus. If you look at the figure, you will notice:

- Camera 1 will transmit image data at S400 speed. This is simply because the camera itself is a 1394a device and S400 is the maximum speed for 1394a devices.
- Camera 2 will transmit image data at S400 speed. The camera is a 1394b device, which means that it is capable of S800 speed. But the camera's path to the host PC passes through a 1394a adapter, so this limits the camera's actual maximum speed to S400.
- Camera 3 will transmit image data at S800 speed. This is because the camera is a 1394b device and its path to the host PC passes through only 1394b devices.
- Camera 4 will transmit image data at S400 speed. This is simply because the camera itself is a 1394a device.
- Camera 5 will transmit image data at S400 speed. The camera is a 1394b device, but its path to the host PC passes through a 1394a hub, so this limits the camera's maximum speed to S400.
- Camera 6 will transmit image data at S400 speed. This is simply because the camera itself is a 1394a device.
- Camera 7 will transmit image data at S800 speed. This is because the camera is a 1394b device and its path to the host PC passes through only 1394b devices.

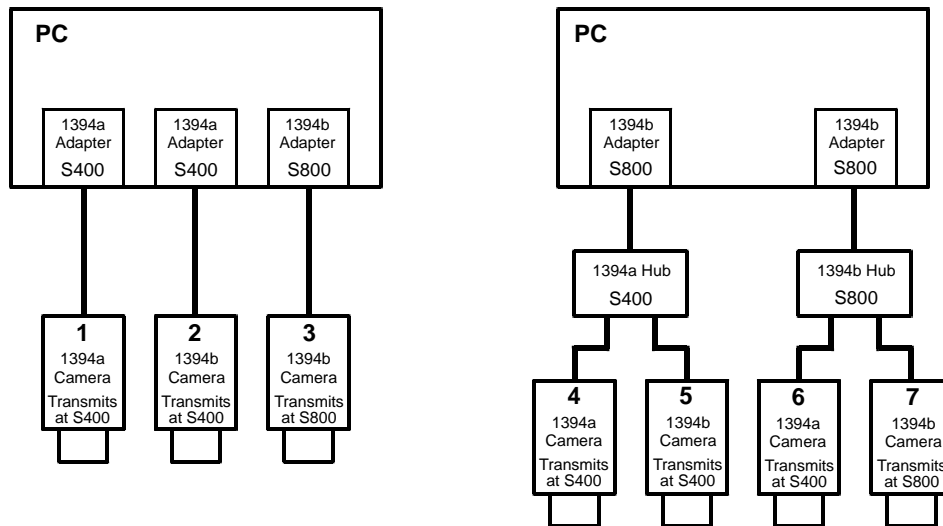


Fig. 36: Examples of Mixed Device Types

**Note**

The driver will always set each device to operate at the fastest possible speed for the current network configuration. This behavior of the driver can't be changed by the user.

So what does all of this mean when we are trying to share bandwidth between devices operating at different speeds on the same bus? Some examples will provide the best explanation.

Example 1: Assume that you have two cameras on the bus and that you want them to capture and transmit images simultaneously. Camera 1 is operating at S400 speed and the camera 2 is operating at S800 speed. Also assume that you want camera one to use 40% of the available bandwidth and camera 2 to use 60%. How would you set the packet size on the cameras so that each one would use the desired portion of the bandwidth available in each bus cycle?

For camera 1, the calculation would be:

$$0.40 \times 4096 = 1638.4 \text{ bytes per packet}$$

(the packet size must be set to a multiple of 4, so we would round the setting down to 1636)

For camera 2, the calculation would be:

$$0.60 \times 8192 = 4915.2 \text{ bytes per packet}$$

(the packet size must be set to a multiple of 4, so we would round the setting down to 4912)

So in this case, you would set the packet size for camera 1 to 1636 bytes and for camera 2 to 4912 bytes.



You may be asking why we multiply the percentage for camera 1 by 4096 and the percentage for camera 2 by 8192. The reason is:

During the part of the bus cycle when the packet for camera 1 is transmitted, the bus will operate at S400 speed. At S400, the maximum number of bytes that can be transmitted in a bus cycle is 4096.

During the part of the bus cycle when the packet for camera 2 is transmitted, the bus will operate at S800 speed. At S800, the maximum number of bytes that can be transmitted in a bus cycle is 8192.

Example 2: Assume that you have three cameras on the bus and that you want these camera to capture and transmit images simultaneously. Camera one is operating at S800 speed and is set for a packet size of 4200 bytes. Camera two is operating at S800 speed and is set for a packet size of 1800 bytes. Camera 3 is operating at S400 speed and is set for a packet size of 1000 bytes. How much of the available bandwidth would each camera use?

For camera 1, the calculation would be:

$$4200 / 8192 = 51.3\%$$

For camera 2, the calculation would be:

$$1800 / 8192 = 22.0\%$$

For camera 3, the calculation would be:

$$1000 / 4096 = 24.4\%$$

If you add these three results together, you find that 97.7% of the available bandwidth is being used. Keep in mind that if the sum was greater than 100%, you would need to lower the packet size setting for one or more of the cameras.

## 10.2.1 Recommended Packet Size

When you change the value of the packet size setting on a camera, there is something that you must keep in mind. If you lower the packet size setting, the camera takes longer to transmit each acquired image. And if you lower the packet size enough, it will begin to restrict the maximum frame rate that the camera can achieve. A read only parameter called the Recommended Packet Size can help you avoid this problem.

The recommended packet size parameter indicates the lowest value you can use for the packet size setting without restricting the camera's maximum allowed frame rate. Assume, for example, that you checked the packet size parameter value and you found it to be 2400. This would mean that if you set the camera's packet size to 2400 bytes or more, the camera's maximum allowed frame rate would not be affected by the packet size setting. And if you set the packet size lower than 2400, the camera's maximum allowed frame rate would be affected. The farther below 2400 you set the packet size, the more restricted the maximum frame rate would be.

You read the value of the Recommended Packet Size parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to get the parameter values:

```
// RecommendedPacketSize
int64_t recommendedPacketSize = Camera.RecommendedPacketSize.GetValue();
```

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

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

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

For more information about the camera's maximum allowed frame rate and how it can be restricted by the packet size setting, see Section 6.10 on [page 65](#).

# 11 Troubleshooting and Support

This section outlines the resources available to you if you need help working with your camera. It also provides some basic troubleshooting information that you can use to solve problems.

## 11.1 Tech Support Resources

The troubleshooting resources in this section of the manual will help you to find the cause of many common problems. If you need more assistance, you can contact the Basler technical support team for your area. Basler technical support contact information is located in the front pages of this manual.

If you do decide to contact Basler technical support, please take a look at the form that appears on the last two pages of this section before you call. Filling out this form will help make sure that you have all of the information the Basler technical support team needs to help you with your problem.

You will also find helpful information such as frequently asked questions, downloads, and application notes on the Basler website at:

[www.baslerweb.com/indizes/beitrag\\_index\\_en\\_22089.html](http://www.baslerweb.com/indizes/beitrag_index_en_22089.html)

## 11.2 Obtaining an RMA Number

Whenever you want to return material to Basler, you must request a Return Material Authorization (RMA) number before sending it back. The RMA number **must** be stated in your delivery documents when you ship your material to us! Please be aware that if you return material without an RMA number, we reserve the right to reject the material.

You can find detailed information about how to obtain an RMA number on the Basler website at:

[www.baslerweb.com/beitraege/beitrag\\_en\\_79701.html](http://www.baslerweb.com/beitraege/beitrag_en_79701.html)

## 11.3 Troubleshooting with the Camera LED

If the camera boots up successfully, the LED on the back of the camera will light and will remain green continuously.

If an error condition is detected, the LED will begin to flash. The number of flashes indicates the detected error as shown in Table 15.

LED State	Status Indication
Off	No power to the camera
Continuous green	The camera is OK.
Continuous red	Internal error. Contact Basler technical support.

Table 15: LED Indications

## 11.4 Troubleshooting Charts

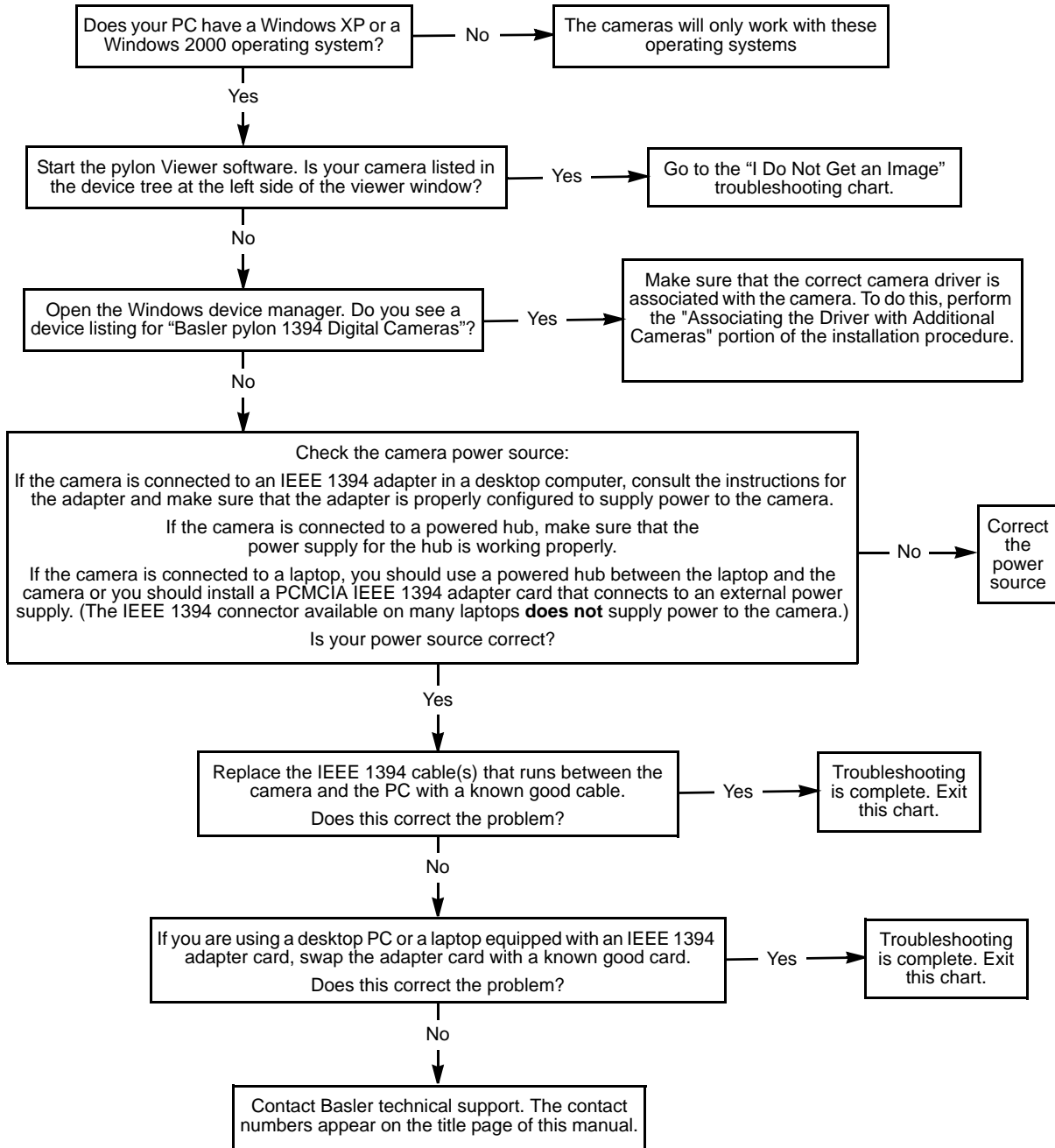
The following pages contain several troubleshooting charts that can help you find the cause of problems users sometimes encounter. The charts assume that you are familiar with the camera's features and settings. If you are not, we suggest that you review the camera manual before you troubleshoot a problem.

The charts also assume that you have the pylon Viewer software installed on your host PC and that you are familiar with using the software.

For more information about the pylon Viewer, see Section 3.1 on [page 17](#).

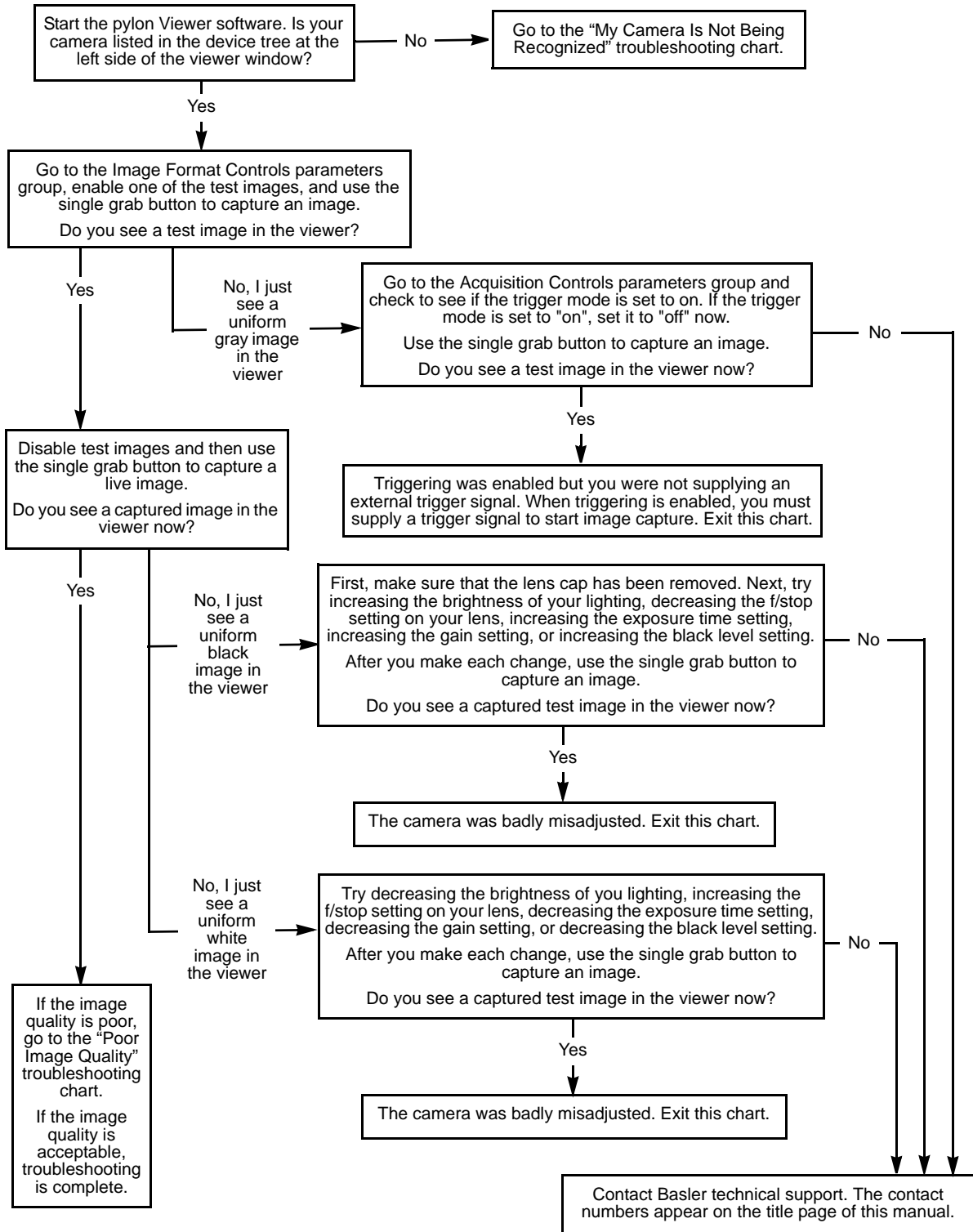
## 11.4.1 My Camera Is Not Being Recognized

Use this chart if your camera is connected to a PC, but is not being recognized by the PC.



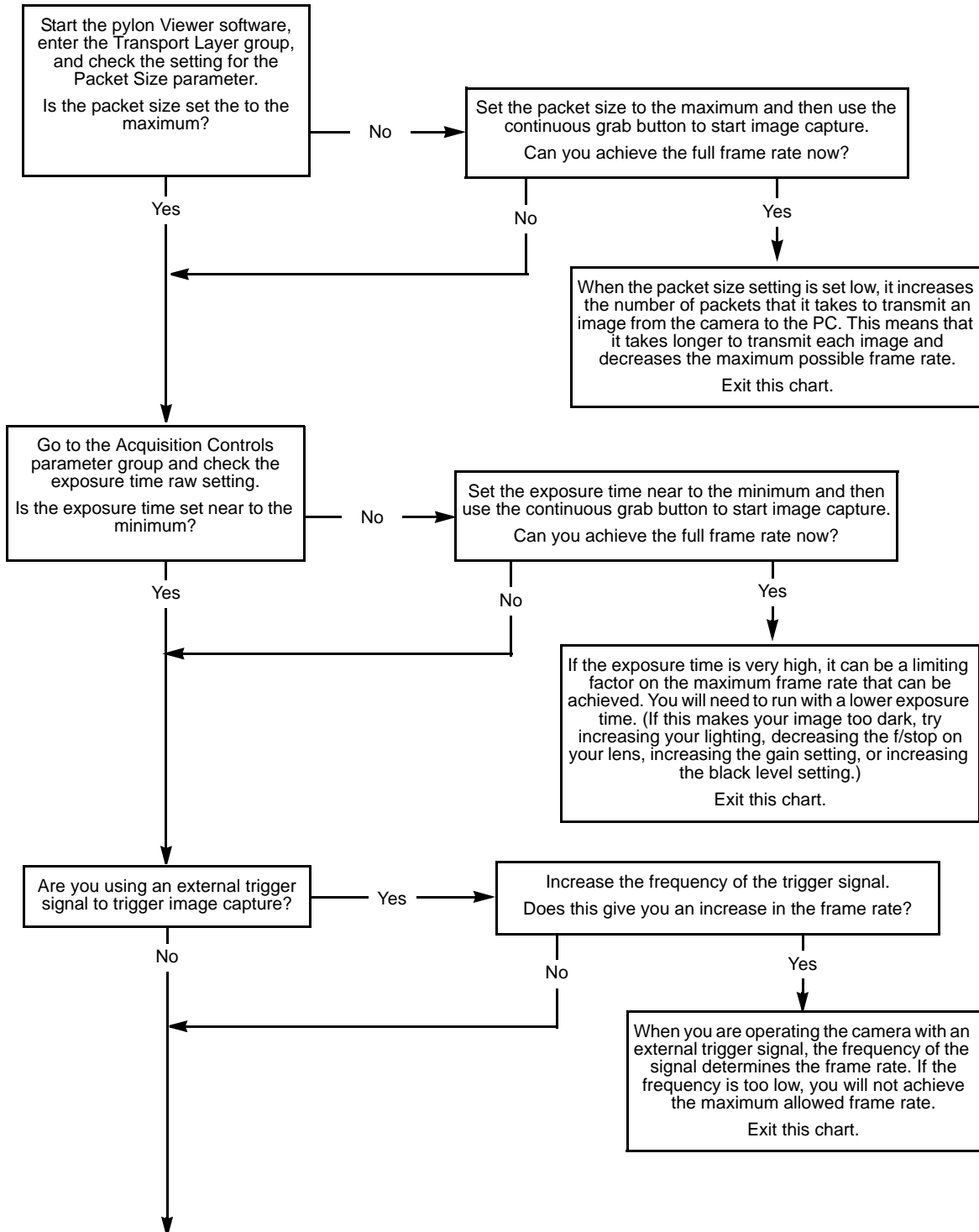
## 11.4.2 I Do Not Get an Image

Use this chart if you get no image at all when you attempt to capture an image. If you get a poor quality image, use the "Poor Image Quality" chart.

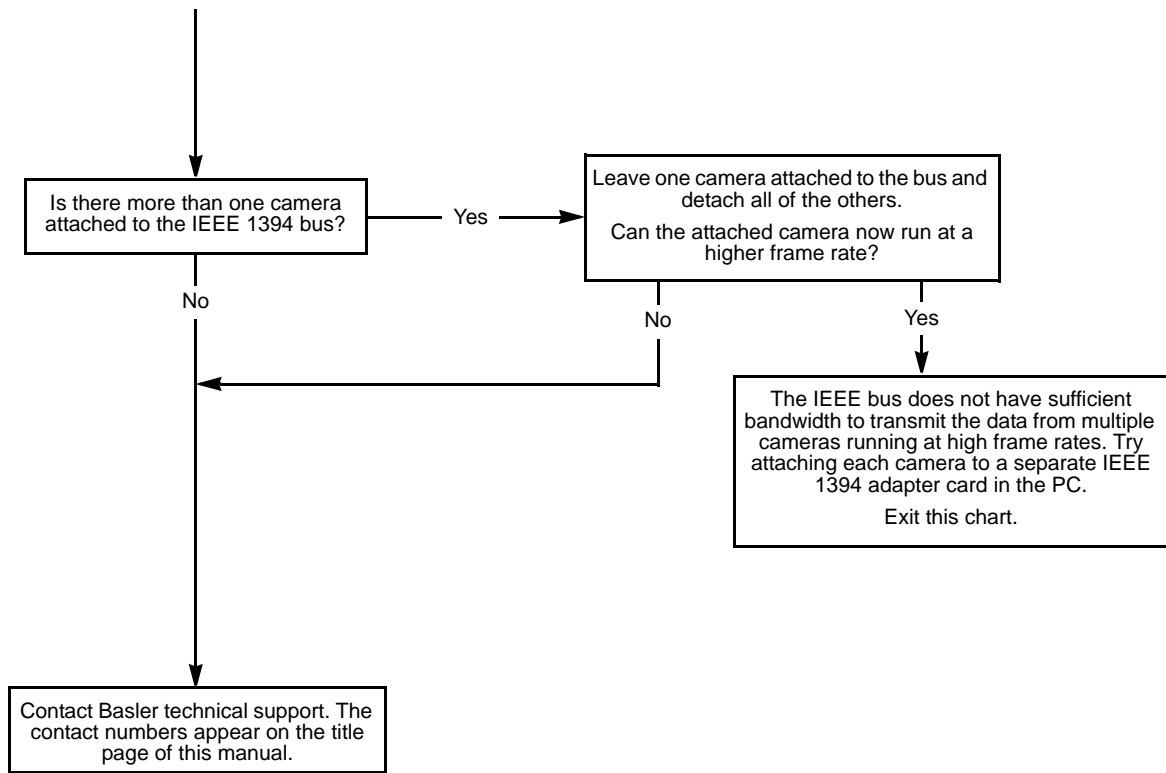


## 11.4.3 I Can Not Get the Full Frame Rate

Use this troubleshooting chart if you are attempting to run the camera at its maximum stated frame rate and you are not able to do so.

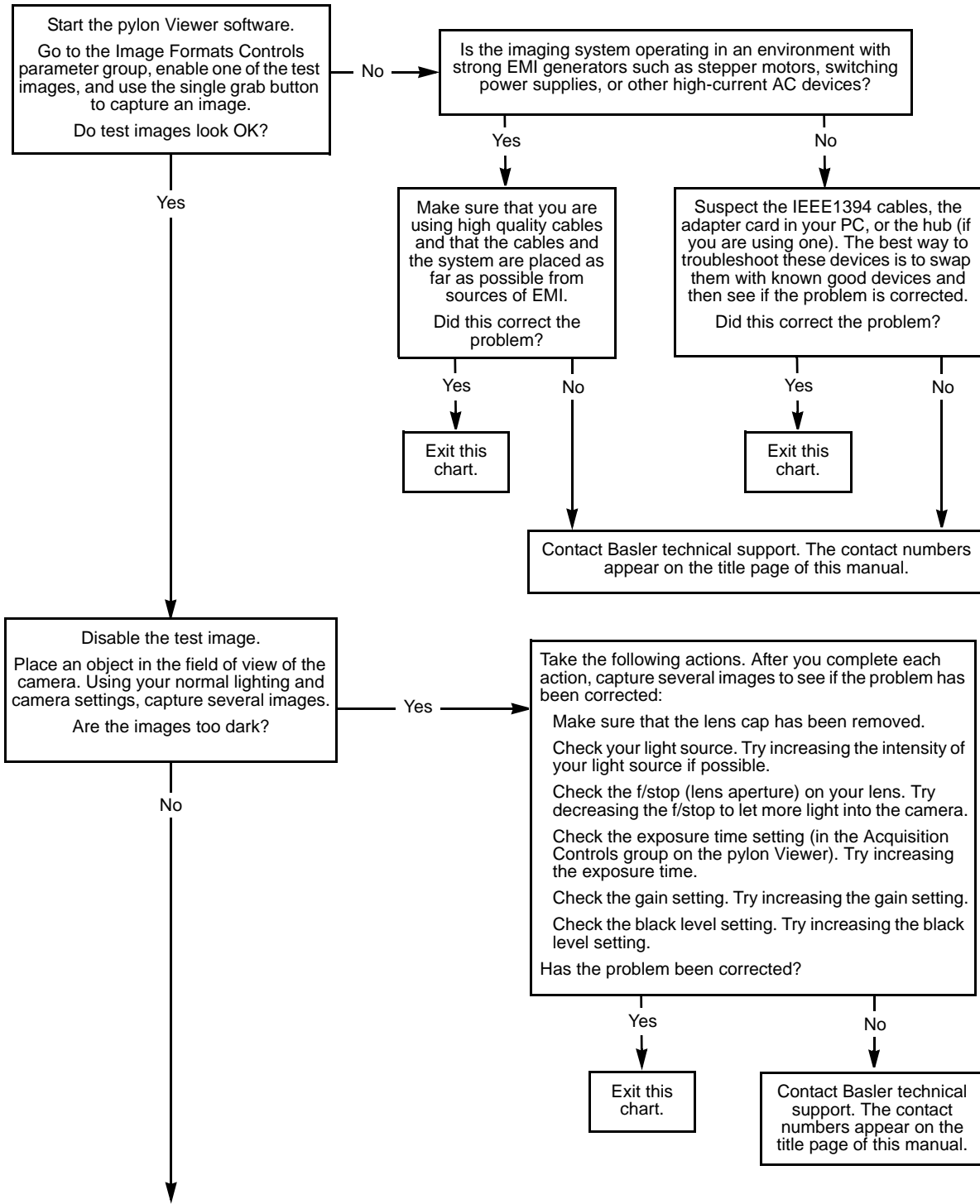


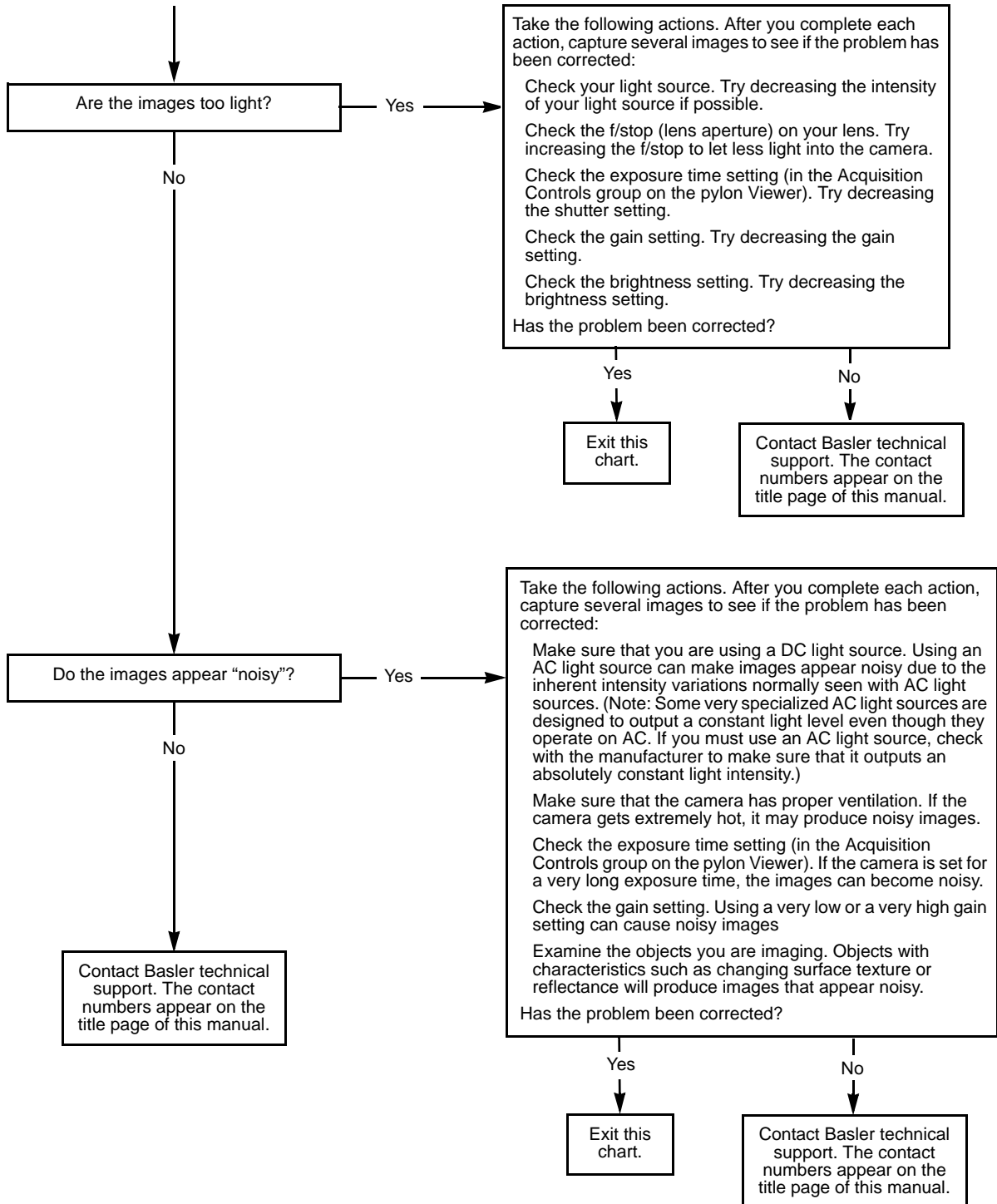




## 11.4.4 I Get Poor Image Quality

Use this chart if you can capture images, but they are poor quality. (If you can't capture images at all, use the "I Do Not Get an Image" troubleshooting chart.)





## 11.5 Before Contacting Basler Technical Support

To help you as quickly and efficiently as possible when you have a problem with a Basler camera, it is important that you collect several pieces of information before you contact Basler technical support.

Copy the form that appears on the next two pages, fill it out, and fax the pages to your local dealer or to your nearest Basler support center. Or, you can send an e-mail listing the requested pieces of information and with the requested files attached. Basler technical support contact information is shown in the title section of this manual.

- 1 The camera's product ID: \_\_\_\_\_
- 2 The camera's serial number: \_\_\_\_\_
- 3 1394 adapter that you use with the camera: \_\_\_\_\_  
\_\_\_\_\_
- 4 Describe the problem in as much detail as possible:  
(If you need more space, use an extra sheet of paper.)  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- 5 If known, what's the cause of the problem?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
  
- 6 When did the problem occur?  
 After start.                       While running.  
 After a certain action (e.g., a change of parameters):  
\_\_\_\_\_  
\_\_\_\_\_  
  
\_\_\_\_\_  
\_\_\_\_\_

7 How often did/does the problem occur?  Once.  Every time.  
 Regularly when:  
 \_\_\_\_\_  
 Occasionally when:  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

8 How severe is the problem?  Camera can still be used.  
 Camera can be used after I take this action:  
 \_\_\_\_\_  
 \_\_\_\_\_  
 Camera can no longer be used.

9 Did your application ever run without problems?  Yes  No

10 Parameter set

It is very important for Basler technical support to get a copy of the exact camera parameters that you were using when the problem occurred.

To make a copy of the parameters, use the dump register tool. To get the tool, go to: [www.baslerweb.com/beitraege/beitrag\\_en\\_19478.html](http://www.baslerweb.com/beitraege/beitrag_en_19478.html) (the tool is available for XP PCs only). Send the generated file to Basler technical support. Or, you can look up the settings with the pylon Viewer.

If you cannot access the camera, please try to state the following parameter settings:

Pixel format: \_\_\_\_\_  
 Packet size: \_\_\_\_\_  
 Exposure time: \_\_\_\_\_  
 Frame rate: \_\_\_\_\_

11 Live image/test image

If you are having an image problem, try to generate and save live images that show the problem. Also generate and save test images. Please save the images in BMP format, zip them, and send them to Basler technical support.



## Revision History

Doc. ID Number	Date	Changes
AW00075301000	25 Nov 2008	Initial release of this document.
AW00075302000	17 June 2009	<p>Updated sensor name "Micron MT9V022" to "Aptina MT9V022" in Section 1.2 on <a href="#">page 2</a>.</p> <p>Updated minimum allowed exposure times in Section 6.4 on <a href="#">page 51</a>.</p> <p>Added the digital shift feature in Section 9.3 on <a href="#">page 105</a>.</p> <p>Corrected the indications of x offset and y offset in Figure 29 in Section 9.4 on <a href="#">page 111</a>.</p> <p>Added the reverse X feature in Section 9.5 on <a href="#">page 115</a>.</p> <p>Added the trigger delay feature in Section 9.8 on <a href="#">page 120</a>.</p> <p>Added the acquisition status feature in Section 9.9 on <a href="#">page 121</a> and added a reference in Section 6.2.3 on <a href="#">page 42</a>.</p> <p>Removed the incorrect information about a Device User ID in Section 9.11 on <a href="#">page 125</a> and replaced it with the correct information about the Device ID parameter.</p> <p>Added the high gain factory setup and the standard factory setup (formerly the "default set") in Section 9.12 on <a href="#">page 127</a>.</p>





## Feedback

Your feedback will help us improve our documentation. Please click the link below to access an online feedback form. Your input is greatly appreciated.

<http://www.baslerweb.com/umfrage/survey.html>



## Index

### A

acquisition frame rate  
     and AOI size .....65, 70  
     maximum allowed .....65, 70  
 acquisition frame rate abs parameter  
     ..... 38, 42, 49  
 acquisition mode parameter  
     ..... 38, 41, 42, 48, 49  
 acquisition start command  
     ..... 38, 41, 42, 48, 49, 121  
 acquisition status .....121  
 active configuration set .....127  
 AOI  
     see area of interest  
 API .....17  
 area of interest  
     default resolution .....112  
     explained .....111  
     setting .....112

### B

bandwidth ..... 68, 69, 73  
     sharing with multiple cameras .....133  
 bit depth .....2, 3  
 black level  
     explained .....104  
     setting .....104  
 black level raw parameter .....104  
 black level selector .....104

### C

cables  
     IEEE 1394 .....28  
     standard I/O cable .....28  
 camera power requirements ..... 2, 3, 31  
 cleaning the camera and sensor .....14  
 code snippets, proper use .....13  
 configuration set loaded at startup .....131  
 configuration sets .....127–131  
 connector types .....27  
 connectors .....23, 27

### D

debouncer  
     and exposure start delay ..... 62  
     explained ..... 118  
     setting ..... 119  
     signal delay ..... 118  
 default configuration set ..... 127  
 device firmware version parameter ..... 125  
 device ID parameter ..... 125  
 device model name parameter ..... 125  
 device scan type parameter ..... 125  
 device vendor name parameter ..... 125  
 digital shift ..... 105  
 disable parameter limits  
     explained ..... 117

### E

electromagnetic interference ..... 11  
 electrostatic discharge ..... 11  
 EMI ..... 11  
 environmental requirements ..... 12  
 ESD ..... 11  
 exposure  
     controlling with an ExTrig signal ..... 44  
     overlapped ..... 54, 56  
 exposure active signal ..... 61  
 exposure mode  
     timed ..... 40, 45  
     trigger width ..... 45  
 exposure modes ..... 45  
 exposure start delay ..... 62  
 exposure time  
     maximum possible ..... 51  
     minimum allowed ..... 51  
     setting ..... 51  
 exposure time abs parameter ..... 53  
 exposure time base abs parameter ..... 52  
 exposure time raw parameter ..... 51  
 external trigger signal  
     controlling exposure with ..... 44  
     min high/low time ..... 44

**F**

factory setup .....	127
high gain factory setup .....	127
standard factory setup .....	127
frame rate	
controlling with a hardware trigger ....	44
controlling with a software trigger ....	42
maximum allowed .....	65, 70
setting with a parameter .....	38
frame readout time .....	63
frame transmission end time .....	63
free run .....	38
functional description .....	19

**G**

gain	
setting .....	100, 102

**H**

high gain factory setup .....	127
humidity .....	12

**I**

IEEE 1394b device information .....	30
input line	
configuring .....	87
electrical characteristics .....	33
installation	
hardware .....	15
software .....	15
integrate enabled signal .....	61
inverter	
output line .....	89
IP30 .....	7

**L**

LED .....	23, 140
lens adapter .....	2, 3
line inverter parameter .....	89
line selector .....	88
line source parameter .....	88
line status parameter .....	96

**M**

max height parameter .....	125
max width parameter .....	125
maximum acquisition frame rate .....	65, 70
mechanical stress tests .....	9
mirror imaging .....	115
models .....	1
mono 12 packed pixel format .....	80
mono 16 pixel format .....	78
mono 8 pixel format .....	76
multiple cameras on a bus .....	133

**O**

optical size of the sensor .....	2, 3
output line	
configuring .....	88
electrical characteristics .....	34
inverter .....	89
voltage requirements .....	34
output lines	
voltage requirements .....	34
overlapped exposure .....	54, 56

**P**

packet size .....	68, 73, 133, 136, 138
packet size parameter .....	68, 73
parameter sets .....	127
parameter sets, saving .....	128
parameters loaded at startup .....	131
pin assignments .....	24, 25
pixel data formats .....	75
YUV 422 (YUYV) packed .....	83
YUV 422 packed .....	82
pixel format parameter .....	75
pixel formats	
mono 12 packed .....	80
mono 16 .....	78
mono 8 .....	76
pixel size .....	2, 3
pixel transmission sequence .....	86
polarity, power .....	31
power consumption .....	31
precautions .....	13
programmable exposure mode	
with an external trigger signal .....	45
protection class .....	7
pylon API .....	17
pylon Viewer .....	17

**R**

recommended packet size parameter ...138  
 resulting frame rate abs parameter  
   ..... 65, 69, 70, 73  
 return material authorization .....139  
 reverse X .....115  
 RMA number .....139

**S**

s400 speed .....135  
 s800 speed .....135  
 saving parameter sets .....127, 128  
 sensor  
   CCD architecture .....20  
   CMOS architecture .....22  
   optical size .....2, 3  
   pixel size .....2, 3  
   size .....1, 2, 3  
   type .....2, 3  
 sensor height parameter .....125  
 sensor width parameter .....125  
 serial number .....14  
 sets of parameters, saving .....128  
 software development kit .....17  
 software trigger .....40  
 spectral response .....4–6  
 standard factory setup .....127, 130  
 startup parameter set .....131  
 startup set .....128, 131  
 stress tests .....9  
 support .....148

**T**

temperature .....12  
 test image selector .....122  
 test images .....122  
 time delay time base abs parameter .....92  
 timed exposure mode .....40, 45  
 timer delay .....93  
 timer delay abs parameter .....93  
 timer delay raw parameter .....92  
 timer delay time .....92  
 timer delay time base .....92  
 timer duration .....94  
 timer duration abs parameter .....95  
 timer duration raw parameter .....94  
 timer duration time base .....94  
 timer duration time base abs parameter ..94

timer selector ..... 91, 92, 94, 95  
 timer trigger source parameter ..... 91  
 transition threshold ..... 32  
 trigger delay ..... 120  
 trigger mode parameter ..... 37, 40, 47  
 trigger ready signal ..... 57  
 trigger selector parameter ..... 37, 40, 47  
 trigger software command ..... 41, 42  
 trigger source parameter ..... 40, 47  
 trigger width exposure mode ..... 45  
 troubleshooting  
   charts ..... 141  
   with the LED ..... 140

**U**

user configuration set ..... 127  
 user output selector ..... 89  
 user output value parameter ..... 89

**V**

ventilation ..... 12  
 viewer ..... 17  
 voltage, input ..... 31

**Y**

YUV 422 (YUYV) data range ..... 85  
 YUV 422 (YUYV) packed pixel data format  
   ..... 83  
 YUV 422 data range ..... 83  
 YUV 422 packed pixel data format ..... 82

