Basler dart

USER'S MANUAL FOR BCON CAMERAS

Document Number: AW001369
Version: 02   Language: 000 (English)
Release Date: 01 November 2016

FCC and CE conformity

Conformity tests have been performed on dart BCON cameras. The cameras have been found to comply with the CE and FCC requirements pursuant to EN 55022 and FCC Part 15, Subpart B rules.

Life support applications

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

Warranty Information

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

Do not remove the camera's product label

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

Do not remove the camera front

If you are using a dart S-mount or a dart CS-mount camera, do not remove the camera front. The camera front and the circuit board are firmly riveted. Both parts can be damaged if you remove the camera front.

Prevent contact with foreign substances

Do not allow e.g. liquid, flammable or metallic material to get in contact with the board. Otherwise, the camera may fail or cause a fire.

Avoid electromagnetic fields

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

Transport in original packaging

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

Clean with care

Follow the cleaning instructions in Chapter 2 on page 3.

Read the manual

Read the manual carefully before using the camera.

Contacting Basler Support Worldwide

Europe, Middle East, Africa

Basler AG
An der Strusbek 60–62
22926 Ahrensburg
Germany

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

support.europe@baslerweb.com

The Americas

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

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

support.usa@baslerweb.com

Asia-Pacific

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

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1 About this Document

This document contains important information about how to operate and configure Basler dart BCON cameras properly and most efficiently. It also contains information about designing your hardware to work with the Basler BCON interface.

Read these instructions before using Basler dart BCON cameras.

1.1 Target Audience

The Basler dart BCON User’s Manual is aimed at experienced hardware and software engineers proficient in electronics, software development, and embedded system design.

The document is written for a target audience that has intermediate to advanced technical skills in the areas mentioned above.

The document assumes that users have experience in the following areas:

- System on a Chip (SoC) or System on Module (SoM) architectures
- serial communication
- FPGA programming
- development of frame grabbing procedures
- bus architectures such as I²C
- Embedded Linux operating systems

1.2 Typographic Conventions

The following fonts are used in this manual to draw attention to program elements etc.:

Font	Element or Symbol	Example
Arial, bold	Program interface such as menu commands, windows, dialog boxes, field and button names	Press the **OK** button.
	Emphasis (importance)	Do **not** open the housing.
	Path to directories, file and directory names	Copy the files into the **Data\Statistics** folder.
`Consolas`	Source code	`camera.Gain.SetValue(10);`
`Consolas, italic`	Placeholder in source code	`SET relPath=`*`DllDir`*
Key symbol (if available)	Key names	Press the ↑ key.
UPPER case		Press the ESC key.
>	Succession of commands	Select **File** menu > **Open**.

1.3 Further Reading

For more information about the technologies and protocols used to operate Basler BCON cameras, refer to the following external documents:

- *I²C-Bus Specification and User Manual*:
  http://www.nxp.com/documents/user_manual/UM10204.pdf
- *LVDS Owner's Manual*:
  http://www.ti.com/lit/ml/snla187/snla187.pdf
- *FH41 Series* (Connector Data Sheet):
  http://www.hirose.co.jp/cataloge_hp/ed_FH41_20140305.pdf
- *Channel Link Design Guide:*
  http://www.ti.com/lit/ml/snla167/snla167.pdf
- *GenICam Pixel Format Names and Values:*
  http://www.emva.org/wp-content/uploads/GenICamPixelFormatValues.pdf

The documents are provided by the respective owners, and the links might be changed by them.

2 Precautions

⚠ DANGER

Electric Shock Hazard

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

You must use power supplies which meet the Safety Extra Low Voltage (SELV) and Limited Power Source (LPS) requirements.

⚠ WARNING

Fire Hazard

Unapproved power supplies may cause fire and burns.

You must use power supplies which meet the Limited Power Source (LPS) requirements.

NOTICE

Dust on the sensor can impair the camera's performance.

- Every time you handle the camera without a lens attached, make sure that the camera is pointing down so that no dust can reach the sensor.
- If the camera is not installed, store it in its original packaging.

NOTICE

Heat can damage the camera.

Make sure that you provide sufficient heat dissipation to keep the operation temperature of the device below the values indicated in Section 3.6.1 on page 21. For more information about providing heat dissipation, see Section 3.6.2 on page 22.

NOTICE

Incorrectly mounted lenses can damage camera components.

- When mounting a lens on the camera, do not overtighten the lens. Otherwise, the screw threads of the lens mount can be damaged.
- For dart **CS-mount** variants:
  - Make sure that the lens does not intrude into the camera body more than 7 mm (color cameras) or 11 mm (mono cameras). Otherwise, the IR cut filter (color cameras) or the sensor (mono cameras) can be damaged.
  - If you want to use C-mount lenses for dart CS-mount cameras, make sure that you attach a CS- to C-mount adapter ring to the C-mount lens before mounting it on a dart CS-mount camera. Otherwise, the lens may reach too far into the camera and damage the IR cut filter (color cameras) or the sensor (mono cameras).
- For dart **S-mount** variants:
  - On S-mount cameras, the lens is screwed in and out to reach the desired level of focus. S-mount lenses do not have a defined flange and therefore will not stop before they touch (and possibly scratch or break) the sensor glass.
  - Do not screw in the lens deeper than 11.7 mm, especially during focusing.

For more information, see Section 3.4.2 on page 16.

NOTICE

Voltage outside of the specified range can cause damage.

- You must supply camera power that complies with the individual voltage requirements of the BCON interface lines:
  - The nominal voltage for the power supply line (VCC) is **5 V**.
  - The nominal voltage for the I²C interface lines (I2C_SCL, I2C_SDA, I2C_ID) is **3.3 V**.
  - For more information, see Section 4.3 on page 27.
- The dart BCON cameras must only be connected to other limited power sources (LPS) / Safety Extra Low Voltage (SELV) circuits that do not represent any energy hazards.

NOTICE

Electrostatic discharge (ESD) can damage the sensor and the circuit board.

- Use anti-static clothes and materials, e.g. conductive shoes, anti-static gloves, and ESD protection wrist straps to decrease the risk of electrostatic discharge.
- Use conductive materials and install conductive mats at the point of installation (e.g. floor, workplace) to prevent the generation of static electricity.
- Control the humidity in your environment. Low humidity can cause ESD problems.

NOTICE

Incorrect cleaning can damage camera components.

- Before cleaning, disconnect the camera from camera power by removing the connector.
- After the cleaning procedure, make sure the cleaning material has evaporated before you reconnect the plugs.
- For dart **bare board**, dart **S-mount** and dart **CS-mount mono** variants, i.e. for dart BCON cameras where the sensor is accessible:
  - Avoid cleaning the surface of the camera's sensor.
    If you must clean it, use a soft, lint-free cloth dampened with a small quantity isopropanol.
  - Use a cloth that will not generate static charge during cleaning (cotton is a good choice). Electrostatic discharge might damage the sensor.
- Cleaning the IR cut filter of dart **CS-mount color** variants (the sensor cannot be cleaned):
  - Try not to touch the IR cut filter and do not clean the IR cut filter mechanically. The glass of the IR cut filter can break if you apply too much pressure.
  - Use clean, oil-free compressed air to clean the IR cut filter. Be careful not to apply too much air pressure.
- Do not use solvents or thinners to clean the board or the camera front or both. They can damage the surface.

NOTICE

Conductive contact can damage the circuit board.

Whenever you work with the camera, make sure that the circuit board has no conductive contact with other objects. Conductive contact can cause short circuit or overvoltage damage.

3 Technical Specifications and Requirements

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

3.1 Models

The current Basler dart BCON camera models are listed in the top row of the specification tables on the next pages of this manual. The camera models are differentiated by their resolution, their maximum frame rate at full resolution, and whether the camera's sensor is mono or color.

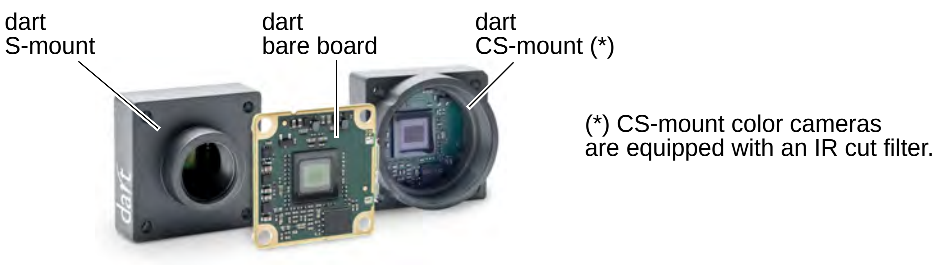


Fig. 1: dart BCON Variants

All dart BCON camera models are available in three variants:

- **Bare board**: This variant consists of a circuit board only.
- **S-mount**: This variant consists of a circuit board with a camera front attached. S-mount lenses can be attached to the lens mount on the camera front.
- **CS-mount**: This variant consists of a circuit board with a camera front attached. CS-mount lenses can be attached to the lens mount on the camera front.

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, to a subset of models, or a variant, will be so designated.

> The dart S-mount color cameras are not equipped with an IR cut filter. If you want to operate a dart S-mount color camera with an IR cut filter, you must attach a lens with an integrated IR cut filter to the camera.

3.2 General Specifications

Specification	daA1280-54bm/bc
Resolution (H x V pixels)	1280 x 960
Sensor Type	Aptina AR0134 Progressive scan CMOS Global shutter
Optical Size	1/3"
Effective Sensor Diagonal	6 mm
Pixel Size (H x V)	3.75 µm x 3.75 µm
Max. Frame Rate (at full resolution)	54 fps
Mono/Color	Mono or color (color cameras include a Bayer pattern RGB filter on the sensor)
Pixel Formats	Mono models: Mono 8 Mono 12 Color models: Bayer 8 Bayer 12 RGB 8 YCbCr422
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	Programmable via the camera API
Camera Power Requirements	Nominal +5 VDC, supplied via the camera's 28-pin FFC connector ≈ 1.3 W (typical) @ 5 VDC
I/O Lines	1 input line (CC0) and 2 output lines (two bits on data lane X0)
Lens Mount	S-mount, CS-mount, without mount (bare board)
Size (L x W x H)	Bare board model: 4.8 mm x 27 mm x 27 mm S-mount model: 17.5 mm x 29 mm x 29 mm CS-mount model: 16.9 mm x 29 mm x 29 mm
Weight	S-mount and CS-mount model: < 15 g Bare board model: < 5 g
Conformity	CE (includes RoHS), GenICam 2.4 (including PFNC 1.1 and SFNC 2.1), UL (in preparation) The EU Declaration of Conformity is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 5.0.4 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM)

Table 1: General Specifications (daA1280-54bm/bc)

Specification	daA1600-60bm/bc
Resolution (H x V pixels)	1600 x 1200
Sensor Type	e2v EV76C570 Progressive scan CMOS Global shutter
Optical Size	1/1.8"
Effective Sensor Diagonal	9 mm
Pixel Size (H x V)	4.5 µm x 4.5 µm
Max. Frame Rate (at full resolution)	60 fps
Mono/Color	Mono or color (color cameras include a Bayer pattern RGB filter on the sensor)
Pixel Formats	Mono models: Mono 8 Mono 12 (*) Color models: Bayer 8 Bayer 12 (*) RGB 8 YCbCr422 (*) 12-bit image data based on 10-bit sensor data. See Section 10.1 on page 80.
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	Programmable via the camera API
Camera Power Requirements	Nominal +5 VDC, supplied via the camera’s 28-pin FFC connector
	≈ 1.3 W (typical) @ 5 VDC
I/O Lines	1 input line (CC0) and 2 output lines (two bits on data lane X0)
Lens Mount	S-mount, CS-mount, without mount (bare board)
Size (L x W x H)	Bare board model: 5.9 mm x 27 mm x 27 mm S-mount model: 17.5 mm x 29 mm x 29 mm CS-mount model: 16.9 mm x 29 mm x 29 mm
Weight	S-mount and CS-mount model: < 15 g Bare board model: < 5 g
Conformity	CE (includes RoHS), GenICam 2.4 (including PFNC 1.1 and SFNC 2.1), UL (in preparation) The EU Declaration of Conformity is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 5.0.4 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM)

Table 2: General Specifications (daA1600-60bm/bc)

Specification	daA2500-14bm/bc
Resolution (H x V pixels)	2592 x 1944
Sensor Type	Aptina MT9P031 Progressive scan CMOS Rolling shutter
Optical Size	1/2.5"
Effective Sensor Diagonal	7.13 mm
Pixel Size (H x V)	2.2 µm x 2.2 µm
Max. Frame Rate (at full resolution)	14 fps
Mono/Color	Mono or color (color cameras include a Bayer pattern RGB filter on the sensor)
Pixel Formats	Mono models: Mono 8 Mono 12 (*) Color models: Bayer 8 Bayer 12 (*) RGB 8 YCbCr422
Synchronization	Via hardware trigger, via software trigger, or free run
Exposure Time Control	Via hardware trigger or programmable via the camera API
Camera Power Requirements	Nominal +5 VDC, supplied via the camera's 28-pin FFC connector
	≈ 1.4 W (typical) @ 5 VDC
I/O Lines	1 input line (CC0) and 2 output lines (two bits on data lane X0)
Lens Mount	S-mount, CS-mount, without mount (bare board)
Size (L x W x H)	Bare board model: 4.8 mm x 27 mm x 27 mm S-mount model: 17.5 mm x 29 mm x 29 mm CS-mount model: 16.9 mm x 29 mm x 29 mm
Weight	S-mount and CS-mount models: < 15 g Bare board model: < 5 g
Conformity	CE (includes RoHS), GenICam 2.4 (including PFNC 1.1 and SFNC 2.1), UL (in preparation) The EU Declaration of Conformity is available on the Basler website: www.baslerweb.com
Software	Basler pylon Camera Software Suite (version 5.0.4 or higher) Available for Windows (x86, x64) and Linux (x86, x64, ARM)

Table 3: General Specifications (daA2500-14bm/bc)

3.3 Spectral Response

3.3.1 Mono Camera Spectral Response

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

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

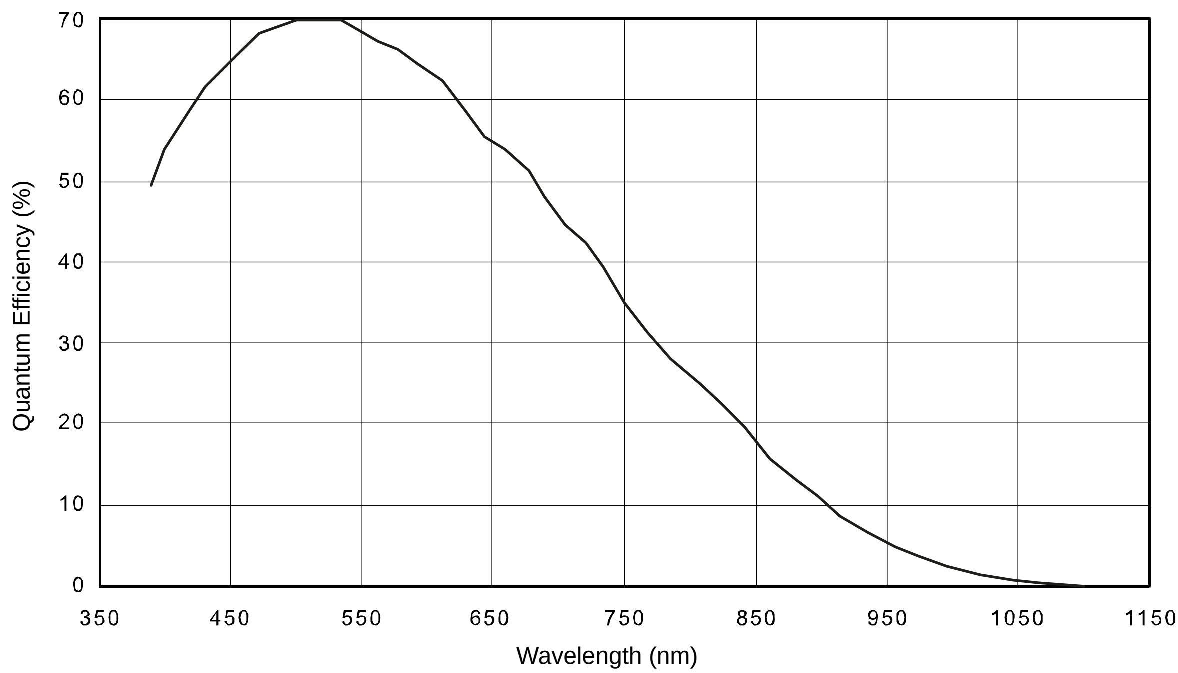


Fig. 2: daA1280-54bm Spectral Response (From Sensor Data Sheet)

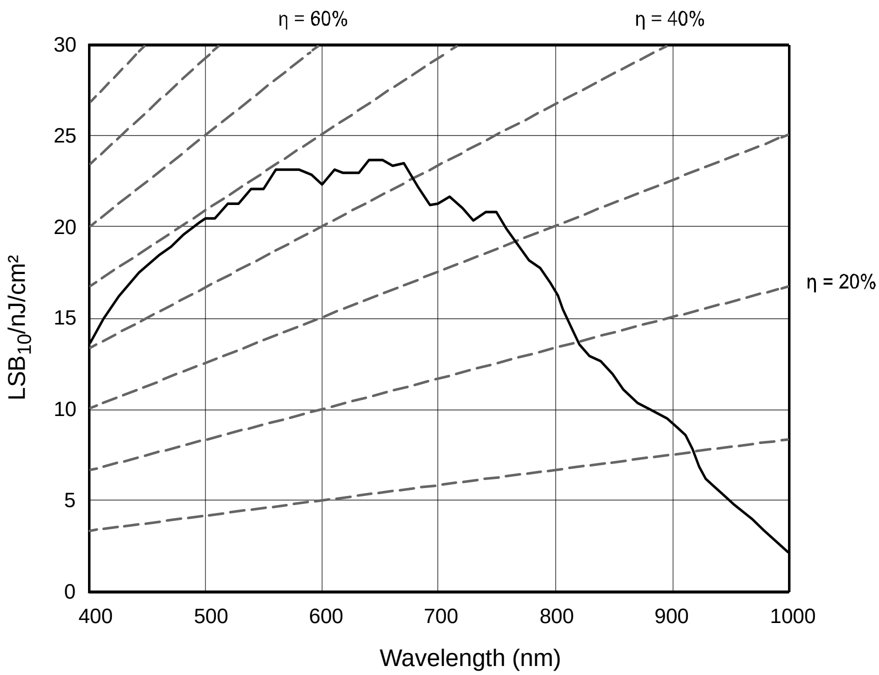


Fig. 3: daA1600-60bm Spectral Response (From Sensor Data Sheet)

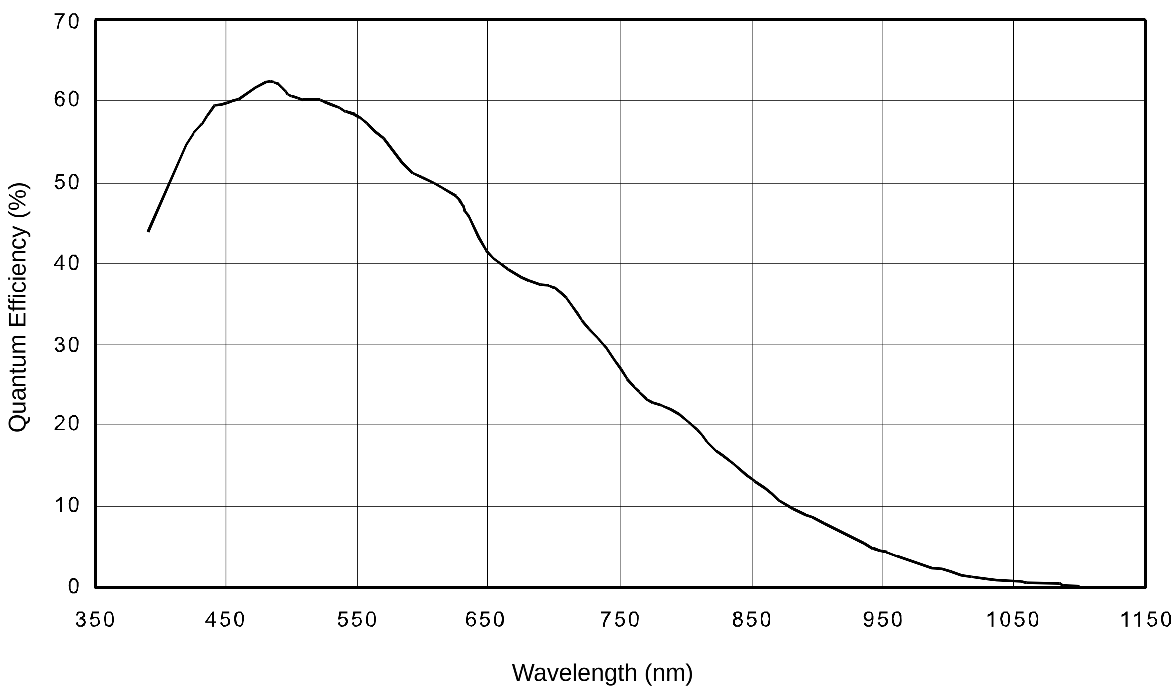


Fig. 4: daA2500-14bm Spectral Response (From Sensor Data Sheet)

3.3.2 Color Camera Spectral Response

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

i	The spectral response curves exclude lens characteristics, light source characteristics, and IR cut filter characteristics. To obtain best performance from color models of the camera, we recommend using a dielectric IR cut filter. The filter should transmit in a range from 400 nm to 700 ... 720 nm, and it should cut off from 700 ... 720 nm to 1100 nm. If you are using a ■ dart **bare board** color camera, we recommend installing an IR cut filter or a lens with an integrated IR cut filter when integrating the camera into the system. ■ dart **S-mount** color camera, we recommend attaching a lens with an integrated IR cut filter to the camera. ■ dart **CS-mount** color camera, a suitable IR cut filter is already built into the lens adapter. For more information, see Section 4.6 on page 30.

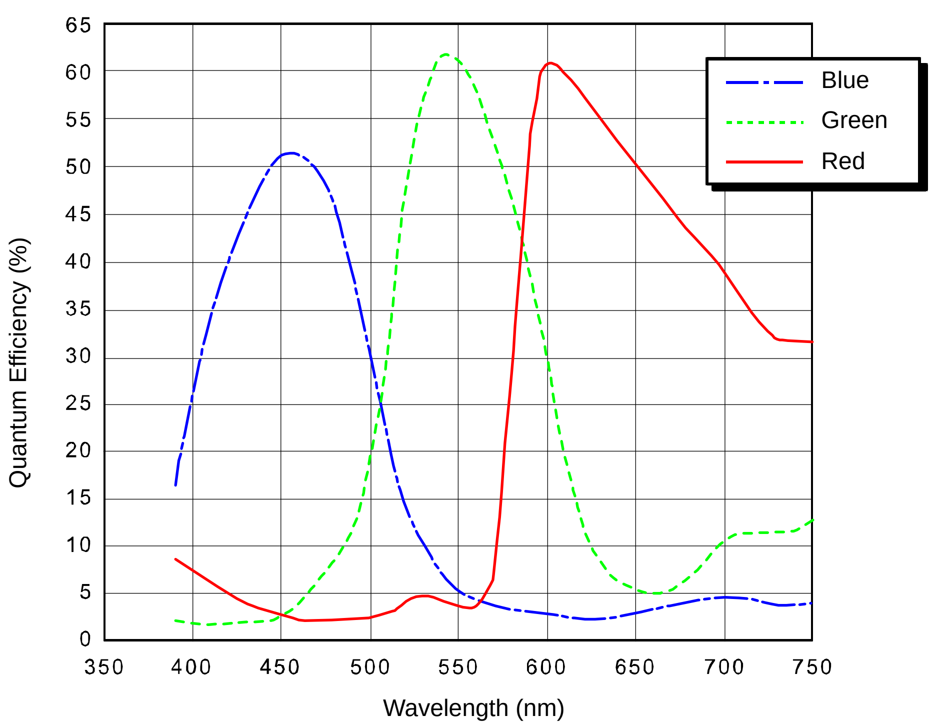


Fig. 5: daA1280-54bc Spectral Response (From Sensor Data Sheet)

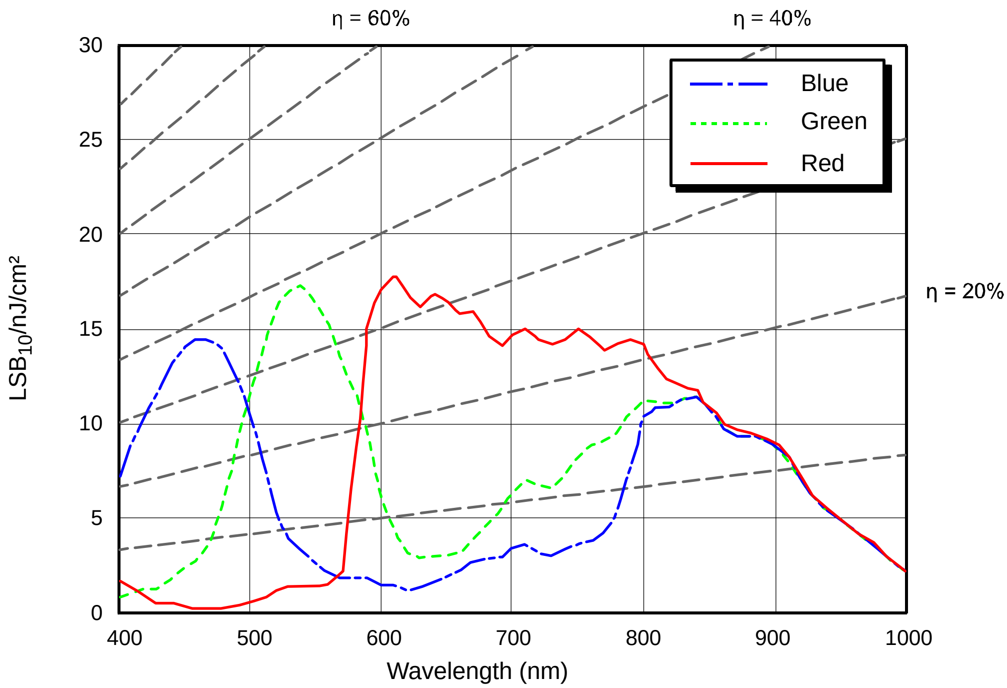


Fig. 6: daA1600-60bc Spectral Response (From Sensor Data Sheet)

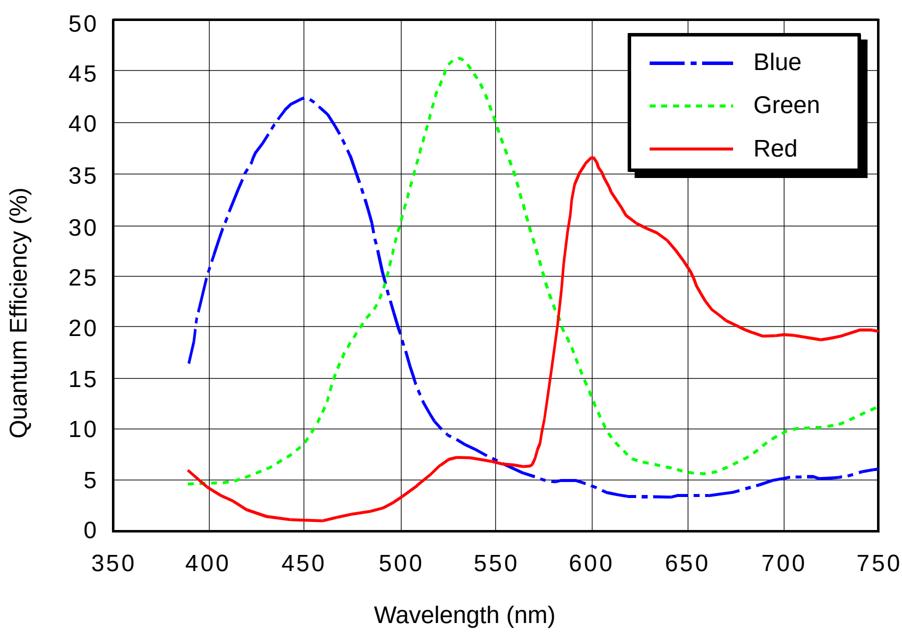


Fig. 7: daA2500-14bc Spectral Response (From Sensor Data Sheet)

3.4 Mechanical Specifications

3.4.1 Camera Dimensions and Mounting Points

The dimensions in millimeters for

- dart bare board cameras are as shown in Figure 8.
- dart cameras equipped with an S-mount lens adapter are as shown in Figure 10.
- dart cameras equipped with a CS-mount lens adapter are as shown in Figure 11.

All dart BCON cameras are equipped with mounting and heat dissipation holes on the bottom as shown in the drawings.

Bare Board Cameras (daA1600-60bm/bc)

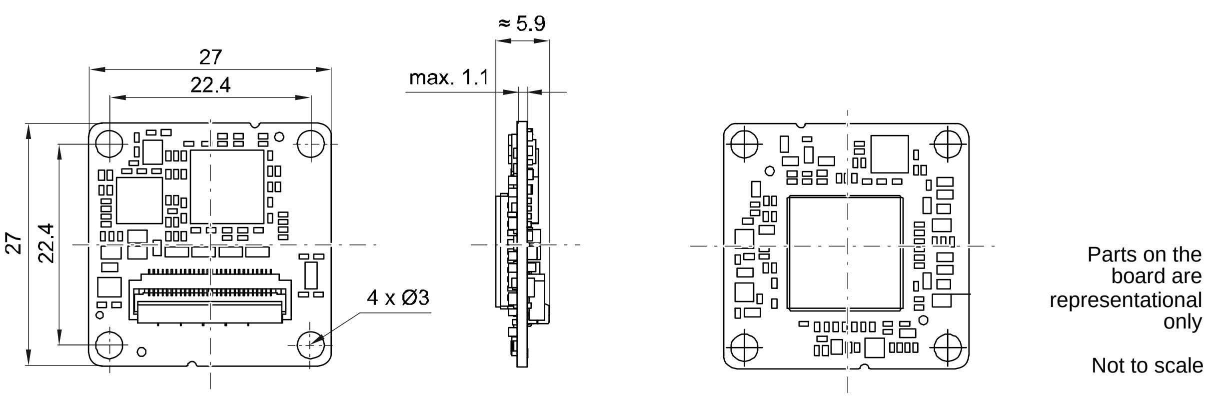


Fig. 8: Mechanical Dimensions (in mm) for Bare Board Cameras (daA1600-60bm/bc)

Bare Board Cameras (Other Models)

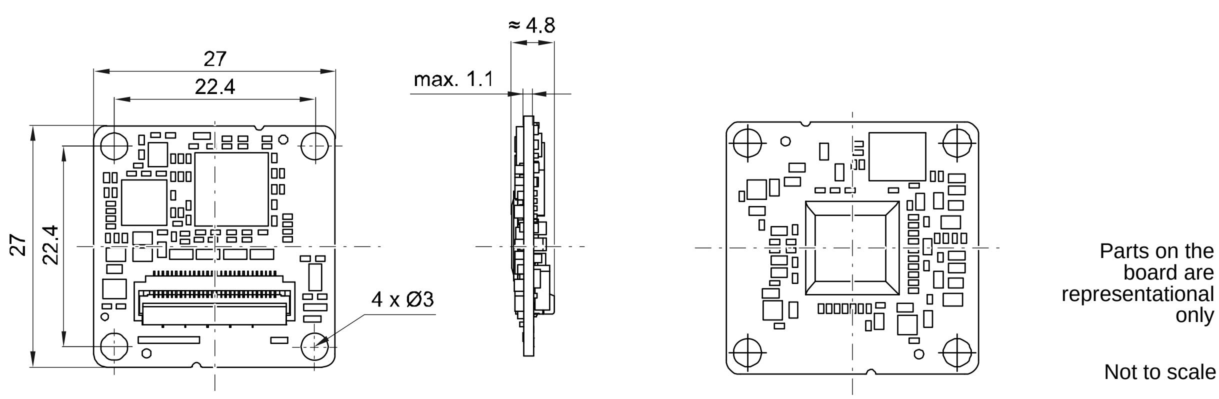


Fig. 9: Mechanical Dimensions (in mm) for Bare Board Cameras

S-Mount Cameras

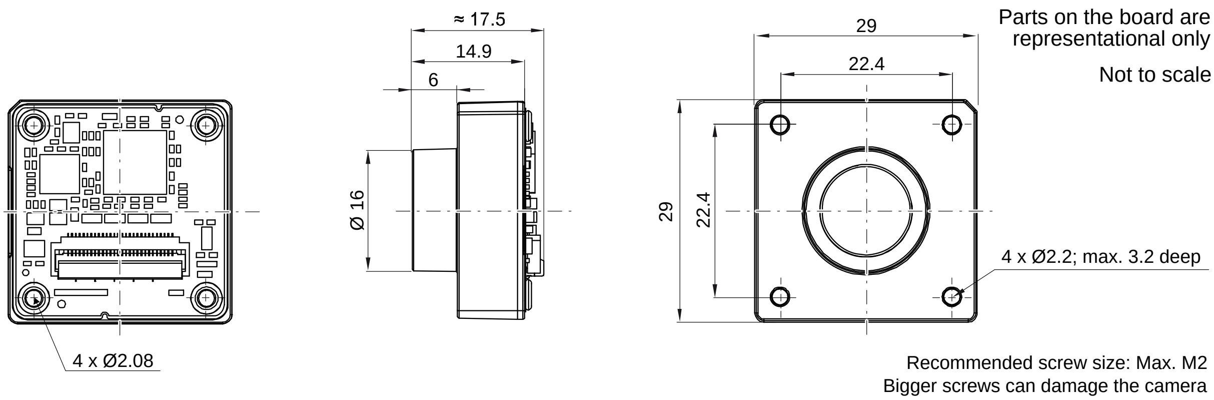


Fig. 10: Mechanical Dimensions (in mm) for Cameras with S-mount Lens Adapter

CS-Mount Cameras

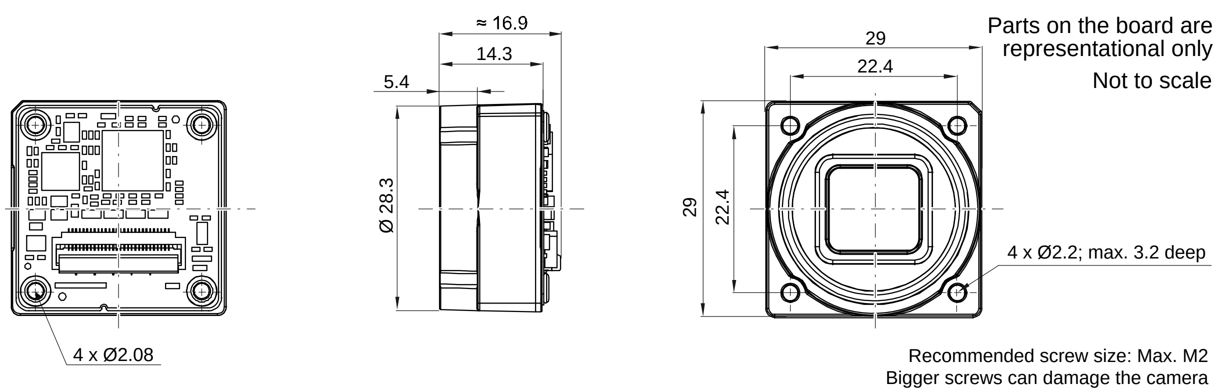


Fig. 11: Mechanical Dimensions (in mm) for Cameras with CS-mount Lens Adapter

3.4.2 Maximum Lens Dimensions

When mounting a lens on the dart S-mount or CS-mount camera models, observe the following:

- The lens must not intrude into the camera body more than the values given below.

Camera Model	Lens Adapter Type	Maximum Allowed Lens Intrusion (in mm)
daA1280-54bm daA2500-14bm	S-mount	11.7
	CS-mount	11
daA1280-54bc daA2500-14bc	S-mount	11.7
	CS-mount	7
daA1600-60bm	S-mount	10.7
	CS-mount	10
daA1600-60bc	S-mount	10.7
	CS-mount	7

Table 4: Maximum Allowed Lens Intrusion for dart BCON Camera Models

- The length of the threads on the camera's lens adapter is 7.5 mm for dart S-mount camera models and 5.6 mm for dart CS-mount camera models.
- For S-mount camera models, Basler recommends attaching an O-ring (Ø11 mm x Ø1.5 mm) to the S-mount lens. This makes it easier to adjust the lens.
- The CS-mount variants of the daA1600-60bm/bc cameras are shipped with a 0.5 mm spacer ring. The spacer ring optimizes the flange focal distance of these cameras. When mounting a lens on the camera, attach the spacer ring to the lens.

The maximum lens dimensions are also shown in Figure 12 on page 17 and Figure 13 on page 18.

NOTICE

Screwing in the lens too deep can damage camera components.

On S-mount cameras, the lens is screwed in and out to reach the desired level of focus. S-mount lenses do not have a defined flange and therefore will not stop before they touch (and possibly scratch or break) the sensor glass.

NOTICE

Incorrectly mounted C-mount lenses can damage camera components.

If you want to use C-mount lenses for dart CS-mount cameras, make sure that you attach a CS-mount adapter ring to the C-mount lens before mounting it on a dart CS-mount camera. Otherwise, the lens may reach too far into the camera and damage the IR cut filter (color cameras) or the sensor (mono cameras).

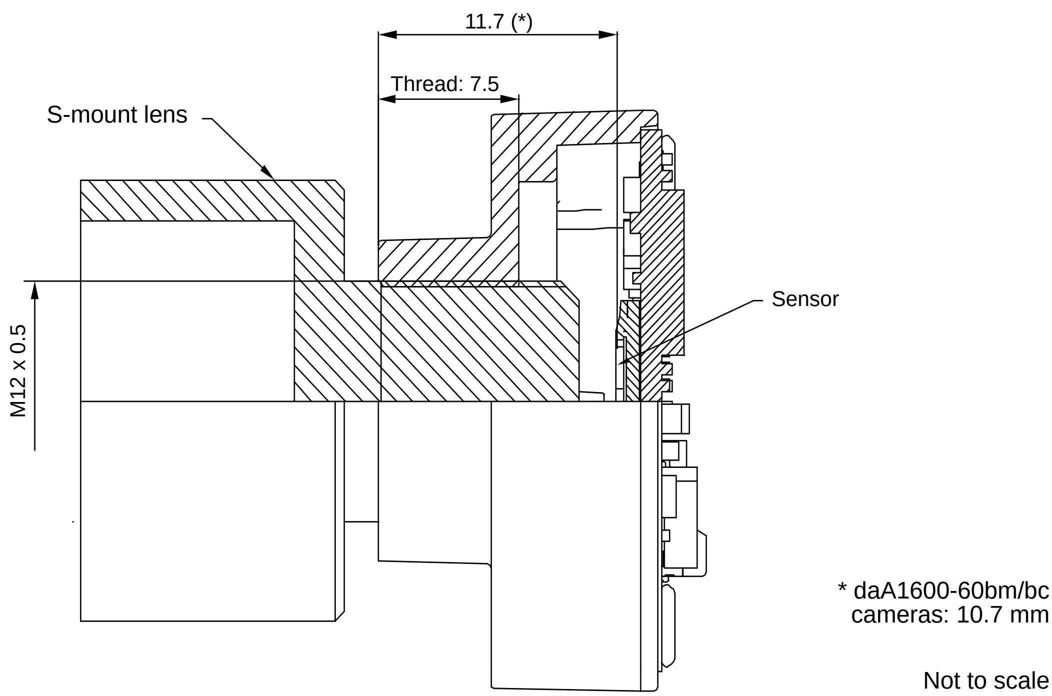


Fig. 12: Maximum Lens Dimensions (in mm) for Cameras with S-mount Lens Adapter

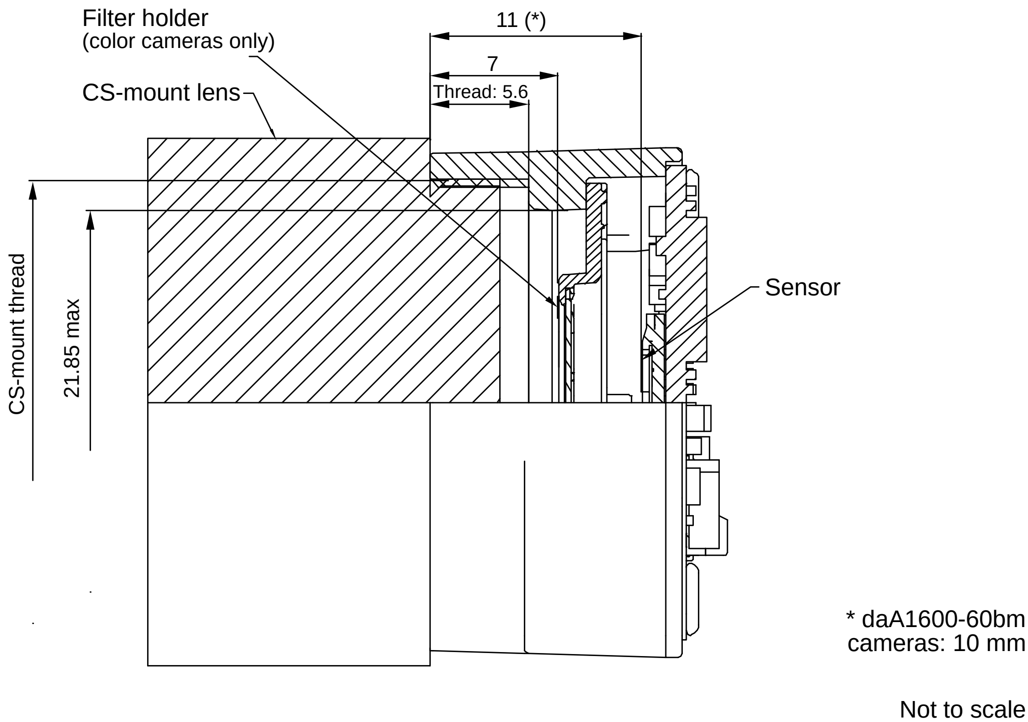


Fig. 13: Maximum Lens Dimensions (in mm) for Cameras with CS-mount Lens Adapter

3.4.3 Mechanical Stress Test Results

The following mechanical stress tests were performed on the Basler dart cameras:

- **dart bare board cameras:** See the stress tests for the dart S-mount camera. Because the circuit board of bare board models is the same as the circuit board of S-mount and CS-mount models, individual stress tests are not carried out for bare board cameras.
- **dart S-mount and CS-mount cameras:** The cameras were subjected to the stress tests listed in Table 5. 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 60068-2-27	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 5: Mechanical Stress Tests for dart S-mount Cameras

- The mechanical stress tests for S-mount cameras were performed with a dummy lens attached. The dummy lens had a mass of 30 g.
- The mechanical stress tests for CS-mount cameras were performed with a dummy lens attached. The dummy lens had a mass of 66 g.

Using a heavier lens requires an additional support for the lens.

3.5 Avoiding EMI and ESD Problems

BCON uses low voltage differential signaling (LVDS) to transmit image data from the camera to the image processing unit. LVDS is a differential signal transmission standard with two parallel traces for each signal. The voltage change on one trace is always the exact inverse of the other. Their electric fields cancel each other out, resulting in a reduced emission of electromagnetic radiation.

Due to the close proximity of the parallel traces, electric fields can't induce a significant differential voltage. This makes LVDS resistant against such interference. However, this does not apply to electromagnetic interference (EMI), unless the parallel traces are twisted, which is usually not the case for PCB traces and also not for flexible flat cables (FFCs).

Therefore, dart BCON cameras are prone to EMI and also electrostatic discharge (ESD). 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:

- Install the camera as far as possible from devices generating sparks or strong electromagnetic fields.
- Keep the cable away from unshielded electric motors, transformers, and unshielded coils.
- Decrease the risk of electrostatic discharge (ESD) 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.
- For more information about LVDS, see the *LVDS Owner's Manual*. The download link is provided in Section 1.3 on page 2.

i	For more information about avoiding EMI and ESD, see the application note *Avoiding EMI and ESD in Basler Camera Installations*. To download the application note, go to the Downloads section of the Basler website: www.baslerweb.com

3.6 Environmental Requirements

3.6.1 Temperature and Humidity

Requirements	dart Bare Board Models	dart S-mount and CS-mount Models
Device temperature during **operation**	0 °C ... +75 °C * (+32 °F ... +167 °F) *	0 °C ... +50 °C † (+32 °F ... +122 °F) †
Device temperature during **storage**	-20 °C ... +80 °C (-4 °F ... +176 °F)	
Humidity	20% ... 80%, relative, non-condensing	
Ambient temperature according to UL 60950-1	max. +50 °C (+122 °F)	
UL 60950-1 test conditions: no lens attached to the camera and without efficient heat dissipation; ambient temperature kept at +50 °C (+122 °F).		
* Temperature measured at the hottest point on the board. This point is significantly hotter than the other parts on the board. See Figure 14. † Temperature measured at the outside of the camera. See Figure 15.		

Table 6: Temperature and Humidity Requirements

Temperature Measurement Points

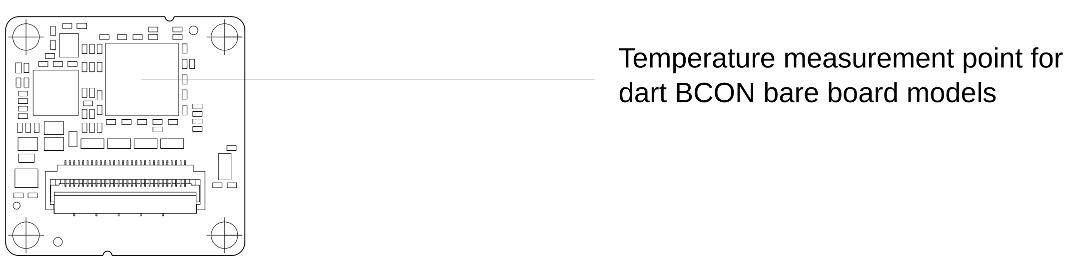


Fig. 14: Device Temperature Measurement Point (Bare Board Models)

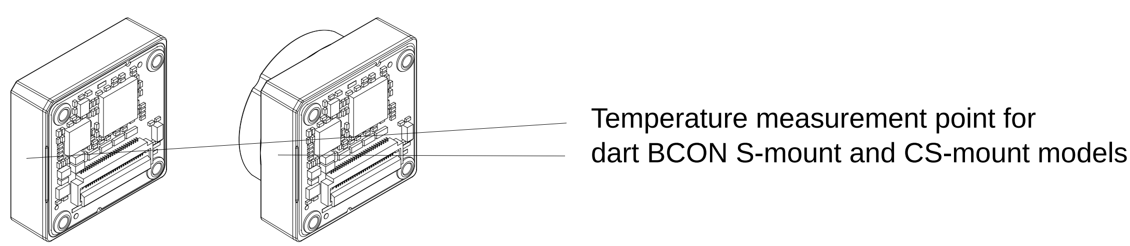


Fig. 15: Device Temperature Measurement Point (S-mount and CS-mount Models)

3.6.2 Heat Dissipation

You must provide sufficient heat dissipation to keep the operation temperature of the Basler dart below the values indicated in Section 3.6.1 on page 21.

Since each installation is unique, Basler does not supply a strictly required technique for proper heat dissipation. Instead, we provide the following general guidelines:

On all dart cameras, there are four holes at the corners of the camera board, designed for installing the camera. You can also use the holes to dissipate heat.

Depending on the dart variant, different components are used to dissipate heat:

- Basler bare board variants:
  The metallic borders of the holes are designed to dissipate heat to connecting metallic components.
- Basler dart S-mount and CS-mount variants:
  Rivets are placed in the four holes (see Figure 16). These rivets can be used to dissipate heat towards connected metallic components.

Fig. 16: Rivet at the Corner of the dart Camera Board

Usage of the holes or rivets depends on your system design. In all cases, make sure that the holes or rivets have contact to metallic components in your system. This way, the heat can dissipate towards the metallic components.

Three examples of how you can provide heat dissipation:

- Figure 17 (a): The camera front touches a mounting plate. Heat dissipates via the rivets, the camera front and the mounting plate.
- Figure 17 (b): The camera rear side touches a mounting plate. Heat dissipates via the rivets and the mounting plate.
- The use of a fan to provide air flow over the camera is an efficient method of heat dissipation.

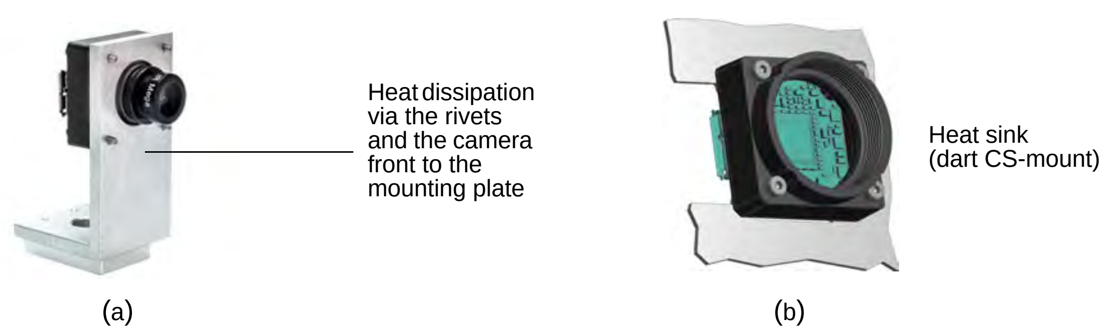


Fig. 17: Heat Sink Examples

4 Physical Interface

4.1 General Description of the Camera Connections

The camera is interfaced to external circuitry via a 28-pin flat flex cable (FFC) connector. The connector allows camera integration into embedded systems with both FPGAs and SoCs (Systems on a Chip).

The connector provides connections for LVDS image transfer, input and output signals, and power supply. Furthermore, standardized I²C lines are integrated, which are used for camera configuration via the Basler pylon Camera Software Suite.

There is also an LED indicator located on the back of the camera.

Figure 18 shows the location of the connector and the LED indicator.

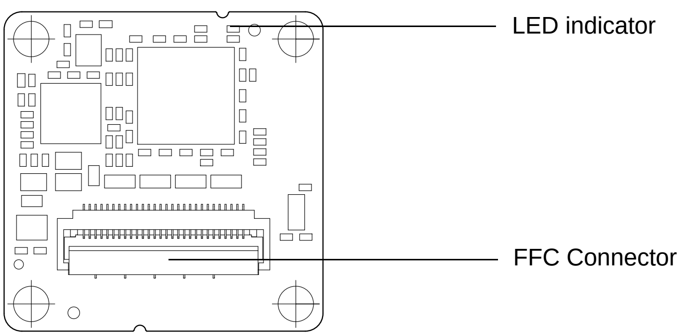


Fig. 18: Camera Connector and LED Indicator

4.2 FFC Connector

The 28-pin connector on the camera is a flexible flat cable (FFC) connector by Hirose Electric Co. It is specifically designed for shielded and impedance-controlled flexible flat cables. However, standard flexible flat cables without a ground layer can also be inserted. They will make good electrical contact, but under certain circumstances, you will notice a high rate of bit errors.

The connector order code is Hirose FH41-28S-0.5SH(05).

4.2.1 Connector Pin Assignments

Pin 1 Position

As shown in Figure 19, pin 1 of the FFC connector is indicated by an arrow, and pin 28 is indicated by the number 28.

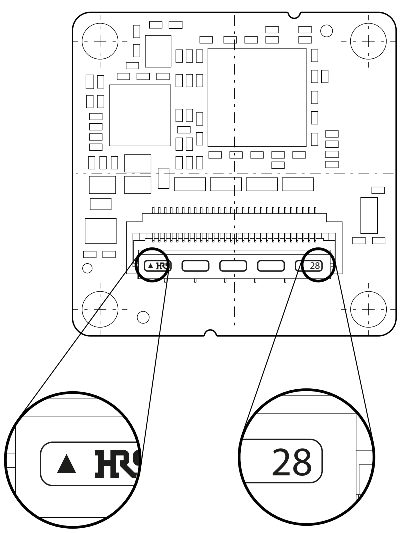


Fig. 19: Pin Positions

Pin Assignments and Description

Pin # Camera Side	Pin Name	Pin # System Side	Function	Notes
1	GND	28	Power Supply	All GND pins, including the five shield GND contacts on the opposite side of the connector, must be connected to a common GND plane. Avoid long PCB traces, prefer direct vias.
2	VCC	27	Power Supply	Single camera supply voltage, 5 VDC nominal. Connect all pins.
3	VCC	26	Power Supply	
4	VCC	25	Power Supply	
5	GND	24	Power Supply	See note for pin 1.
6	I2C_ID	23	Configuration Interface	I^{2}C interface address selector. Connect to GND for an I^{2}C slave address of 0x3C. Connect to 3.3 VDC for an I^{2}C slave address of 0x3D.
7	I2C_SDA	22	Configuration Interface	I^{2}C interface data line. A pull-up resistor to 3.3 VDC is required. Refer to the I^{2}C standard for the appropriate resistor value. Note that FFCs can have a significant line capacitance.
8	GND	21	Power Supply	See note for pin 1.
9	I2C_SCL	20	Configuration Interface	I^{2}C interface clock line. A pull-up resistor to 3.3 VDC is required. Refer to the I^{2}C standard for the appropriate resistor value. Note that FFCs can have a significant line capacitance.
10	GND	19	Power Supply	See note for pin 1.
11	CC0-	18	Trigger and General Purpose Input	LVDS input to the camera, mainly as external trigger. No termination resistor is required at the system side.
12	CC0+	17	Trigger and General Purpose Input	
13	GND	16	Power Supply	See note for pin 1.
14	X3-	15	Serial Data Output	LVDS output data lane 3. Use differential routing. A 100 ohm termination resistor is required across the differential pair.
15	X3+	14	Serial Data Output	
16	GND	13	Power Supply	See note for pin 1.

Table 7: Pin Assignments and Description

Pin # Camera Side	Pin Name	Pin # System Side	Function	Notes
17	XCLK-	12	Serial Data Output	LVDS clock output. Use differential routing. A 100 ohm termination resistor is required across the differential pair.
18	XCLK+	11	Serial Data Output	
19	GND	10	Power Supply	See note for pin 1.
20	X2-	9	Serial Data Output	LVDS output data lane 2. Use differential routing. A 100 ohm termination resistor is required across the differential pair.
21	X2+	8	Serial Data Output	
22	GND	7	Power Supply	See note for pin 1.
23	X1-	6	Serial Data Output	LVDS output data lane 1. Use differential routing. A 100 ohm termination resistor is required across the differential pair.
24	X1+	5	Serial Data Output	
25	GND	4	Power Supply	See note for pin 1.
26	X0-	3	Serial Data Output	LVDS output data lane 0. Use differential routing. A 100 ohm termination resistor is required across the differential pair.
27	X0+	2	Serial Data Output	
28	GND	1	Power Supply	See note for pin 1.

Table 7: Pin Assignments and Description

4.3 Voltage Requirements

4.3.1 Absolute Maximum Ratings

Pin Name	**Absolute Minimum**	**Absolute Maximum**
VCC	-0.3 V	6.5 V
CC0+ CC0-		4.2 V
I2C_SCL I2C_SDA I2C_ID		3.6 V

4.3.2 Input Voltages

Pin Name	**Minimum**	**Nominal** (recommended)	**Maximum**	**Notes**
VCC	4.5	5.0 V	5.5 V	
CC0+ CC0-	0.2 V	-	1.8 V	Common mode voltage
	0.1 V	-	-	Differential voltage
I2C_SCL I2C_SDA I2C_ID	0.0 V	0.0 V	1.0 V	Input Low Voltage
	2.4 V	3.3 V	3.5 V	Input High Voltage

Table 8: Recommended Input Voltages (Camera Supply, LVDS Inputs, I^2C ID, Data and Clock Lines)

4.3.3 Output Voltages

Pin Name	Minimum	Nominal (recommended)	Maximum	Notes
X1-; X1+ X2-; X2+ X3-; X3+ X4-; X4+ XCLK-; XCLK+	1.125 V	1.25 V	1.375 V	Common mode voltage
	247 mV	-	600 mV	Differential voltage

Table 9: Output Voltages (LVDS Outputs - Data Lanes and Clock Lines)

4.4 Timing Characteristics

Pin	Min.	Max.	Notes
XCLK	20 MHz	84 MHz	Selectable via control interface. Settable in steps of 8, i.e. 20 MHz, 28 MHz, 36 MHz, etc.
X0-X3	140 Mb/s	588 Mb/s	Selectable via control interface.
I2C_SCL	0 kHz	400 kHz	In the *I²C-Bus Specification and User Manual*, this mode is called "fast-mode" with 400 kbit/s.

Table 10: Timing Characteristics

4.5 LED Indicator

There is an LED indicator on the back of the camera board (see Figure 18 on page 23).

Green LED is ...	Description
Dimming up/down	The camera is being configured.
Lit permanently	Camera is configured and operative.
Blinking rapidly	Internal error.

Table 11: LED Statuses

The LED can be turned off permanently by setting the DeviceIndicatorMode parameter to Inactive.

If the parameter is set to Inactive and the setting is stored in the startup set, the LED will light up for approximately 1 second. For more information about the startup set, see Section 11.15 on page 126.

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

```
// Turn off the LED indicator (LED is turned off permanently)
camera.DeviceIndicatorMode.SetValue(DeviceIndicatorMode_Inactive);

// Turn on the LED indicator (LED is on during camera operation)
camera.DeviceIndicatorMode.SetValue(DeviceIndicatorMode_Active);
```

4.6 IR Cut Filter

The dart **CS-mount color** cameras are equipped with an IR cut filter. The filter is mounted in a filter holder located in the lens mount.

The filter on dart CS-mount color cameras has the following spectral characteristics:

Wavelength (nm)	Transmittance
420 – 600	$T_{min} \geq 90\%$ $T_{avg} \geq 92\%$
650 ± 10	T = 50%
700 – 1000	$T_{avg} \leq 1\%$
1000 – 1100	$T_{avg} \leq 5\%$

Table 12: IR Cut Filter Spectral Characteristics

If you want to operate a dart S-mount color camera with an IR cut filter, you must attach a lens with an integrated IR cut filter to the camera.

NOTICE

Using a lens with a too long thread length can damage the IR cut filter or the filter holder.

Make sure that you do not damage the IR cut filter by using lenses with a too long lens thread. Otherwise, the IR cut filter or the filter holder will be damaged or destroyed and the camera will no longer operate. For more information, see Section 3.4.2 on page 16.

5 BCON Interface

5.1 Interface Specifications

<table>
<tr><th>Specification</th><th>Value</th><th>Notes</th></tr>
<tr><td>Data lanes</td><td>4</td><td></td></tr>
<tr><td>Clock lines</td><td>1</td><td>Used for the word clock.</td></tr>
<tr><td>Control lines</td><td>3</td><td>I²C interface lines (clock, data, ID).</td></tr>
<tr><td>Input lines</td><td>1</td><td></td></tr>
<tr><td>Output lines</td><td>2</td><td></td></tr>
<tr><td>Serialization factor</td><td>7</td><td rowspan="3">For more information, see Section 5.2.2 on page 33.</td></tr>
<tr><td>Data bus width</td><td>28 bit</td></tr>
<tr><td>Bus width reserved for image data</td><td>24 bit</td></tr>
<tr><td>Word clock frequency
Settable in increments of 8, i.e. 20 MHz, 28 MHz, 36 MHz, etc.</td><td>12-84 MHz</td><td rowspan="3">For more information, see Section 5.3 on page 41.</td></tr>
<tr><td>Word clock duty cycle</td><td>4/7</td></tr>
<tr><td>Bit clock frequency</td><td>$f_{WordClk}$ * 7</td></tr>
</table>

Table 13: BCON Interface Specifications

BCON and ChannelLink

The BCON specifications are similar to the ChannelLink-I specifications, in particular the specifications for Texas Instrument's DS90CR28x deserializer. This applies to the following:

- Separation of data, clock, and control lines
- Number of data lanes
- Serialization factor
- Bus width
- Clock frequencies
- Word clock duty cycle

For more information about Channel Link, see the *Channel Link Design Guide*. The download link is provided in Section 1.3 on page 2.

5.2 BCON Data Output

The data output by the four data lanes (X0, X1, X2, X3) consists of the following information:

- **Frame valid bit** (FVal): Indicates that a valid frame is being transmitted.
- **Line valid bit** (LVal): Indicates that a valid line is being transmitted.
- **Frame information** (FrameInfo): Includes information about the pixel format and the number of pixels per clock cycle. Sent before each frame transmission.
- **Pixel data** (Line 1, Line 2, ..., Line n): Pixel data, line by line.
- **Line checksum** (Cksum 1, Cksum 2, ..., Cksum n): Check sum for each line of pixel data. Sent after each line transmission.
- **BCON output bits** (Output 0 and Output 1): Two bits that can be used to transmit camera output signals, e.g. Flash Window or Exposure Active.

Frame information, pixel data, and line checksums are transmitted on a common 24-bit image data channel ("Data").

5.2.1 Data Flow

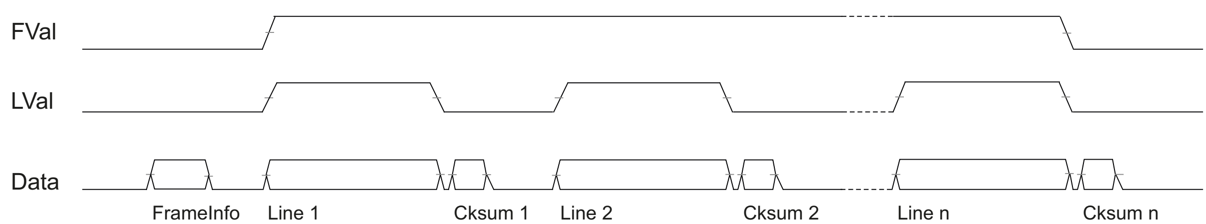


Fig. 20: Image Data Flow (Schematic Design)

The BCON interface does not provide data flow control. Also, in contrast to the CameraLink interface, the BCON interface does not provide a data valid bit (DVal). Only FVal and LVal are used as synchronization signals.

i	Whenever FVal is low and the camera is not sending FrameInfo or Cksum data, a sync pattern is transmitted. This allows you to set up the correct word alignment. For more information, see Section 5.2.6 on page 37.

5.2.2 Data Serialization and Timing

BCON data output is serialized using up to 28 bits on up to four data lanes (X0 to X3).

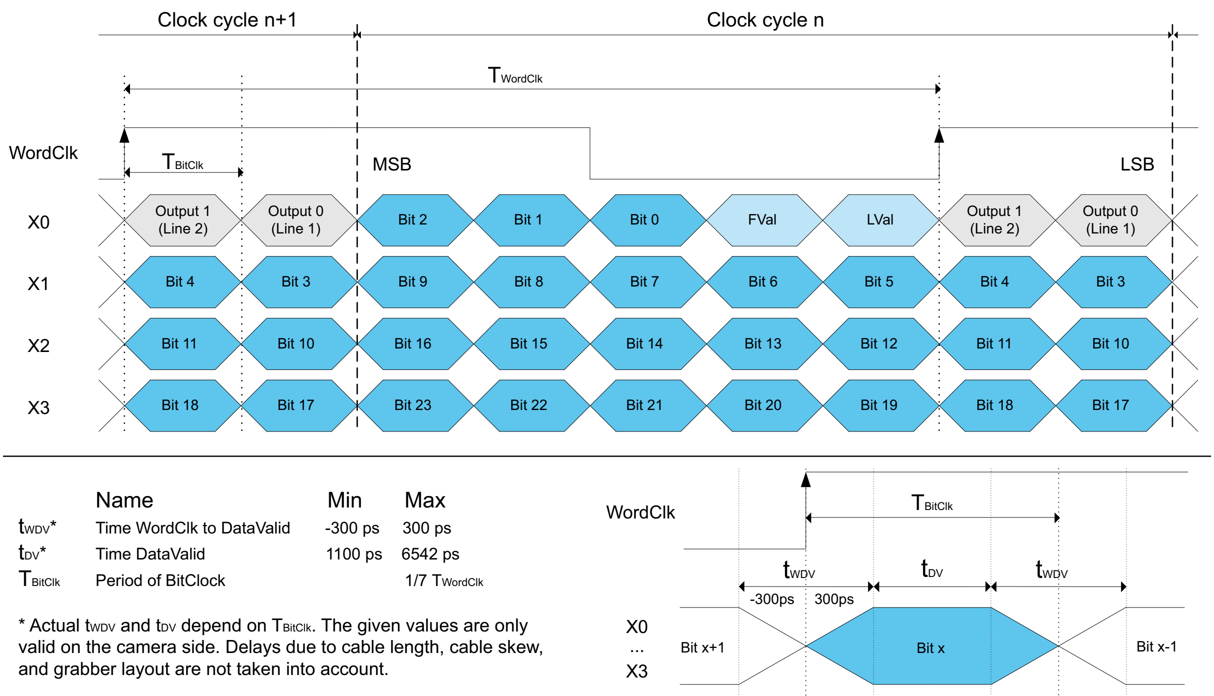


Fig. 21: BCON Serialization and Timing

- Two bits are reserved for the output signals (Output 0 and Output 1).
  They are always transmitted on data lane X0 at position 0 and 1.
- Another two bits are reserved for the synchronization signals (FVal and LVal).
  They are always transmitted on data lane X0 at position 2 and 3.
- The remaining 24 bits are reserved for the image data channel including frame information, pixel data, and line checksums.
  The actual number of bits and active data lanes used depends on the data being transmitted:

Data Transmitted	Bits Used	Data Lanes Active
Frame information 8-bit pixel data 8-bit checksum	Bit 0 - Bit 7	X0, X1
12-bit pixel data 12-bit checksum	Bit 0 - Bit 11	X0, X1, X2
16-bit pixel data 16-bit checksum	Bit 0 - Bit 15	X0, X1, X2
24-bit pixel data 24-bit checksum	Bit 0 - Bit 23	X0, X1, X2, X3

Table 14: Data Channel Usage

If you are only using 8-bit pixel formats, e.g. because you are using a monochrome camera, the X2 and X3 data lanes are never active. Accordingly, if you are only using 8-bit or 12-bit pixel formats, X3 is never active.

Basler recommends, however, to always connect all BCON data lanes from X0 to X3.

For more information about pixel formats and bit depths, see Section 5.2.8.1 on page 38.

5.2.3 Frame Valid Bit and Line Valid Bit

The frame valid bit (FVal) and the line valid bit (LVal) indicate that a valid frame or line is being transmitted.

LVal and FVal are transmitted on data lane X0 at position 2 and 3 (see Figure 21 on page 33).

5.2.4 Frame Information

As shown in Figure 20 on page 32, a "FrameInfo" block is sent before each frame transmission. Each FrameInfo block includes the following information:

- Pixel format valid for the following pixel data.
- Number of pixels per clock cycle.

Frame information is always transmitted on **bits 0 through 7** of the 24-bit data channel:

	MSB																							**LSB**
Data Channel Bits	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	●	●	●	●	●	●	●	●
● = Frame information, x = not used																								

Table 15: Data Channel Usage During FrameInfo transmission

In every FrameInfo block, the following information is transmitted:

Clock Cycle	Description	Bit Length	Enumeration	Value
n	Command word	8	-	0b10111101
n+1	Pixels per clock cycle value	8	One	0b00000000
			Two	0b00000001
n+2 ... n+5	Pixel format value ▪ Transmitted in 4 Bytes, least significant byte sent first ▪ Adheres to the *GenICam Pixel Format Names and Values* specification. The download link is provided in Section 1.3 on page 2.	32	Mono 8	0x01080001
			Mono 12	0x01100005
			Bayer GR 8 *	0x01080008
			Bayer RG 8 *	0x01080009
			Bayer GB 8 *	0x0108000A
			Bayer BG 8 *	0x0108000B
			Bayer GR 12 *	0x01100010
			Bayer RG 12 *	0x01100011
			Bayer GB 12 *	0x01100012
			Bayer BG 12 *	0x01100013
			YCbCr422	0x0210003B
			RGB 8	0x02180014

* The standard Bayer filter alignment for Basler dart cameras is GB. If supported, other alignments can be set using the Reverse X and Reverse Y features. For more information, see Section 11.9 on page 105.

Table 16: FrameInfo Block Structure

Example: If the camera is configured for two pixels per clock cycle and Mono 12 pixel data, the camera sends the following information before each frame transmission:

Clock Cycle	Data	Notes
n	0b10111101	Command word
n+1	0b00000001	Two pixels per clock cycle
n+2	0b00000101	Mono 12 pixel format value = 0x01100005 = 0b00000001 00010000 00000000 00000101 Least significant byte sent first
n+3	0b00000000	
n+4	0b00010000	
n+5	0b00000001	

Table 17: FrameInfo Data (Example)

5.2.5 Line Checksum

The Basler BCON interface uses even parity in order to detect transmission errors.

For each image line, a checksum is calculated. The checksum is added to the output data at the end of each line. The checksum consists of a number of parity bits depending on the selected pixel format. For every bit position of every pixel of the current line, an even parity bit is calculated.

The checksum is transmitted directly after the data of the last pixel in the image line has been transmitted (see also Figure 20 on page 32).

Example: Assume the camera is set for Mono 8 pixel data and one pixel per clock cycle. To demonstrate the principle, also assume that the image ROI width is set to 5 pixels only. This means that each image line consists of 5 x 8 bit image data. Table 18 shows sample data for one image line under these conditions.

Data Channel Bit Position	Sample Image Data					Count of 1-bits	Checksum Bit (Even Parity Bit)	
	Clock Cycle							
	n	n+1	n+2	n+3	n+4			
Bit 0	1	1	0	1	0	3	**1**	LSB
Bit 1	1	0	1	0	0	2	**0**	
Bit 2	1	0	0	0	0	1	**1**	
Bit 3	1	1	1	0	1	4	**0**	
Bit 4	1	0	1	0	0	2	**0**	
Bit 5	0	0	0	1	1	2	**0**	
Bit 6	0	0	0	0	0	0	**0**	
Bit 7	1	1	0	0	1	3	**1**	MSB

Table 18: Line Checksum Data (Example)

In the example shown above, the line checksum data is 0b10000101.

5.2.6 Sync Pattern

Whenever the frame valid bit (FVal) is low and the camera is not sending frame information or line checksum data, a fixed sync pattern is transmitted. This allows you to set up the correct word alignment.

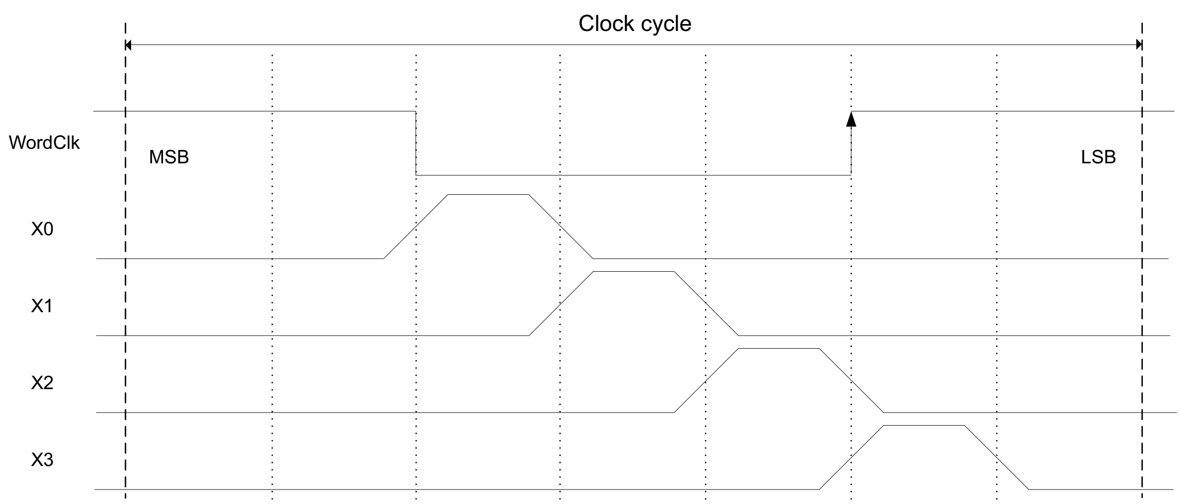


Fig. 22: BCON Sync Pattern

5.2.7 BCON Output Bits

The BCON interface provides two bits that can be used to transmit two camera output signals: Output 0 and Output 1.

Output 0 and Output 1 are transmitted on data lane X0 at position 0 and 1 (see Figure 21 on page 33).

Depending on the camera model, the following output signals can be available:

- Flash Window
- Exposure Active
- User Output 1
- User Output 2

For more information about configuring output signals, see Section 8.2 on page 53.

5.2.8 Pixel Data

5.2.8.1 Pixel Formats

Pixel Format	Available on	Bit Depth
Mono 8	Mono cameras only	8
Bayer 8	Color cameras only	
Mono 12	Mono cameras only	12
Bayer 12	Color cameras only	
YCbCr422	Color cameras only	16
RGB 8	Color cameras only	24

Table 19: Available Pixel Formats

For image data transmission, the BCON interface does not distinguish between the individual 8-bit pixel formats (Mono 8 / Bayer 8). Similarly, the BCON interface does not distinguish between the individual 12-bit pixel formats (Mono 12 / Bayer 12).

However, information about the specific pixel format used for frame acquisition is included in the FrameInfo part of the data stream. The FrameInfo part is sent before each frame transmission. For more information, see Section 5.2.3 on page 34.

5.2.8.2 Pixels per Clock Cycle

On all BCON cameras, you can set the number of pixels transmitted per clock cycle.

One Pixel per Clock Cycle

This mode is available **for all pixel formats**.

In this mode, the camera transmits pixel data as follows:

- On each clock cycle, data for **one** pixel is transmitted via the BCON interface.
- On the first clock cycle, the camera transmits the data for the first pixel in line one. On the second clock cycle, the camera transmits the data for the second pixel in line one, and so on until the line is complete.
- When line one is complete, the camera sends a checksum for the line. After that, transmission of line two begins and proceeds in a similar fashion. Data transmission continues line by line until all frame data has been transmitted.

During pixel data transmission, the 24 bits of the data channel will be used as shown in Table 20:

	MSB																							LSB
Data Channel Bits	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Mono 8 Bayer 8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	●	●	●	●	●	●	●	●
Mono 12 Bayer 12	x	x	x	x	x	x	x	x	x	x	x	x	●	●	●	●	●	●	●	●	●	●	●	●
YCbCr422	x	x	x	x	x	x	x	x	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
RGB8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
● = Pixel data, x = not used																								

Table 20: Data Channel Usage (One Pixel per Clock Cycle)

Two Pixels per Clock Cycle

This mode is only available **for 8-bit and 12-bit pixel formats** (i.e. not available for YCbCr422 and RGB 8 pixel formats).

In this mode, the camera transmits pixel data as follows:

- On each clock cycle, the data for **two** pixels are transmitted via the BCON interface.
- On the first clock cycle, the camera transmits the data for the first two pixels in line one. On the second clock cycle, the camera transmits the data for the next two pixels in line one, and so on until the line is complete.
- When line one is complete, the camera sends a checksum for the line. After that, transmission of line two begins and proceeds in a similar fashion. Data transmission continues line by line until all frame data has been transmitted.

During pixel data transmission, the 24 bits of the data channel will be used as shown in Table 21:

	MSB																							**LSB**
Data Channel Bits	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Mono 8 Bayer 8	x	x	x	x	x	x	x	x	○	○	○	○	○	○	○	○	●	●	●	●	●	●	●	●
Mono 12 Bayer 12	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●	●	●	●	●	●	●	●	●
● = Data for pixel n, ○ = Data for pixel n+1, x = not used																								

Table 21: Data Channel Usage (Two Pixels per Clock Cycle)

Setting the Number of Pixels per Clock Cycle

You can use the pylon API to set the number of pixels per clock cycle from within your application software. The following code snippet illustrates using the API to set the clock speed:

```
// Set the camera to transmit two pixels per clock cycle
Camera.BConPixelsPerClockCycle.SetValue(BConPixelsPerClockCycle_Two);
```

5.3 BCON Output Clock

The base clock for the BCON output is the word clock WordClk. Its frequency can range from 12 MHz to 84 MHz (settable in increments of 8, i.e. 12, 20, 28, etc.).

The word clock duty cycle is 4/7, i.e, the word clock signal is active (high) during 4/7 of each period (≈57.14% of the time).

The word clock is transmitted on the separate output clock line XCLK.

The bit clock frequency f_{BitClk} is not transmitted, but can be calculated. The bit clock frequency is 7 times faster than the word clock frequency:

$f_{BitClk} = f_{WordClk} * 7$

Setting the Clock Speed

You can use the pylon API to set the BCON pixel clock speed from within your application software. The following code snippet illustrates using the API to set the clock speed:

```
// Set the BCON output clock speed to 84 MHz
Camera.BCONClockFrequency.SetValue(BCONClockFrequency_MHz_84);
```

5.4 BCON Input

The BCON input line can be used to send a trigger or other input signals to the camera.

The state of the input can be read via the camera’s control interface at any time.

5.5 I^{2}C Interface

The camera is configured via a standard I^{2}C interface. For a detailed description, refer to the *I^{2}C-bus Specification and User Manual*. The download link is provided in Section 1.3 on page 2.

5.5.1 I^{2}C Features

In addition to the mandatory I^{2}C features, the Basler dart BCON camera uses the optional Clock Stretching feature.

For detailed information about the features, see the *I^{2}C-Bus Specification and User Manual*.

Clock Stretching

The Basler dart BCON camera uses clock stretching, i.e. it may hold I2C_SCL low.

When designing your system, make sure that the I^{2}C master can handle clock stretching.

5.5.2 Basler Specifics

The following sections provide information about settings that deviate from the I^{2}C bus specifications.

The control channel of the BCON interface is designed as an I^{2}C bus interface. A Basler BCON camera is an I^{2}C slave device that can handle data transfers at up to 400 kbit/s ("Fast Mode" according to I^{2}C).

Fast Mode (Fm) devices are downward compatible and can communicate with Standard Mode (Sm) devices in a 0 to 100 kbit/s I^{2}C bus system.

However, as they are not upward compatible, Fast Mode devices should not be integrated in an I^{2}C bus system that is configured for a faster mode, i.e. Fast Mode Plus (Fm+) or High Speed Mode (Hs). Fast Mode devices cannot follow the higher transfer rate of the faster modes. Therefore, unpredictable states might occur.

These limitations have to be observed when designing the I^{2}C bus master.

5.5.2.1 Additional I^{2}C Control Line (I2C_ID)

In addition to the standard I^{2}C control lines (I2C_SDA data line, I2C_SCL clock line), there is an additional I^{2}C control line, the I2C_ID line.

Normally, the I2C_ID line lets you define the slave address that the camera should respond to. You can connect up to two cameras to a single I²C bus. The logical level of the I2C_ID line determines which slave address is used (i.e. to which the camera should respond).

Logical Level of I2C_ID Line	Slave Address	Communication with
0 (low)	0x3c	Camera 1
1 (high)	0x3d	Camera 2

Normally, the I2C_ID line is permanently wired to a fixed level, either low (0 V) or high (3.3 V), depending on the desired slave address.

The I2C_ID line can also be used to reset the camera. For detailed information, see Section 5.5.2.2 on page 43.

5.5.2.2 Reset Function via I2C_ID Line

If you want to reset the camera because it does not respond anymore, you can activate a reset function via the I2C_ID line.

Using the reset function

When you switch on the supply voltage, ensure the following:

- The I2C_ID line must have the correct logical level according to the desired slave address.
- The logical level at system start must not be changed during operation, only in case of a reset.

To reset the camera:

The I2C_ID line is set to a certain logical level, i.e. either to 0 (low) or to 1 (high).

1. Set the I2C_ID line to the opposite logical level for at least 100 ms.
2. Set the I2C_ID line back to its original logical level.

   The switch back will trigger the reset of the microcontroller.

   The FPGA will be loaded again, but not reset.

6 Hardware Design Recommendations

This chapter provides recommendations for designing your hardware to work with the Basler BCON interface.

For more information about the BCON interface, see Section 4.2.1 on page 24.

Identifiers such as X1, XCLK, CC0, or I2C_SDC refer to the pin names given in Section 4.2.1 on page 24.

6.1 BCON Data Output (X0, X1, X2, X3, XCLK)

BCON uses four data lanes (X0 through X3) and one clock line (XCLK) to transmit image data from the camera to the image processing unit.

During data transmission, signal reflections can be a major cause of bit errors. Every signal change in the transmitter launches a wave that travels to the receiver. If the termination resistor at the other end of the trace matches the impedance of the differential trace, it consumes most of the energy of the wave.

Therefore, both the differential traces and the termination resistors must have an impedance of 100 ohm. If they do not match the target impedance, the reflected wave can disturb the original signal and cause bit errors.

Design Recommendations for Data Transmission

Printed Circuit Board (PCB)

- Design the PCB to have a differential impedance of 100 ohms. If this is not possible, make the traces between the connector and the receiver as short as possible.
- Run the differential traces as close as possible, but leave a gap of at least three trace widths between the individual pairs.
- Ensure that the two traces of a differential pair have the same length. A mismatch of 5 mm or less is not critical.
- Ensure that the lengths of all pairs do not differ more than 20 mm.
- Place a 100 ohm termination resistor across the traces of each pair as close to the receiver's pins as possible.

- Avoid vias. Run the traces from the connector to the receiver on a single PCB layer if possible.
- Avoid stubs. Run the traces through the termination resistor's solder pads rather than branching the trace. Even small stubs can cause reflections.

Flexible Flat Cable (FFC)

- Choose an FFC with a differential impedance of 100 ohms to avoid problems with electromagnetic interference (EMI). For more information about avoiding problems with EMI, see Section 3.5 on page 20.
- Keep the cable away from unshielded electric motors, transformers, and unshielded coils.
- If you use an FFC without a ground layer, keep the cable away from materials with a high dielectric loss.
- If you use an FFC with a ground layer, make sure to insert the cable with the ground tab facing down.
- When ordering a custom flexible flat cable, specify the FFC connector (see next section) to ensure that the ground tab is in the correct position and the retaining tabs have the correct shape.
- When using a standard flexible flat cable without a ground layer and retaining tabs, keep the cable short and avoid close proximity to other materials. Note that the pull-out force is reduced when using a cable without retaining tabs.

FFC Connector

- Basler recommends to use the same connector on camera and system side, although other connectors with the same number of pins can be used.
- The order code of the FFC connector on camera side is Hirose FH41-28S-0.5SH(05). This connector is optimized for impedance controlled flexible flat cables.
- Make sure that the FFC contact pads meet the mechanical specification of the Hirose FH41 connector to avoid shorting the GND layer and the data lines. For more information, see the *FH41 Series* connector data sheet. The download link is provided in Section 1.3 on page 2.

6.2 Trigger and General Purpose Input (CC0)

The LVDS input line can be used to send a trigger signal to the camera. Although the trigger frequency is very moderate, make sure to observe the LVDS design rules to avoid noise being picked up. Noise can cause false trigger events.

In addition, the LVDS input can be used as a general purpose input. The state of the input can be read via the camera's control interface at any time.

The input does not need a termination resistor. The line is terminated at camera side.

In other respects, the same recommendations as for the BCON data output lines apply (see Section 6.1 on page 44).

6.3 I²C Configuration Interface (I2C_SDC, I2C_SDA, I2C_ID)

Basler dart BCON cameras use the I²C interface to exchange configuration data with the controlling processor.

I²C is an open-collector bus interface with two signal lines, SDA and SCL. Connect these signal lines to the appropriate ports of an I²C controller. Typically, an I²C controller is part of the microcontroller or the System on a Chip (SoC) that you will be using to process the image data.

For more information about I²C, see the *I²C-Bus Specification and User Manual*. The download link is provided in Section 1.3 on page 2.

NOTICE
Voltage outside of the specified range can cause damage. You must supply camera power that complies with the individual voltage requirements of the BCON interface lines: ▪ The nominal voltage for the power supply line (VCC) is **5 V**. ▪ The nominal voltage for the I²C interface lines (I2C_SCL, I2C_SDA, I2C_ID) is **3.3 V**. For more information, see Section 4.3 on page 27.

Design Recommendations for the I²C Interface

- The voltage level for the I²C bus is 3.3 V. Make sure that the pull-up resistor is connected to 3.3 V.
- If you have only one or two cameras, tie the I2C_ID pin to GND on the first camera and to 3.3 V on the second camera. This helps you to identify which camera is connected to a given data interface.
- The capacitive bus load for a single I²C bus segment is limited to 400 pF. Take into account that long flexible flat cables can significantly contribute to the total bus load. If necessary, use more than one I²C bus or use an I²C bus repeater.
- Choose a suitable pull-up resistor. The optimal value depends on the capacitive load of the I²C bus lines. For determining appropriate resistor values, refer to the *I²C-bus Specification and User Manual*.

6.4 Power Supply (GND, VCC)

The camera needs a single 5 V power supply. Bad power quality can deteriorate the camera's image quality. However, a properly designed switched-mode power supply, as used to supply a USB port, will yield a good result.

Design Recommendations for the Power Supply

- The power supply should be able to supply 1.5 W per camera. Note that long cables can cause a significant voltage drop, leading to a higher power consumption than the camera alone.
- Ground noise can cause bad image quality. Make sure that the power supply input (VCC and GND) is close to the camera connectors. The processor and the memory should not be placed between the power supply input and the camera connector.
- Provide an extra capacitor directly on the camera connector where the supply voltage is delivered to the camera (e.g. 10 µF MLCC - multi-layer ceramic capacitor). If the supply is noisy, a low-pass-filter consisting of a capacitor and an inductor can help to smoothen the supply voltage.

6.5 LVDS Receiver

The BCON interface is designed to work with FPGAs as well as Channel Link deserializer devices. For full speed operation, the FPGA must support a bit rate of at least 560 Mb/s. The Channel Link deserializer device must support 28 bits and 84 MHz or more.

For a list of suitable Channel Link devices, see the *Channel Link Design Guide*. The download link is provided in Section 1.3 on page 2.

7 Installation

7.1 Hardware Installation

To install the camera:

1. Connect the flexible flat cable with the camera and the image processing system.
   To insert the cable:
   a. Push the end of the cable firmly into the connector with the ground tab facing down until the retaining tabs slide into the recesses on both sides.
   b. Flip the locking bar down to lock the connector.
2. Turn on the power supply.
   The camera's indicator LED lights up.
3. Check the power consumption of the camera. If the camera needs more than 1.5 W, turn the power off immediately and check for short circuits and miswirings.

7.2 Software Installation

To install the camera software:

1. Download the pylon Camera Software Suite for Linux (ARM) from www.baslerweb.com:
   a. If your ARM processor has a floating point unit, download the **hardfloat** version (file name pylon-x.x.x.xxxx-**armhf**.tar.gz).
   b. If your ARM processor doesn’t have a floating point unit, or if you are unsure, download the **softfloat** version (file name pylon-x.x.x.xxxx-**armel**.tar.gz).
2. Use an unzip tool of your choice to unpack the TAR archive to your home directory.
3. Change to the archive’s main directory and open the "INSTALL" text file.
4. Follow the instructions to install the pylon Camera Software Suite for Linux.
   If you want to operate dart BCON cameras only, you don’t need to install USB support.
5. For information about how to proceed, e.g. how to assign an I²C interface in pylon and how to operate your camera, see the *Programmer's Guide and API Reference for Basler pylon BCON* included in the pylon Camera Software Suite for Linux. The guide can be found in the unpacked archive, in the **pylon-5.0.x.yyyy-arch/doc/BCON/index.html** file.

8 I/O Control

8.1 Configuring Input Lines and Signals

8.1.1 Using the Input Line for Frame Start Triggering

The camera is equipped with one input line built into the BCON interface. This line is designated as CC0 as specified in Section 4.2.1 on page 24.

On the camera side, you can select this input to act as the source signal for the frame start trigger (also known as "hardware triggering").

Whenever a proper electrical signal is applied to the selected line, the camera will recognize the signal as signal for the frame start trigger.

The electrical signal must be appropriately timed.

For more information, see Section 9.2 on page 59.

8.1.2 Input Line Debouncers

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

i	We recommend setting the debouncer value so that it is slightly greater than the longest expected duration of an invalid signal. Setting the debouncer to a value that is ▪ **too short** will result in accepting invalid signals. ▪ **too long** will result in rejecting valid signals.

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

Figure 23 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.

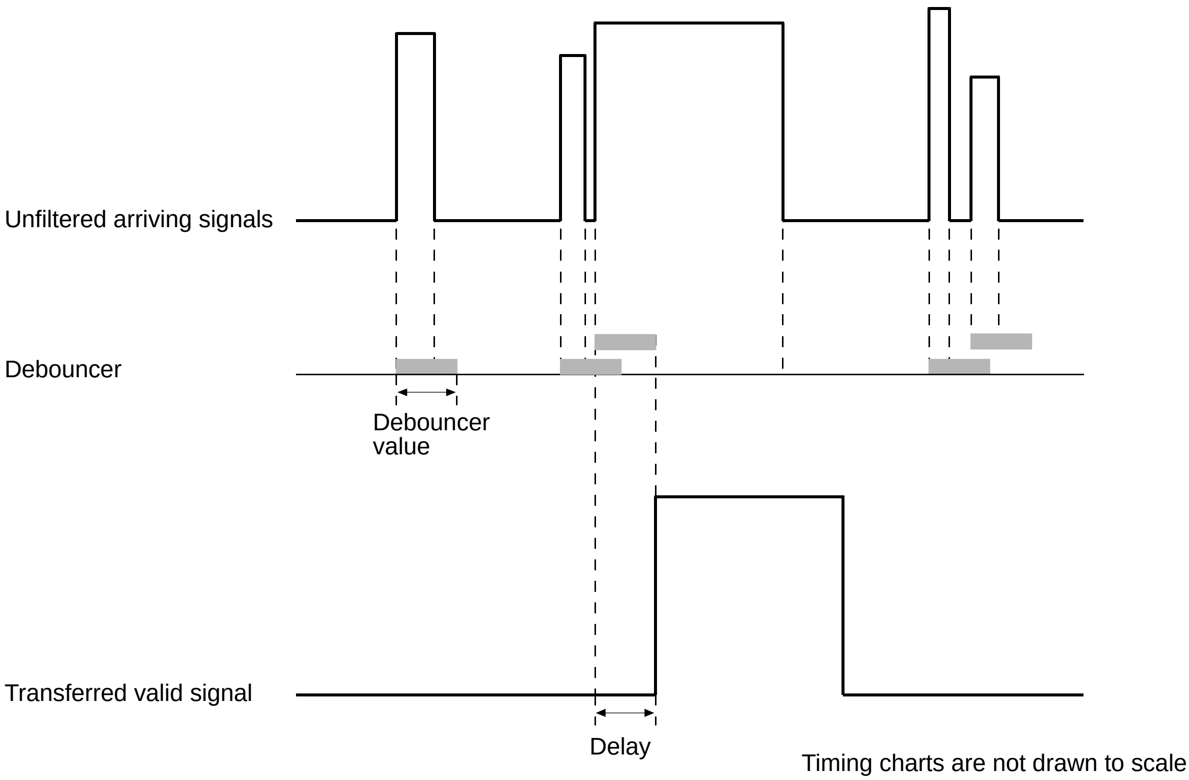


Fig. 23: Filtering of Input Signals by the Debouncer

Setting the Debouncer

You can set a debouncer value for the input line CC0.

The debouncer value is determined by the value of the LineDebouncerTime parameter value. The parameter is set in microseconds and can be set in a range from 0 to 10 µs.

To set the debouncer:

1. Use the LineSelector parameter to select input line CC0.
2. Set the LineDebouncerTime parameter to the desired debouncer time (in µs).

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

```
// Select the input line
camera.LineSelector.SetValue(LineSelector_CC0);
// Set the parameter value e.g. to 10 microseconds
camera.LineDebouncerTime.SetValue(10.0);
```

8.1.3 Input Line Inverter

You can set the input line CC0 to invert or not to invert the incoming electrical signal.

To set the invert function:

1. Use the LineSelector parameter to select input line CC0.
2. Set the value of the LineInverter parameter to true to enable inversion on the selected line or to false to disable inversion.

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

```
// Select the input line
camera.LineSelector.SetValue(LineSelector_CC0);
// Enable the inverter on the input line
camera.LineInverter.SetValue(true);
```

8.2 Configuring Output Lines and Signals

8.2.1 Selecting a Source Signal for an Output Line

In the BCON interface, two bits are reserved for camera output signals. In the pylon API, the two bits are represented as output lines Line 1 and Line 2.

To make an output line useful, you must select a source signal for the line. The camera has the following standard output signals available:

Output Signal	Valid for Camera Models
Flash Window	daA2500-14bm/bc
Exposure Active	daA1280-54bm/bc, daA1600-60bm/bc
User Output 1	All models
User Output 2	All models

Table 22: Available Output Signals

For more information about the output signals User Output 1 and User Output 2, see Section 8.2.2 on page 53.

To set a camera output signal as the source signal for an output line:

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

You can set the LineSelector and the LineSource parameter values from within your application software by using the Basler pylon API.

The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the Flash Window signal as the source signal for the BCON output line 1
camera.LineSelector.SetValue(LineSelector_Line1);
camera.LineSource.SetValue(LineSource_FlashWindow);
```

For more information about

- the flash window signal, see Section 9.6.2 on page 76.
- setting the status of a user settable output line, see Section 8.2.2 on page 53.

8.2.2 Setting the Status of a User Settable Output Line

The output lines can be designated as "user settable", which means that you can assign a state (high or low) to them using the UserOutputValue parameter.

You can use this to control external events or devices, e.g. a light source.

i	If you have the line inverter enabled on an output line and if the line is designated as user settable, the user setting initially sets the status of the line which is then inverted by the line inverter.

Two user settable output signals are available: User Output 1 and User Output 2. For more information about output signals, see Section 8.2.1 on page 53

To designate an output line as user settable:

1. Use the LineSelector parameter to select output line Line 1 or Line 2.
2. Set the LineSource parameter to UserOutput1 or UserOutput2.

To set the status of a user settable output line:

1. Use the UserOutputSelector parameter to select output line Line 1 or Line 2.
2. Set the value of the UserOutputValue parameter to true (1) or false (0). This will set the state of the output line.

You can set the parameters from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to designate an output line as user settable, set the status of the output line, and get informed about its current status:

```
// Set output line Line 1 to user settable
camera.LineSelector.SetValue(LineSelector_Line1);
camera.LineSource.SetValue(LineSource_UserOutput1);
// Set the status of output line 2
camera.UserOutputSelector.SetValue(UserOutputSelector_UserOutput2);
camera.UserOutputValue.SetValue(true);
// Get informed about the current user output value setting for output line 2
bool b = camera.UserOutputValue.GetValue();
```

8.2.3 Output Line Inverter

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

To set the invert function for an output line:

1. Use the LineSelector parameter to select output line Line 1 or Line 2.
2. Set the value of the LineInverter parameter
   - to true to enable inversion on the selected line
   - to false to disable inversion on the selected line.

You can set the LineSelector and the LineInverter 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 line inverter on output line Line 1
camera.LineSelector.SetValue(LineSelector_Line1);
camera.LineInverter.SetValue(true);
```

8.3 Checking the Status of the I/O Lines

8.3.1 Checking the Status of an I/O Line

You can determine the current status of any input or output line. The status depends on whether an electrical signal is present to the line and whether the line inverter is enabled.

To check the status of an I/O line:

1. Use the LineSelector parameter to select, for example, input line CC0.
2. Read the value of the LineStatus parameter to determine the current status of the line. A value of true means the line's status is currently high and a value of false means the line's status is currently low.

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

```
// Select input line CC0 and read the status
camera.LineSelector.SetValue(LineSelector_CC0);
bool b = camera.LineStatus.GetValue();
// Select BCON output line 1 line and read the status
camera.LineSelector.SetValue(LineSelector_Line1);
bool b = camera.LineStatus.GetValue();
```

8.3.2 Checking the Status of All Lines

Reading the LineStatusAll parameter value allows you to check the current status of all input and output lines.

The Line Status All parameter is a 32-bit value. As shown in Figure 24, certain bits in the value are associated with each I/O line, and each of these bits will indicate the state of the associated line.

If a bit is 0, it indicates that the state of the associated line is currently low. If a bit is 1, it indicates that the state of the associated line is currently high.

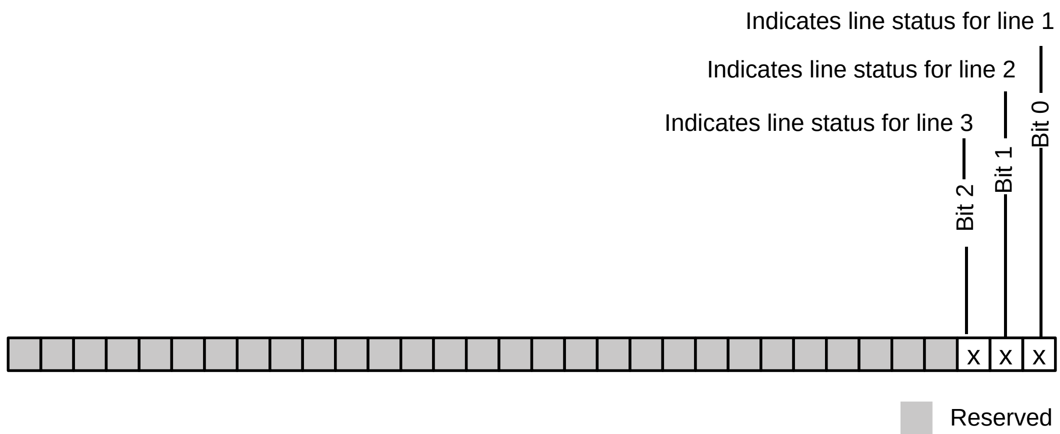


Fig. 24: Bit Field of the LineStatusAll Parameter: Bit Numbers and Assignment of I/O Lines

The following table refers to the bit field shown in Figure 24 and lists all possible LineStatusAll parameter values (hexadecimal numbers) and related binary numbers.

LineStatusAll Parameter Value		Binary Expression of the LineStatusAll Parameter Value		
Hexadecimal Number	**Binary Number**	**Line 1 (Output Line)**	**Line 2 (Output Line)**	**Line 3 (Input Line)**
0x0	000	0	0	0
0x1	001	1	0	0
0x2	010	0	1	0
0x3	011	1	1	0
0x4	100	0	0	1
0x5	101	1	0	1
0x6	110	0	1	1
0x7	111	1	1	1

Table 1: LineStatusAll Parameter Values and Corresponding Binary Expressions

To check the status of all I/O lines with a single operation:

1. Read the value of the LineStatusAll parameter to determine the current status of both I/O lines.

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

```
// Read the line status of both I/O lines. Because the GenICam interface does not
// support 32-bit words, the line status is reported as a 64-bit value.
int64_t i = camera.LineStatusAll.GetValue();
```

9 Image Acquisition Control

9.1 Acquisition Start and Stop Commands and the Acquisition Mode

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

Executing an **AcquisitionStop** command terminates the camera's ability to acquire frames. When the camera receives an AcquisitionStop command and if the camera

- is currently not acquiring a frame, the camera stops acquiring frames immediately.
- is currently acquiring a frame, the frame acquisition process will be allowed to finish and the camera's ability to acquire new frames will be terminated.

The camera's **AcquisitionMode** parameter has two settings: single frame and continuous. The use of AcquisitionStart and AcquisitionStop commands and the camera's AcquisitionMode parameter setting are related.

If the camera's AcquisitionMode parameter

- is set for **single frame**, after an AcquisitionStart command has been executed, a single frame can be acquired. When acquisition of one frame is complete, the camera will execute an AcquisitionStop command internally and will no longer be able to acquire frames. To acquire another frame, you must execute a new AcquisitionStart command.
- is set for **continuous frame**, after an AcquisitionStart command has been executed, frame acquisition can be triggered as desired. Each time a frame trigger is applied while the camera is in a "waiting for frame trigger" acquisition status, the camera will acquire and transmit a frame. The camera will retain the ability to acquire frames until an AcquisitionStop command is executed. Once the AcquisitionStop command is received, the camera will no longer be able to acquire frames.

When the camera's acquisition mode is set to single frame, the maximum possible acquisition frame rate for a given ROI can't be achieved. This is because the camera performs a complete internal setup cycle for each single frame and because it can't be operated with overlapped exposure.

For more information about overlapped image acquisitions, see Section 9.5 on page 73.

Setting the Acquisition Mode and Issuing Start/Stop Commands

You can set the AcquisitionMode parameter value and you can execute AcquisitionStart or AcquisitionStop commands from within your application software by using the Basler pylon API. The code snippet below illustrates using the API to set the AcquisitionMode parameter value and to execute an AcquisitionStart command, where line 1 is taken as an example.

The snippet also illustrates setting several parameters regarding frame triggering. These parameters are discussed later in this chapter.

```
camera.AcquisitionMode.SetValue(AcquisitionMode_SingleFrame);
camera.TriggerMode.SetValue(TriggerMode_On);
camera.TriggerSource.SetValue(TriggerSource_CC0);
camera.TriggerActivation.SetValue(TriggerActivation_RisingEdge);
camera.ExposureMode.SetValue(ExposureMode_Timed);
camera.ExposureTime.SetValue(3000.0);
camera.AcquisitionStart.Execute();
```

9.2 The Frame Start Trigger

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

In order for the camera to be in a "waiting for frame start trigger" acquisition status:

- The AcquisitionMode parameter must be set correctly.
- A proper AcquisitionStart command must be applied to the camera.

For more information about the AcquisitionMode parameter and about AcquisitionStart and AcquisitionStop commands, see Section 9.1 on page 57.

There are three ways to operate the camera using frame start trigger signals:

- Frame start trigger signals are generated internally by the camera, and frame acquisition will be done automatically. This is also known as the **"free run"**. For more information, see Section 9.2.1.1 on page 59.
- Frame start trigger signals are applied via **software**. Each time a TriggerSoftware command is executed via the pylon API, the camera will begin a frame acquisition. For more information, see Section 9.2.2 on page 62.
- Frame start trigger signals are applied via **hardware**. Each time a proper electrical signal is applied to the input line, the camera will begin a frame acquisition. For more information, see Section 9.2.3 on page 63.

9.2.1 Trigger Mode

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

9.2.1.1 TriggerMode = Off (Free Run)

When the TriggerMode parameter is set to Off, the camera will generate all required frame start trigger signals internally, and you do not need to apply frame start trigger signals to the camera. This status is also known as "free run".

With the trigger mode set to Off, the way the camera will operate the frame start trigger depends on the setting of the camera's AcquisitionMode parameter. If the AcquisitionMode parameter is set to

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

The rate at which the frame start trigger signals are generated in the continuous frame Acquisition Mode can be determined by the camera's AcquisitionFrameRate parameter:

Acquisition Frame Rate	Frame Start Trigger Rate
Acquisition frame rate < maximum allowed frame rate	Acquisition frame rate
Acquisition frame rate >=maximum allowed frame rate	Maximum allowed frame rate

Table 23: Frame Start Trigger Rates in the Continuous Frame Acquisition Mode

For more information about determining the maximum allowed frame rate, see Section 9.7 on page 78.

Exposure Time Control with Trigger Mode Set to Off

When the TriggerMode parameter is set to Off, the exposure time for each frame acquisition is determined by the value of the camera's ExposureTime parameter.

For more information about the camera's ExposureTime parameter, see Section 9.3 on page 67.

Configuring and Enabling the Free Run Mode

The following code snippet illustrates using the API to set the Acquisition Mode to Continuous, the Trigger Mode to Off (free run), and the Acquisition Frame Rate to 60:

```
// Set the acquisition mode to Continuous
camera.AcquisitionMode.SetValue(AcquisitionMode_Continuous);
// Set the timed exposure mode
camera.ExposureMode.SetValue(ExposureMode_Timed);
// Set the exposure time
camera.ExposureTime.SetValue(3000.0);
// Set the frame rate
camera.AcquisitionFrameRate.SetValue(60.0);
// Enable free run mode by setting the trigger mode to Off
camera.TriggerMode.SetValue(TriggerMode_Off);
// Start frame capture
camera.AcquisitionStart.Execute();
```

9.2.1.2 TriggerMode = On (Software or Hardware Triggering)

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

> Do not trigger frame acquisition at a rate that
>
> - exceeds the maximum allowed for the current camera settings:
>   - If you apply frame start trigger signals in **timed** exposure mode and the camera is not ready to receive them, the signal will be ignored.
>   - If you apply frame start trigger signals in **trigger width** exposure mode and the camera is not ready to receive them, the signal may be partly ignored.
>
>   For more information, see Section 9.2.3 on page 63.
> - exceeds the target system's capacity limits for data transfer or storage or both. If you try to acquire more images than the target system is able to process, frames may be dropped. For more information about bandwidth optimization, see the *Installation and Setup Guide for Cameras Used with Basler pylon for Windows* (AW000611).

The TriggerSource parameter specifies the source signal that will act as the frame start trigger signal. The available selections for the TriggerSource parameter are:

- **Software** - When the source signal is set to software, you apply a frame start trigger signal to the camera by executing a TriggerSoftware command via pylon API.
- **CC0** -When the source signal is set to CC0, you apply a frame start trigger signal to the camera by injecting a hardware trigger signal into line CC0 in the BCON interface.

If the TriggerSource parameter is set to CC0, you must also set the TriggerActivation parameter. The available settings for the TriggerActivation parameter are:

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

For more information about

- using a software trigger, see Section 9.2.2 on page 62.
- using a hardware trigger, see Section 9.2.3 on page 63.

Exposure Time Control with Trigger Mode Set to On

If the TriggerMode parameter is set to On and the TriggerSource parameter is

- set to **Software**, the exposure time for each frame acquisition is determined by the value of the camera's ExposureTime parameter.
- set to **CC0**, the exposure time for each frame acquisition can be controlled with the ExposureTime parameter or it can be controlled by manipulating the hardware trigger signal.

For more information about

- controlling exposure time when using a software trigger, see Section 9.2.2 on page 62.
- controlling exposure time when using a hardware trigger, see Section 9.2.3 on page 63.

Immediate Trigger Mode

Available for ...
daA1280-54bm/bc, daA2500-14bm/bc

If the TriggerMode parameter is set to On, the immediate trigger mode setting also takes effect.

You can enable or disable the immediate trigger mode by setting the ImmediateTriggerMode parameter to one of the following values:

- **On:** Exposure starts immediately after triggering, but changes to image parameters become effective with a short delay, i.e., after one or more images have been acquired. This is useful if you want to minimize the exposure start delay, i.e., if you want to start image acquisition as soon as possible, and if your imaging conditions are stable.
- **Off:** Changes to image parameters become effective immediately, but exposure starts with a short delay after triggering.

By default, the parameter is set to Off.

9.2.2 Using a Software Frame Start Trigger

If the TriggerMode parameter is set to On and the TriggerSource parameter is set to Software,

- you must apply a software frame start trigger signal to the camera to begin each frame acquisition.
- the camera's Exposure Mode parameter must be set to Timed. The exposure time is set by the ExposureTime parameter.
- the frame rate will be determined by how often you apply a software trigger signal to the camera.

Configuring and Executing a Software Frame Start Trigger Command

The following code snippet illustrates using the API to set the parameter values and to execute the commands related to software frame start triggering with the camera set for continuous frame acquisition mode:

```
// Set the acquisition mode to Continuous
camera.AcquisitionMode.SetValue(AcquisitionMode_Continuous);
// Set the trigger mode to On
camera.TriggerMode.SetValue(TriggerMode_On);
// Enable software triggering
camera.TriggerSource.SetValue(TriggerSource_Software);
// Set the immediate trigger mode to Off
camera.ImmediateTriggerMode.SetValue(ImmediateTriggerMode_Off);
// Set the timed exposure mode
camera.ExposureMode.SetValue(ExposureMode_Timed);
// Set the exposure time
camera.ExposureTime.SetValue(3000.0);
// Execute an AcquisitionStart command to prepare for frame acquisition
camera.AcquisitionStart.Execute();
     while (!finished)
     {
        // Execute a TriggerSoftware command to apply a frame start
        // trigger signal to the camera
        camera.TriggerSoftware.Execute();
        // Retrieve acquired frame here
     }
camera.AcquisitionStop.Execute();
```

9.2.3 Using a Hardware Frame Start Trigger

If the TriggerMode parameter is set to On and the TriggerSource parameter is set to Line 3, an externally generated electrical signal injected into input line CC0 will act as the frame start trigger signal for the camera.

This type of trigger signal is generally referred to as a hardware trigger signal or as a hardware frame start trigger signal.

A rising edge or a falling edge of the hardware trigger signal can be used to trigger frame acquisition.

If you are using hardware trigger signals, the period of the hardware trigger signal will determine the rate at which the camera is acquiring frames:

$$\frac{1}{\text{ExFSTrig period in seconds}} = \text{frame rate}$$

For example, if you are operating a camera with a hardware trigger signal period of 20 ms (= 0.02 s), the frame rate is 50 fps:

$$\frac{1}{0.02} = 50$$

When adjusting the settings, make sure not to exceed the maximum allowed frame rate. If the maximum allowed frame rate is exceeded, the behavior of the camera can be undefined.

For more information about determining the maximum allowed frame rate, see Section 9.7 on page 78.

Two exposure modes are available for hardware triggering: **timed** and **trigger width**.

Timed Exposure Mode

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

If the camera is set for

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

Figure 25 illustrates timed exposure with the camera set for rising edge triggering.

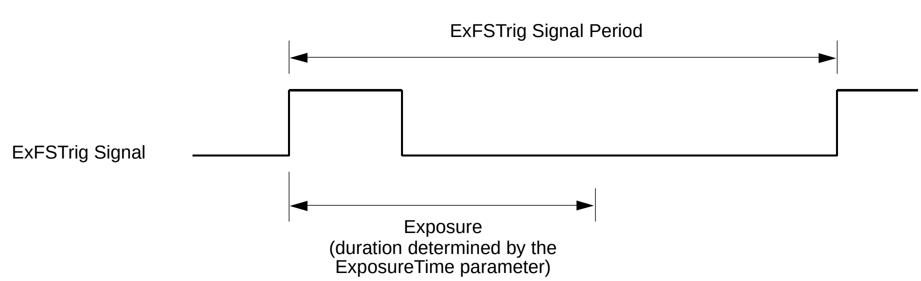


Fig. 25: Timed Exposure with Rising Edge Triggering

If you attempt to trigger a new exposure start while the previous exposure is still in progress, the trigger signal will be ignored. This situation is illustrated in Figure 26 for rising edge triggering.

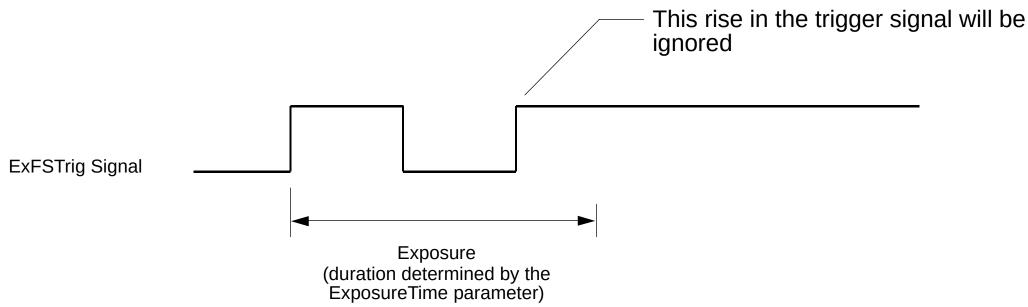


Fig. 26: Overtriggering with Timed Exposure

For more information about the camera's ExposureTime parameter, see Section 9.3 on page 67.

Trigger Width Exposure Mode

Available for ...
daA2500-14bm/bc

When the trigger width exposure mode is selected, the length of the exposure for each frame acquisition will be directly controlled by the ExFSTrig signal. Trigger width exposure is especially useful if you intend to vary the length of the exposure time for each captured frame.

If the camera is set for

- **rising edge triggering**, the exposure time begins when the ExFSTrig signal rises and continues until the ExFSTrig signal falls.
- **falling edge triggering**, the exposure time begins when the ExFSTrig signal falls and continues until the ExFSTrig signal rises.

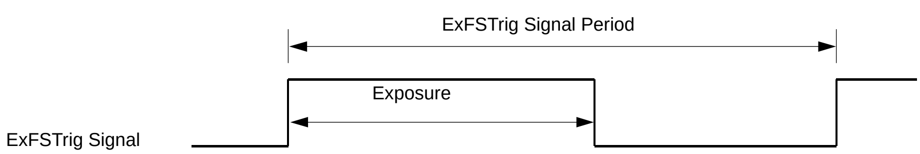


Fig. 27: Trigger Width Exposure with Rising Edge Triggering

If you attempt to trigger a new exposure start while the previous frame acquisition is still in progress, the trigger signal may be partly ignored. Therefore, the exposure time for the new frame may be shorter than the trigger signal. This situation is illustrated in Figure 28 for rising edge triggering.

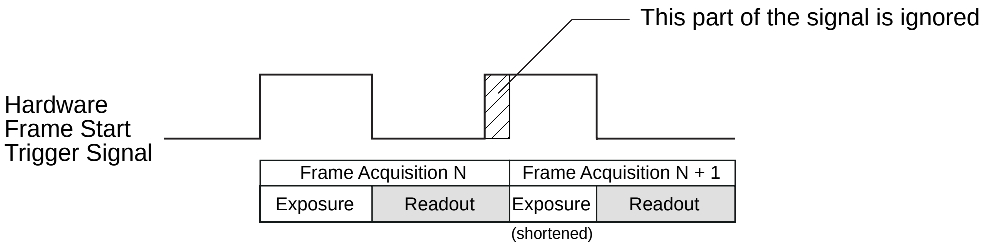


Fig. 28: Overtriggering with Trigger Width Exposure

Configuring and Executing a Hardware Frame Start Trigger Command

You can set all of the parameters needed to perform hardware frame start triggering from within your application by using the Basler pylon API. The following code snippet illustrates using the API to set the camera for single frame acquisition mode. In this example, the timed exposure mode will be used with rising edge triggering:

```
// Set the acquisition mode to single frame
camera.AcquisitionMode.SetValue(AcquisitionMode_SingleFrame);
// Set the trigger mode to On
camera.TriggerMode.SetValue(TriggerMode_On);
// Set the immediate trigger mode to Off
camera.ImmediateTriggerMode.SetValue(ImmediateTriggerMode_Off);
// Set the source for the selected trigger
camera.TriggerSource.SetValue(TriggerSource_CC0);
// Set the trigger activation mode to rising edge
camera.TriggerActivation.SetValue(TriggerActivation_RisingEdge);
// Set for the timed exposure mode
camera.ExposureMode.SetValue(ExposureMode_Timed);
// Set the exposure time
camera.ExposureTime.SetValue(3000.0);
// Execute an AcquisitionStart command to prepare for frame acquisition
camera.AcquisitionStart.Execute();
// Frame acquisition will start when the externally generated
// frame start trigger signal (ExFSTrig signal) goes high
```

9.3 Setting the Exposure Time

This section describes how the exposure time can be adjusted manually by setting the value of the ExposureTime parameter. The camera also has an exposure auto function that can automatically adjust the exposure time.

By default, the exposure auto function is enabled. Manual adjustment of the exposure time will not work.

Set the ExposureAuto parameter to Off before making any manual adjustments.

For more information about the exposure auto function, see Section 11.11.4 on page 117.

Camera Model	Minimum Allowed Exposure Time	Maximum Possible Exposure Time	Increment
daA1600-60bm/bc	10 µs	850000 µs	1 µs
All other models	10 µs	1000000 µs	1 µs

Table 24: Minimum and Maximum Allowed Exposure Time Setting and Increment

Depending on the camera's sensor and the frame rate, the effective exposure time may vary from the exposure time set. The variation is normally in the range of microseconds, but if a very short exposure time is set, this should be taken into account.

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

```
// Set the exposure time to 40.0 µs
camera.ExposureTime.SetValue(40.0);
```

9.4 Electronic Shutter Operation

All dart BCON cameras are equipped with imaging sensors that have an electronic shutter. There are two types of electronic shutter modes used in the sensors: **global** and **rolling**. For rolling shutter, there are two sub-types: electronic rolling shutter (ERS) and global reset release mode (GRR).

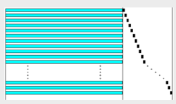
Global Shutter 	**Electronic Rolling Shutter (ERS)** 	**Global Reset Release Mode (GRR)** 
For moving objects	■ For stationary objects/**not** moving objects ■ Lower ambient noise ■ If used for moving objects: Use of flash lighting and flash window recommended	■ For stationary objects/**not** moving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

Table 25: Overview of Shutter Modes

Camera Model	Shutter Type	Supported Shutter Modes
daA1280-54bm/bc daA1600-60bm/bc	Global Shutter	Global Shutter
daA2500-14bm/bc	Rolling Shutter	Electronic Rolling Shutter (ERS) Global Reset Release Mode (GRR)

Table 26: Camera Models and Shutter Configuration

9.4.1 Global Shutter

Camera Model	Global Shutter Available?
daA1280-54bm/bc daA1600-60bm/bc	Yes
daA2500-14bm/bc	No

Table 27: Global Shutter Availability

A main characteristic of a global shutter is that for each frame acquisition, all of the pixels in the sensor start exposing at the same time and all stop exposing at the same time.

This means that image brightness tends to be more uniform over the entire area of each acquired image, and it helps to minimize problems with acquiring images of objects in motion.

Immediately after the end of exposure, pixel data readout begins and proceeds in a linewise fashion until all pixel data is read out of the sensor.

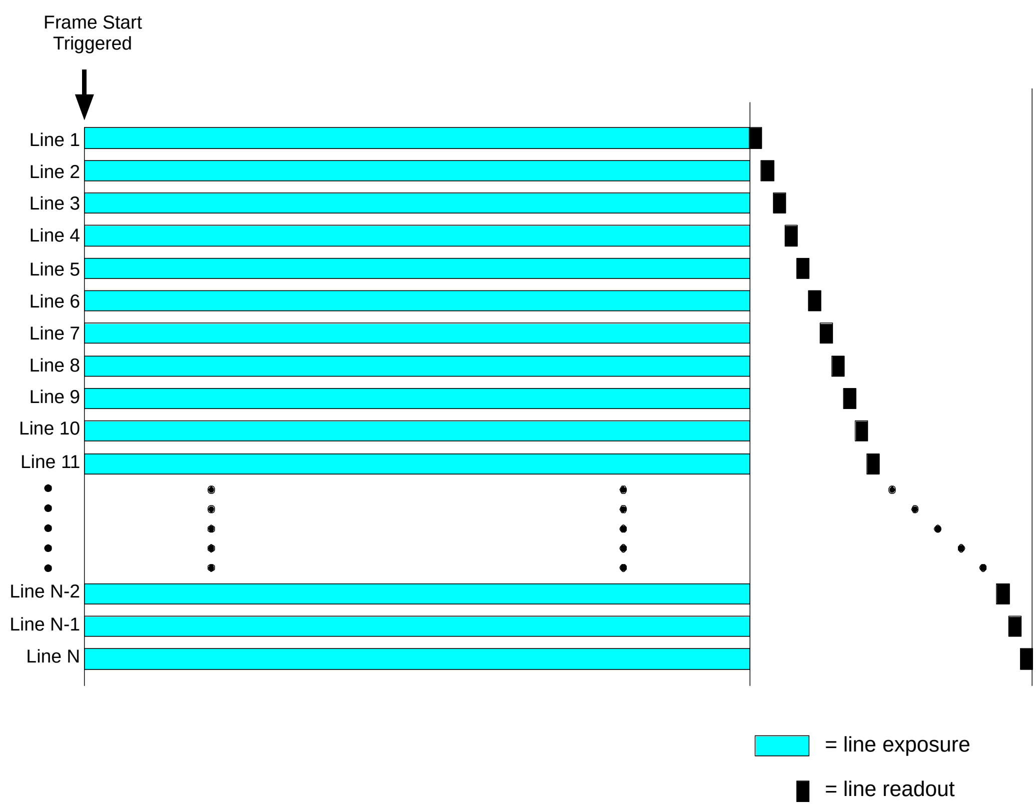


Fig. 29: Global Shutter

9.4.2 Rolling Shutter

Camera Model	Rolling Shutter Available?
daA1280-54bm/bc daA1600-60bm/bc	No
daA2500-14bm/bc	Yes

Table 28: Rolling Shutter Availability

The rolling shutter is used to control the start and stop of sensor exposure. The rolling shutter used in these cameras has two operating modes:

- electronic rolling shutter mode (ERS mode) and
- global reset release mode (GRR mode).

Electronic Rolling Shutter Mode (ERS)

When the shutter is in the electronic rolling shutter operating mode, it exposes and reads out the pixel lines with a temporal offset from one line to the next. When frame start is triggered, the camera

- resets line one of the ROI and begins exposing line one,
- resets line two a short time later (= temporal offset) and begins exposing line two,
- resets line three a short time later than line two (= additional temporal offset) and begins exposing line three.

And so on until the bottom line of pixels is reached (see Figure 30).

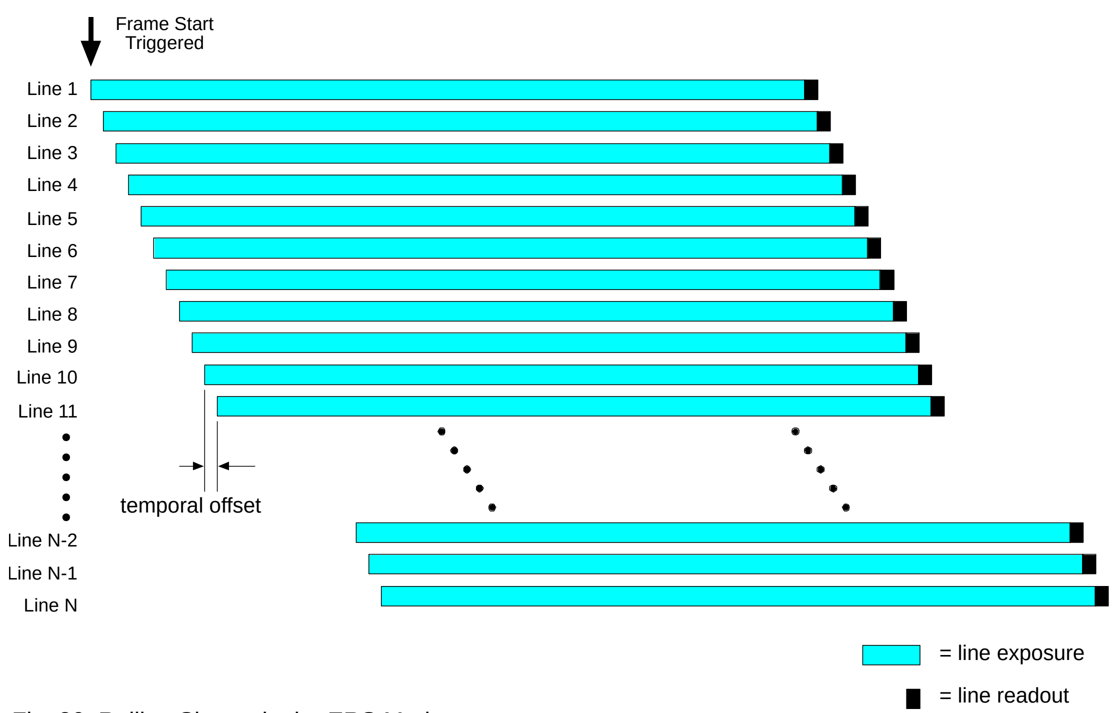


Fig. 30: Rolling Shutter in the ERS Mode

The pixel values for each line are read out at the end of exposure time for the line. The exposure time is the same for all lines and is determined by the ExposureTime parameter setting.

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

Global Reset Release Mode (GRR)

In the global reset release mode, all of the lines in the sensor reset and begin exposing when frame start is triggered. There is a temporal offset from one line to the next in the end of exposure. The exposure time

- for line one is determined by the ExposureTime parameter setting.
- for line two will end a short time (= temporal offset) after the exposure ends for line one.
- for line three will end another short time (= temporal offset) after the exposure ends for line two.

And so on until the bottom line of pixels is reached (see Figure 31).

The pixel values for each line are read out at the end of exposure time for the line.

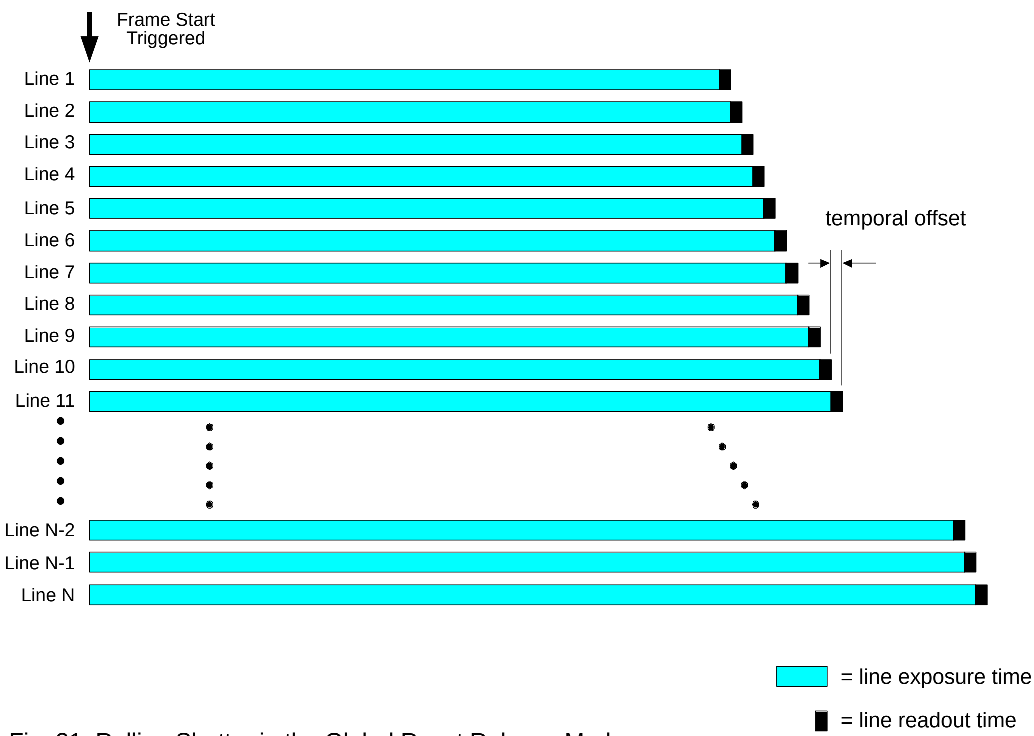


Fig. 31: Rolling Shutter in the Global Reset Release Mode

When the camera is operating with the rolling shutter in the global release mode, the use of flash lighting is most strongly recommended. For more information, see Section "Rolling Shutters and Flash Exposure" in this chapter.

Setting the Sensor Shutter Mode

You can set the sensor shutter mode (electronic rolling shutter or global reset release) from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to set the sensor shutter modes:

```
// Set the electronic rolling shutter mode
camera.SensorShutterMode.SetValue(SensorShutterMode_Rolling);

// Set the global reset release shutter mode
camera.SensorShutterMode.SetValue(SensorShutterMode_GlobalReset);
```

Rolling Shutters and Flash Exposure

If you are using

- the **electronic rolling shutter mode**, you should use flash exposure for capturing images of moving objects.
- the **global reset release mode**, you should use flash exposure for capturing images of both stationary and moving objects.

If you don't use flash exposure when capturing images of

- **stationary** objects, the brightness in each acquired image will vary significantly from top to bottom due to the differences in the exposure times of the lines.
- **moving** objects, the brightness in each acquired image will vary significantly from top to bottom due to the differences in the exposure times of the lines and the images will be distorted due to the temporal shift between the end of exposure for each line.

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

Cameras with a rolling shutter imaging sensor can provide a flash window output signal to aid you in the use of flash lighting. The flash window signal will go high when the flash window for each image acquisition opens and will go low when the flash window closes.

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

For more information about the flash window signal, see Section 9.6.2 on page 76.

9.5 Overlapping Image Acquisitions

There are two common ways for the camera to operate:

- with “non-overlapped” exposure and
- with “overlapped” exposure.

In the **non-overlap** mode of operation, each time a frame is acquired the camera completes the entire readout process (exposure of the pixels + readout of the pixel values from the sensor) before acquisition of the next frame is started. The exposure for a new frame does not overlap the sensor readout for the previous frame. This situation is illustrated in Figure 32 with the camera set for the trigger width exposure mode.

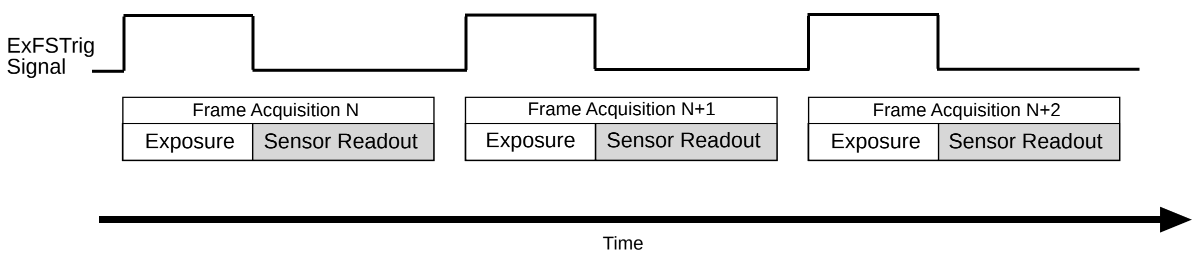


Fig. 32: Non-overlapped Exposure and Sensor Readout

In the **overlap** mode of operation, the exposure of a new frame begins while the camera is still reading out the sensor data for the previously acquired frame. This situation is illustrated in Figure 33 with the camera set for the trigger width exposure mode.

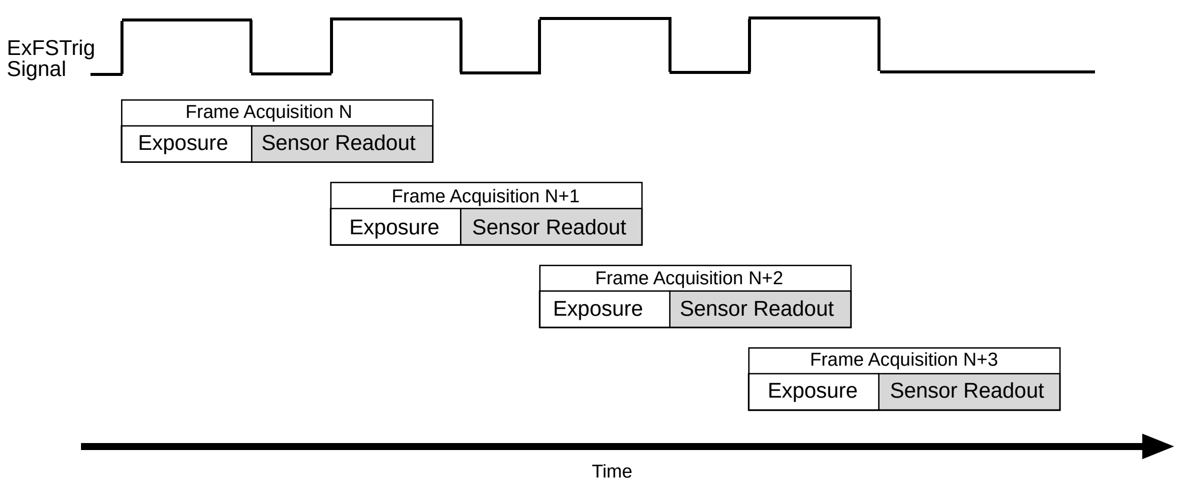


Fig. 33: Overlapped Exposure and Sensor Readout

9.5.1 Automatic Overlapping of Image Acquisitions (All Cameras)

If the camera allows overlapped image acquisitions (see Table 29), it will automatically overlap exposures and readouts to maximize the frame rate or to achieve a specific frame rate.

If the camera does not allow overlapped image acquisitions, the camera's maximum allowed frame rate will be limited by the duration of the frame acquisition process (exposure + readout).

To allow overlapping image acquisitions, the camera must be in the continuous acquisition mode. Overlapping image acquisition can't be performed if the camera's acquisition mode is set to single frame.

For more information about the acquisition mode, see Section 9.1 on page 57.

Camera Model	Rolling Shutter Mode	Trigger Mode	Overlapping Image Acquisitions allowed?
daA1280-54bm/bc daA1600-60bm/bc	Not applicable (global shutter camera)	On (software / hardware)	**No**
		Off (free run)	**Yes (*)**
daA2500-14bm/bc	ERS	On (software / hardware)	**Yes**
	ERS	Off (free run)	**Yes**
	GRR	On (software / hardware)	**No**
	GRR	Off (free run)	**No**
(*) Overlapping image acquisitions are allowed unless you manually disable them using the OverlapMode parameter. For more information, see Section 9.5.2 on page 75.			

Table 29: Conditions for Overlapping Image Acquisitions

For more information about

- rolling shutter modes, see Section 9.4.2 on page 70.
- trigger modes, see Section 9.2.1 on page 59.

9.5.2 Manually Setting the Overlap Mode of Operation (daA1280-54bm/bc, daA1600-60bm/bc)

On daA1280-54bm/bc and daA1600-60bm/bc cameras, you can use the OverlapMode parameter to manually disable or enable overlapping image acquisitions.

If the OverlapMode parameter is set to

- **On**, the sensor is put in the **overlap** mode of operation. The camera will automatically overlap exposures and readouts in the free run mode.
- **Off**, the sensor is put in the **non-overlap** mode of operation. The camera will never overlap exposures and readouts. This can improve image quality, especially when you are operating the camera at low frame rates.

For more information about the overlap and non-overlap mode of operation, see Section 9.5 on page 73.

The following code snippet illustrates using the pylon API to set the OverlapMode parameter value:

```
// Set for the overlapping mode of operation
camera.OverlapMode.SetValue(OverlapMode_On);

// Set for the non-overlapping mode of operation
camera.OverlapMode.SetValue(OverlapMode_Off);
```

9.6 Acquisition Monitoring Tools

9.6.1 Exposure Active Signal

Camera Model	Exposure Active Signal Available?
daA1280-54bm/bc daA1600-60bm/bc	Yes
daA2500-14bm/bc	No

Table 30: Exposure Active Availability

Cameras with a global shutter imaging sensor provide an "exposure active" (ExpAc) output signal.

The signal goes high when the exposure time for each frame acquisition begins and goes low when the exposure time ends as shown in Figure 34. 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.

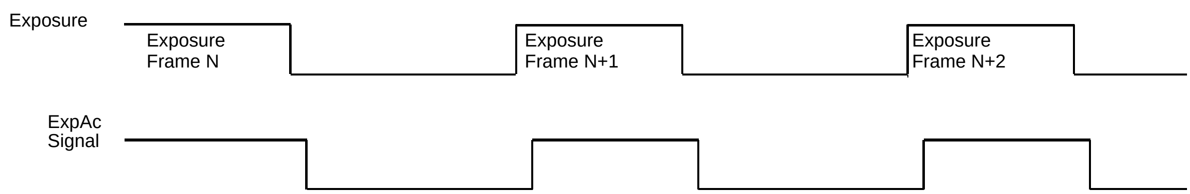


Fig. 34: Exposure Active Signal on Cameras with a Global Shutter

When you use the exposure active signal, be aware that there is a delay (in the range of microseconds) in the rise and the fall of the signal in relation to the start and the end of exposure.

To select the Exposure Active Signal as the source signal for an output line:

1. Use the LineSelector parameter to select output line Line 1 or Line 2.
2. Set the value of the LineSource parameter to the exposure active output signal.

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

```
camera.LineSelector.SetValue(LineSelector_Line1);
camera.LineSource.SetValue(LineSource_ExposureActive);
```

For more information about changing the selection of an output signal as the source signal for the output line, see Section 8.2.1 on page 53.

9.6.2 Flash Window Signal

Camera Model	Flash Window Signal Available?
daA1280-54bm/bc daA1600-60bm/bc	No
daA2500-14bm/bc	Yes

Table 31: Flash Window Availability

Cameras with a rolling shutter imaging sensor provide a flash window output signal to aid you in the use of flash lighting.

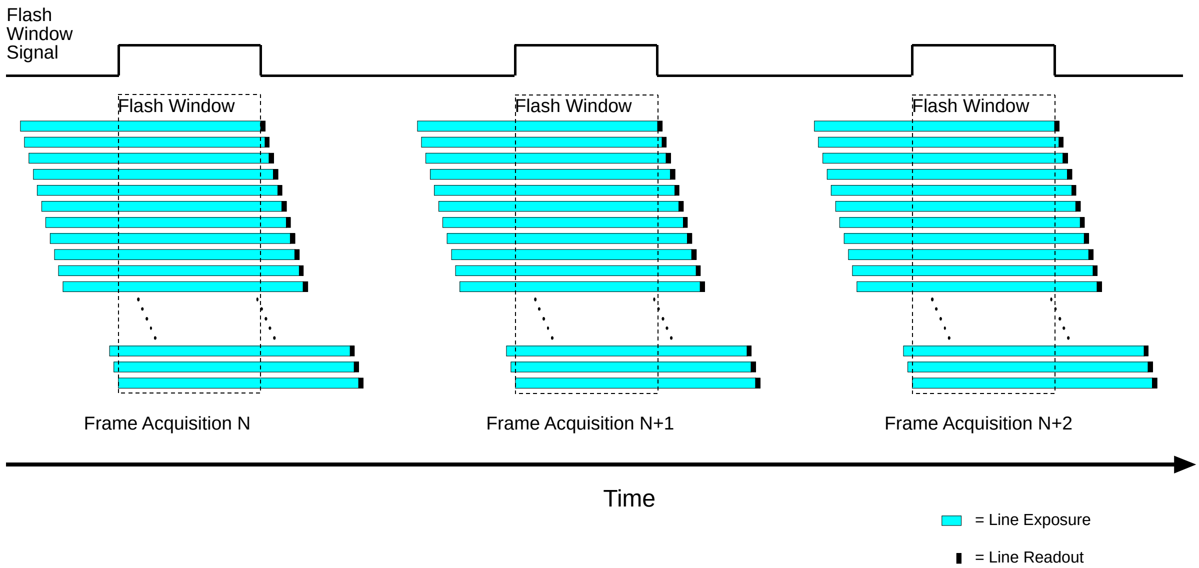


Fig. 35: Flash Window Signal on Cameras with a Rolling Shutter

The flash window signal will go high when the flash window for each image acquisition opens and will go low when the flash window closes. Figure 35 illustrates the flash window signal on a camera with the shutter operating in the electronic rolling shutter mode.

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

For more information about the rolling shutter and the flash window, see Section 9.4.2 on page 70.

To select the Flash Window Signal as the source signal for an output line:

1. Use the LineSelector parameter to select output line Line 1 or Line 2.
2. Set the value of the LineSource parameter to the flash window signal.

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

```
camera.LineSelector.SetValue(LineSelector_Line1);
camera.LineSource.SetValue(LineSource_FlashWindow);
```

For more information about changing the selection of an output signal as the source signal for the output line, see Section 8.2.2 on page 53.

9.7 Maximum Allowed Frame Rate

In general, the maximum allowed acquisition frame rate on any dart BCON camera can be limited by these factors:

- The exposure time for the acquisition of frames. If you use very long exposure times, you can acquire fewer frames per second.
- The amount of time it takes to read an acquired frame out of the imaging sensor and to prepare it for transmission out of the camera. The amount of time varies with the size of the frame. Frames with a smaller height take considerably less time. Frames with a smaller width may also take slightly less time.
- The amount of time it takes to transmit an acquired frame from the camera to the target system. The amount of time depends on the target system's capacity limits for data transfer and the bandwidth assigned to the camera.
- Under certain conditions, overlapping image acquisition is not possible. This decreases the camera's maximum allowed frame rate. For more information about overlapping image acquisitions, see Section 9.5 on page 73.

To determine the maximum allowed acquisition frame rate with your current camera settings, use the Basler pylon API to read the value of the camera's ResultingFrameRate parameter. For more information, see Section 9.7.1 on page 78.

When the camera's acquisition mode is set to single frame, the maximum possible acquisition frame rate can't be achieved. This is because the camera performs a complete internal setup cycle for each single frame and because it can't be operated with overlapped exposure.

For more information about overlapped image acquisitions, see Section 9.5 on page 73.

9.7.1 Using the Basler pylon API to Check the Maximum Allowed Frame Rate

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

```
// Get the resulting frame rate
double d = camera.ResultingFrameRate.GetValue();
```

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

9.7.2 Increasing the Maximum Allowed Frame Rate

If you want to acquire frames at a rate higher than the maximum allowed with the camera's current settings, you must adjust one or more of the factors that can influence the maximum allowed rate.

- Decreasing the height of the region of interest (ROI) can have a significant impact on the maximum allowed frame rate. If possible in your application, decrease the height of the ROI.
- Depending on the sensor, decreasing the width of the ROI may also increase the maximum allowed frame rate. The impact is lower than the impact of the ROI height, but may still be noticeable.
- If you are using long exposure times or small ROIs, your exposure time may limit the maximum allowed frame rate. Try using a shorter exposure time and see if the maximum allowed frame rate increases. You may need to compensate for a lower exposure time by using a brighter light source or increasing the opening of your lens aperture.
- If your camera is equipped with a rolling shutter, use the electronic rolling shutter (ERS) mode rather than the global reset release shutter mode. The ERS mode allows overlapping frame acquisition while the global reset release mode does not. Overlapping frame acquisitions is, however, necessary for achieving the highest frame rates.

If you are working with exposure time, keep in mind that a very long exposure time can severely limit the camera's maximum allowed frame rate.

Example: Assume that your camera is set to use a 1/2 second exposure time. In this case, because each frame acquisition will take at least 1/2 second to be completed, the camera will only be able to acquire a maximum of two frames per second.

For more information about

- ROI settings, see Section 11.7 on page 97.
- the ERS mode, see Section 9.4.2 on page 70.
- overlapping image acquisitions, see Section 9.5 on page 73.

10 Pixel Formats

10.1 Available Pixel Formats

For all dart BCON **mono** cameras, the following pixel formats are available:

- Mono 8
- Mono 12

For all dart BCON **color** cameras, the following pixel formats are available:

- YCbCr422
- Bayer 8
- Bayer 12
- RGB 8

The standard alignment of the Bayer filter to the pixels in the images acquired by the dart BCON color cameras is **GB**.

If you are using dart model daA1600-60bc or daA2500-14bc, the alignment will change if you enable image mirroring.

For more information about

- the color filter alignment, see Section 11.6.1 on page 89.
- image mirroring, see Section 11.9 on page 105.

The image sensor of the daA1600-60uc/um delivers 10 bits of data per pixel. If you set the daA1600-60uc/um for a 12-bit pixel format (Mono 12 or Bayer 12), the camera will output 12-bit image data based on 10-bit sensor data.

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

You can set the PixelFormat parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the pixel format to Bayer GB 12:

```
// Set the pixel format to Bayer GB 12
camera.PixelFormat.SetValue(PixelFormat_BayerGB12);
```

10.2 Details on Pixel Formats for Color Cameras

YCbCr422 Format

All dart BCON color cameras can output color images based on pixel data in YCbCr422 format. This is the default pixel format.

When a color camera is set for this format, each pixel value in the captured image goes through a two step conversion process as it exits the sensor and passes through the camera's electronics. This process yields Y, Cb, and Cr color information for each pixel.

In the first step of the process, a demosaicing algorithm is performed to get RGB data for each pixel. This is required because color cameras with a Bayer filter on the sensor gather only one color of light for each individual pixel.

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

Y = 0.299 R + 0.587 G + 0.114 B

Cb = - 0.16874 R - 0.33126 G + 0.5000 B + 128

Cr = 0.5000 R - 0.41869 G - 0.08131 B + 128

After conversion to the YCbCr color model is complete, the pixel data is transmitted out of the camera.

By default, when the pixel format is set to YCbCr422 or RGB 8, images are acquired in sRGB color space.

You can use the BslColorSpaceMode parameter to change the color space to RGB. For more information, see Section 11.4 on page 85.

Bayer Formats

All dart BCON color cameras can output color images based on the pixel formats Bayer 8 and Bayer 12.

When a color camera is set for one of these Bayer pixel formats, it outputs 8 or 12 bits of data per pixel. For each pixel covered with a red, green, or blue filter, you get 8 or 12 bits of red, green, or blue data. This type of pixel data is sometimes referred to as "raw" output.

For more information about

- the Bayer filter, see Section 11.6.1 on page 89.
- the Balance White feature, see Section 11.6.3 on page 92.

11 Features

11.1 Feature Sequence

Most of the camera features described in this chapter modify the pixel data output by the image sensor. These features are processed in a specific sequence. Knowing the sequence is especially useful if you are configuring multiple features and want to avoid side effects.

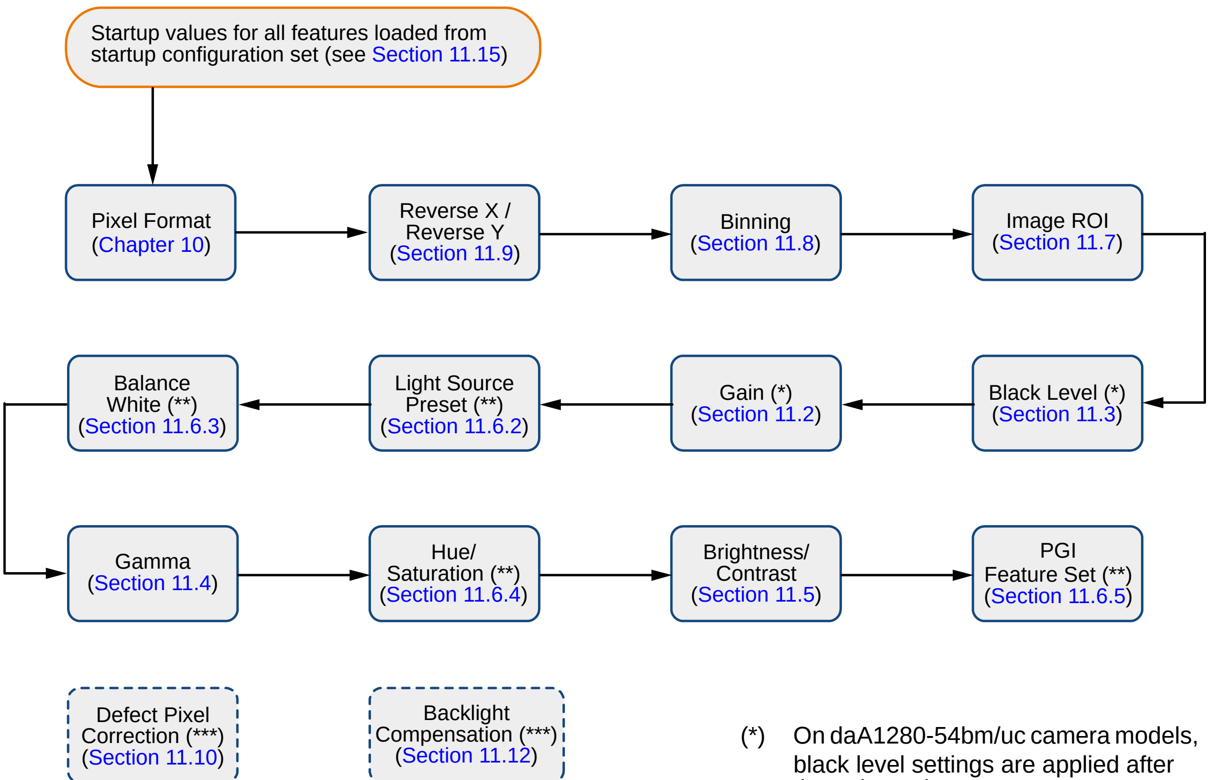


Fig. 36: Feature Sequence

(*) On daA1280-54bm/uc camera models, black level settings are applied after the gain settings.
(**) Color camera models only.
(***) Can be used in any sequence.

11.2 Gain

By default, the gain auto function is enabled. Manual adjustment of the Gain parameter will not work.

Set the GainAuto parameter to Off before making any manual adjustments.

For more information about the gain auto function, see Section 11.11.3 on page 116.

The camera's Gain feature is an analog feature allowing you to adjust gain. As shown in Figure 37, increasing the gain increases the slope of the response curve for the camera. This results in a higher gray value output from the camera for a given amount of output from the imaging sensor. Decreasing the gain decreases the slope of the response curve and results in a lower gray value for a given amount of sensor output.

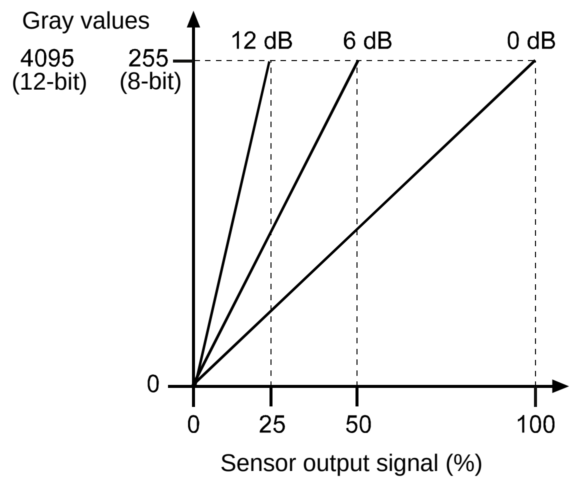


Fig. 37: Gain in dB

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

This section describes how gain can be adjusted manually by setting the value of the Gain parameter. The camera also has a gain auto function that can automatically adjust the gain.

The camera's gain is determined by the value of the Gain parameter. The parameter is adjusted in dB. The maximum regular value varies by camera model (see Table 32).

Camera Model	Max Allowed Setting (in dB)
daA1280-54bm/bc	18
daA1600-60bm/bc daA2500-14bm/bc	24

Table 32: Maximum Allowed Gain Settings (in dB)

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

```
// Set the gain to 0.0359
camera.Gain.SetValue(0.0359);
```

11.3 Black Level

i	On daA1280-54bm/bc cameras, the black level settings are applied to the pixel data **after** the gain settings have been applied. On all other dart BCON cameras, the black level settings are applied **before** the gain settings.

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 (+1) in the pixel values output for the pixels. Decreasing the black level setting will result in a negative offset (-1) in the pixel values output for the pixels.

The black level can be adjusted by changing the value of the BlackLevel parameter.

The range of the allowed settings for the BlackLevel parameter value in DN varies by pixel format as shown in Table 33.

Min Allowed Black Level Setting	Max Allowed Black Level Setting (8-bit pixel format)	Max Allowed Black Level Setting (12-bit pixel format)
0	32	512

Table 33: Minimum and Maximum Black Level Settings [DN]

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

```
// Set the black level to 1.0
camera.BlackLevel.SetValue(1.0);
```

11.4 Gamma

The Gamma feature lets you modify the brightness of acquired images to account for a non-linearity in the human perception of brightness.

The gamma correction can be set manually as desired using the Gamma parameter:

- Gamma = 1: The overall brightness is **not corrected** (unless the pixel format is set to YCbCr422 or RGB8, see section below).
- Gamma < 1: The overall brightness is **increased**.
- Gamma > 1: The overall brightness is **decreased**.

In all cases, black pixels (brightness = 0) and white pixels (brightness = maximum) will not be adjusted. The maximum pixel brightness equals 255 for 8-bit output and 4095 for 12-bit output.

To accomplish gamma correction, a gamma correction value (γ) is applied to the pixel value of each pixel according to the following formula:

$$Y_{corrected} = \left(\frac{Y_{uncorrected}}{Y_{max}}\right)^{\gamma} \times Y_{max}$$

sRGB Gamma Correction and Color Space Mode

If the pixel format is set to YCbCr422 or RGB 8, you can use the BslColorSpaceMode parameter to change the color space for image acquisition. This will enable or disable the application of an additional sRGB gamma correction value.

You can set the BslColorSpaceMode parameter to the following values:

- **sRGB:** The image brightness is optimized for display on an sRGB monitor. An additional gamma correction value of approximately 0.4 is applied. This is the default setting.

  Note that the sRGB gamma correction is independent from the Gamma parameter and will not be reflected in the Gamma parameter value.

  Example: You set the BslColorSpaceMode parameter to sRGB and the Gamma parameter value to 1.2. First, an sRGB gamma correction value of approximately 0.4 is applied to the pixel values. Then, a gamma correction value of 1.2 is applied to the resulting pixel values.
- **RGB:** No additional sRGB gamma correction value is applied.

If the pixel format is set to Bayer 8 or Bayer 12, the BslColorSpaceMode parameter is not available, and gamma correction is always performed in RGB color space.

You can set the parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the color space to sRGB and the gamma correction 1.2 as an example:

```
// Set the color space to sRGB
camera.BslColorSpaceMode.SetValue(BslColorSpaceMode_sRGB);
// Set the Gamma value to 1.2
camera.Gamma.SetValue(1.2);
```

11.5 Brightness/Contrast

The Brightness/Contrast feature lets you make adjustments to the tonal values of your images.

You can adjust brightness and contrast by setting the BslBrightness, BslContrast, and BslContrastMode parameters.

The BslBrightness and BslContrast parameter values are always processed together as parts of a common Brightness/Contrast function, as described in the following sections.

11.5.1 Brightness

Adjusting the brightness allows you to lighten or darken the image by increasing or decreasing its tonal values.

You can adjust the brightness by setting the BslBrightness parameter. The parameter value can range from -1 to 1. By default, the parameter is set to 0, and no brightness adjustment is performed.

Adjusting the brightness moves the pivot point of the Brightness/Contrast function. Figure 38 shows an example for the S-curve contrast mode set. For more information about contrast and contrast modes, see Section 11.5.2 on page 87.

If you increase the brightness, the pivot point moves towards the upper left. As a result, the image will appear lighter. Decreasing the brightness moves the pivot point to the lower right, and the image will appear darker.

8-bit pixel values shown as an example

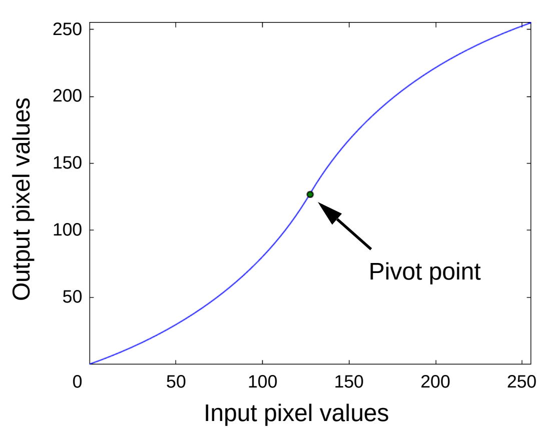


Fig. 38: Contrast = 0.3, Brightness = 0, S-Curve Contrast Mode

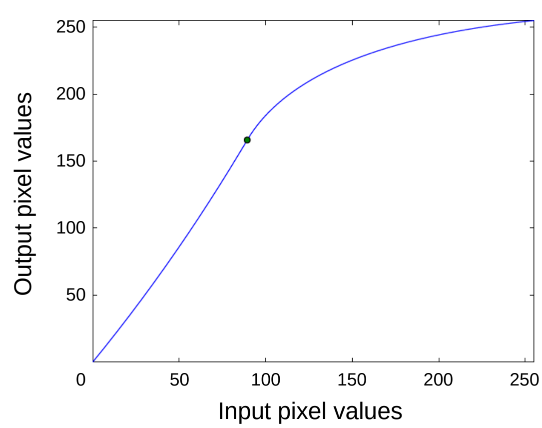


Fig. 39: Contrast = 0.3, Brightness = 0.3, S-Curve Contrast Mode

11.5.2 Contrast

Adjusting the contrast changes the degree of difference between light and dark areas in the image. The more contrast you apply, the more defined the difference will be.

You can adjust the contrast by setting the BslContrast parameter. The parameter value can range from -1 to 1. By default, the parameter is set to 0, and no contrast adjustment is performed.

How the BslContrast parameter works depends on the **contrast mode** set. You can select the contrast mode by setting the BslContrastMode parameter to one of the following values:

- **Linear:** This is the default contrast mode. When this contrast mode is set, a linear function is used to adjust the contrast. Increasing or decreasing the BslContrast parameter increases or decreases the slope of the linear function.

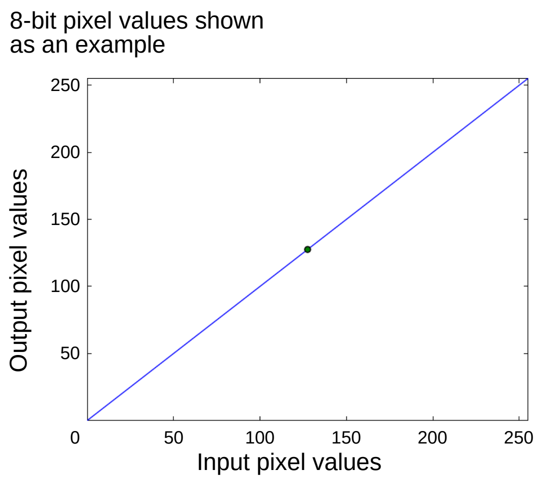


Fig. 40: Contrast = 0, Brightness = 0, Linear Contrast Mode

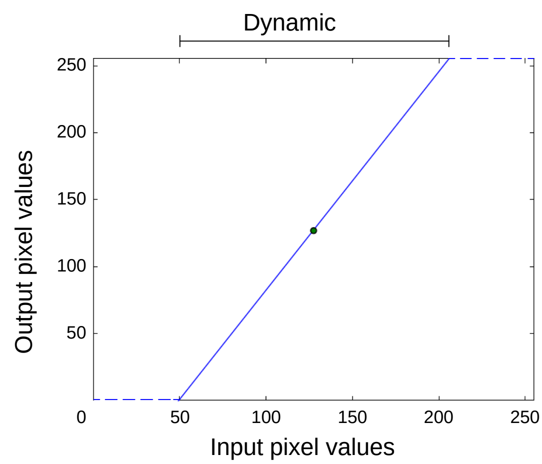


Fig. 41: Contrast = 0.3, Brightness = 0, Linear Contrast Mode

  The higher the slope of the linear function, the more output pixel values will be set to completely black (0) or completely white (maximum pixel value). In the example shown in Figure 41, input values from 0 to approximately 50 are set to completely black, and input values from approximately 205 to 255 are set to completely white.

  This means that the overall range of tonal values, i.e, the dynamic range of the image, is decreased. The darkest and lightest regions of the image will appear completely black or completely white, but the other areas will appear more defined.

  Decreasing the contrast has the opposite effect.

- **S-Curve:** When this contrast mode is set, an S-curve function is applied to adjust the contrast. This allows you to improve perceived contrast while preserving the dynamic range of the image.

  When the contrast mode is set to S-curve and both BslContrast and BslBrightness are set to 0, the Brightness/Contrast function looks as shown in Figure 40. If you increase the contrast, however, the graph of the function is shaped like an "S":

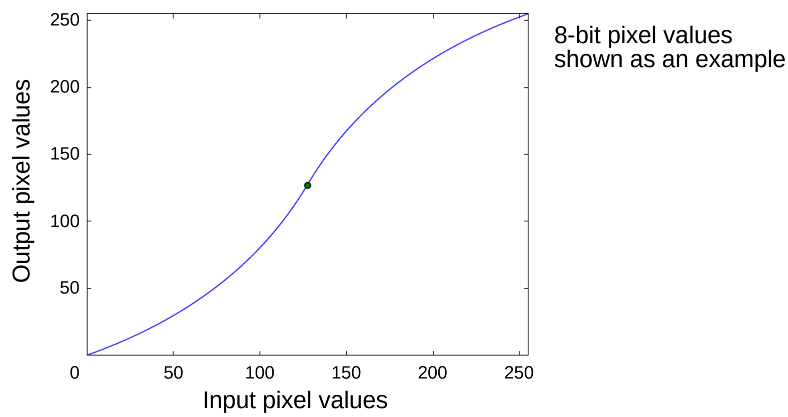


Fig. 42: Contrast = 0.3, Brightness = 0, S-Curve Contrast Mode

As shown in Figure 42, increasing the contrast in S-curve mode has the following effects:

- Because the S-curve is flatter around its starting and end points and steeper around the center, contrast in light and dark areas of the image is reduced, and contrast in mid tones is increased.
- Low input pixel values are lowered and high input pixel values are increased. As a result, extreme dark and light areas of your image are compressed, which further improves the perceived contrast.
- As the curve always starts at (0,0) and ends at (X_{max},Y_{max}), the dynamic range of the image is always preserved.

Setting contrast below 0 in S-curve mode results in an inverted S-curve with opposite effects.

11.5.3 Setting Brightness and Contrast

You can set brightness and contrast within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameters:

```
// Set the Brightness parameter value to 0.5
camera.BslBrightness.SetValue(0.5);
// Set the contrast mode to Linear
camera.BslContrastMode.SetValue(BslContrastMode_Linear);
// Set the Contrast parameter value to 1.2
camera.BslContrast.SetValue(1.2);
```

11.6 Color Enhancement Features

11.6.1 General Considerations

The sensors in the color versions of the Basler dart BCON cameras are equipped with an additive color separation filter known as a Bayer filter. The pixel formats available on color cameras for pixel data output are related to the Bayer pattern.

With the Bayer filter, each individual pixel is covered by a part of the filter that allows light of only one color to strike the pixel. The pattern of the Bayer filter used on the camera is as shown in Figure 43.

As the figure illustrates, within each square of four pixels, one pixel sees only red light, one sees only blue light, and two pixels see only green light. This combination mimics the human eye's sensitivity to color.

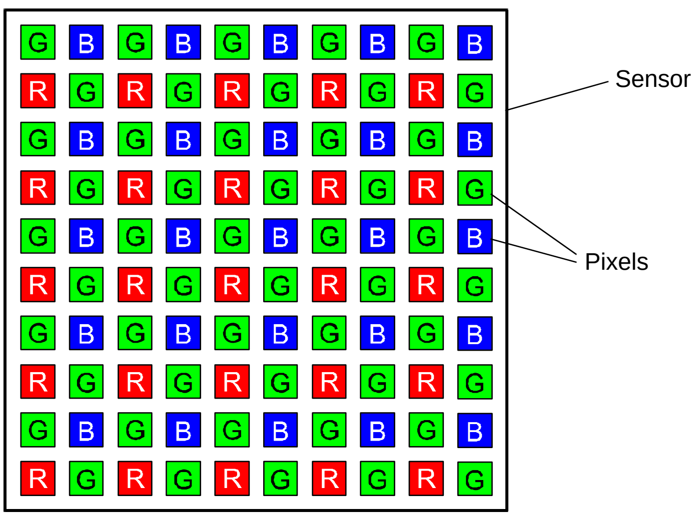


Fig. 43: Bayer Filter Pattern with "GB" Alignment

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

i	The standard alignment of the Bayer filter to the pixels in the images acquired by the dart BCON color cameras is **GB**. If you are using dart model daA1280-54bc, this alignment is fixed. If you are using dart model daA1600-60bc or daA2500-14bc, the alignment will change if you enable image mirroring: If you use ■ the Reverse X feature to mirror the image horizontally, the effective Bayer color filter alignment will be BG. ■ the Reverse Y feature to mirror the image vertically, the effective Bayer color filter alignment will be RG. ■ the Reverse X and the Reverse Y feature, the effective Bayer color filter alignment will be GR. For more information about image mirroring, see Section 11.9.1 on page 105 and Section 11.9.2 on page 106.

11.6.2 Light Source Preset

i	If the balance white auto function is set to Off, you must set the white balance parameters to neutral values before changing the light source preset. See the instructions below.

By using a light source preset, you can correct color shifts caused by a specific light source. This also corrects color inaccuracies caused by the image sensor.

You can set the following light source presets:

- **Off** - No alterations will be made to the pixel values. This also means that the camera does not correct color inaccuracies caused by the image sensor. Therefore, Basler does not recommend using this setting.
- **Tungsten 2800 K** - The camera optimizes the white balance settings for a tungsten light source that has a color temperature of about 2800 K. If the pixel format is set to YCbCr422 or RGB8, the camera also corrects color inaccuracies caused by the image sensor.
- **Daylight 5000 K / 6500 K** - The camera optimizes the white balance settings for daylight lighting that has a color temperature of about 5000 K or 6500 K. If the pixel format is set to YCbCr422 or RGB8, the camera also corrects color inaccuracies caused by the image sensor.

i	After camera power up or reset, the light source preset is set to "Daylight 5000 K" unless you define a startup set with a different light source preset. For more information about the startup set, see Section 11.15 on page 126.

To set a light source preset:

1. If the BalanceWhiteAuto parameter is set to Off, reset the white balance by setting the BalanceRatio parameter value for all color channels (red, green, and blue) to 1. For more information about the BalanceRatio parameter, see Section 11.6.3 on page 92.
2. Set the LightSourcePreset parameter to the desired light source preset.

You can set the parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to configure white balancing and set a light source preset:

```
// When the balance white auto function is disabled,
// white balance must be reset before changing the preset
camera.BalanceWhiteAuto.SetValue(BalanceWhiteAuto_Off);
camera.BalanceRatioSelector.SetValue(BalanceRatioSelector_Red);
camera.BalanceRatio.SetValue(1);
camera.BalanceRatioSelector.SetValue(BalanceRatioSelector_Green);
camera.BalanceRatio.SetValue(1);
camera.BalanceRatioSelector.SetValue(BalanceRatioSelector_Blue);
camera.BalanceRatio.SetValue(1);
// Set the light source preset to "Tungsten 2800 K"
camera.LightSourcePreset.SetValue(LightSourcePreset_Tungsten2800K);

// When the balance white auto function is enabled,
// the preset can be changed right away
camera.BalanceWhiteAuto.SetValue(BalanceWhiteAuto_Continuous);
camera.LightSourcePreset.SetValue(LightSourcePreset_Daylight5000K);
```

11.6.3 Balance White

By default, the balance white auto function is enabled. Manual adjustment of the white balance will not work.

Set the BalanceWhiteAuto parameter to Off before making any manual adjustments.

For more information about the balance white auto function, see Section 11.11.5 on page 119.

The Balance White feature allows you to manually correct color shifts. A digital gain correction can be applied per color (red, green, blue) so that white objects in the camera's field of view appear white in the acquired images.

While the Light Source Preset feature is useful to preconfigure the white balance based on fixed preset values (see Section 11.6.2 on page 90), the Balance White feature lets you fine-tune the white balance for your specific imaging conditions.

The white balance is effective on all pixel data output formats including "raw" pixel formats (Bayer 8 and Bayer 12).

Setting the White Balance

You can perform white balancing by adjusting the intensity of the colors red, green, and blue in your images. Each color can be individually adjusted using the BalanceRatio parameter.

To set the white balance:

1. Set the BalanceRatioSelector parameter to Red, Green, or Blue.
2. Set the BalanceRatio parameter to the desired value for the selected color channel.

The BalanceRatio parameter value can range from 1 to 7.984375:

- BalanceRatio = 1: The intensity of the color is **unaffected** by the white balance mechanism.
- Balance ratio > 1: The intensity of the color is **increased** relative to the other two colors.
- Balance ratio < 1: The intensity of the color is **decreased** relative to the other two colors.

The increase or decrease in intensity is proportional. For example, if the balance ratio for a color is set to 1.25, the intensity of that color is increased by 25 %.

You can set the BalanceRatioSelector and the BalanceRatio parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value for green as an example:

```
// Select the green color channel and set its intensity to 125%
camera.BalanceRatioSelector.SetValue(BalanceRatioSelector_Green);
camera.BalanceRatio.SetValue(1.25);
```

11.6.4 Hue/Saturation

ℹ	The Hue/Saturation feature is only available if the pixel format is set to YCbCr422 or RGB 8. For more information about pixel formats, see Chapter 10 on page 80.

The Hue/Saturation feature lets you adjust the color appearance in your images.

Hue

Adjusting the **hue** shifts the colors of the image. This can be useful, e.g., for correcting minor color shifts or creating false-color images.

You can adjust the hue by setting the BslHue parameter. The parameter can be set in a range from -180° to 180°. The parameter value refers to a rotation of the RGB color cube. By default, the parameter is set to 0° (no color shift).

Saturation

Adjusting the **saturation** changes the colorfulness (intensity) of the colors. A higher saturation, for example, will make colors easier to distinguish.

You can adjust the saturation by setting the BslSaturation parameter. By default, the parameter is set to 1 (normal saturation). Lower parameter values result in more muted colors that are closer to gray. Higher parameter values result in more vivid, vibrant colors.

Setting Hue and Saturation

You can set hue and saturation within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameters:

```
// Set the Hue parameter value to 5 degrees
camera.BslHue.SetValue(5);
// Set the Saturation parameter value to 1.4
camera.BslSaturation.SetValue(1.4);
```

11.6.5 PGI Feature Set

The Basler PGI feature set is integrated into all Basler dart BCON color cameras. It consists of four image optimization features:

- Denoising
- 5 × 5 Debayering
- Color Anti-Aliasing
- Sharpness Enhancement

With the exception of the Sharpness Enhancement feature, these optimizations are done **automatically** and can’t be controlled by the user.

The PGI image optimizations take effect whenever the pixel format is set to YCbCr422 or RGB 8. To disable PGI image optimizations, set the pixel format to Bayer 8 or Bayer 12.

For more information about

- pixel formats, see Chapter 10 on page 80.
- the Sharpness Enhancement feature, see Section 11.6.6 on page 95.
- the PGI image optimizations, see the white paper "Better Image Quality with Basler PGI". You can find the white paper in the "Documents" section of the Basler website: www.baslerweb.com.

11.6.6 Sharpness Enhancement

i	The Sharpness Enhancement feature is only available if the pixel format is set to YCbCr422 or RGB 8. For more information about pixel formats, see Chapter 10 on page 80.

The Sharpness Enhancement feature lets you increase the sharpness of the captured images. The higher the sharpness, the more distinct the image subject's contours will be.

You can adjust the amount of sharpness by setting the SharpnessEnhancement parameter value. The parameter value can range from 0 (no sharpening applied) to 1 (maximum sharpening applied).

i	If you use the Gain and the Sharpness Enhancement feature at the same time, increasing the Gain parameter will reduce the amount of sharpening applied. The value of the SharpnessEnhancement parameter will remain the same, but the image will appear less sharpened. This adjustment is done automatically to keep image noise as low as possible.

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

```
// Set the SharpnessEnhancement parameter value to 0.39
camera.SharpnessEnhancement.SetValue(0.39);
```

11.6.7 A Procedure for Setting the Color Enhancements

When setting the color enhancements on the camera, we recommend using the procedure outlined below.

i	The procedure aims at producing a color reproduction on a monitor that is optimized for human vision. The optimum for machine vision may require different color enhancement settings.

To set the color enhancements:

1. Arrange your camera so that it is viewing a scene similar to what it will view during actual operation. Make sure that the lighting for the scene is as close as possible to the actual lighting you will be using during normal operation. Using lighting that represents your normal operating conditions is extremely important.

   We recommend including a standard color chart within your camera's field of view when you are adjusting the color enhancements. One widely used chart is the ColorChecker® chart (also known as the Macbeth chart).
2. Set the exposure auto, gain auto, and balance white auto functions to Off.
3. Reset the white balance by setting all BalanceRatio parameter values to 1.
4. Make sure the settings for gain and black level are at their minimums.
5. Set the LightSourcePreset parameter to the value that is most appropriate for your lighting. For example, if you use tungsten incandescent light, select the Tungsten2800K light source preset.
6. Begin capturing images and check the basic image appearance.
7. Set the exposure time, black level, and gain so that you are acquiring good quality images. It is important to make sure that the images are not overexposed. Overexposure can have a significant negative effect on the fidelity of the color in the acquired images. Generally, the settings for black level and gain should be as low as possible.
8. Adjust the white balance. Make sure a white or light gray object is imaged while white balance is carried out. An easy way to set the white balance is to use the Once mode of operation on the camera's Balance White Auto feature.
9. If necessary, set the gamma value. When gamma is set correctly, there should be a smooth transition from the lightest to the darkest gray scale targets on your color chart or on a gray scale.
10. If necessary, fine-tune the colors and tonal values by adjusting brightness, contrast, hue, and saturation.
11. Examine the colors and see if they are satisfactory at this point. If not, restart from step 3.

    .

i	Certain conditions outside the camera, such as the lighting, the camera lens, filter, or the monitor settings are relevant to the reproduction of color in the image. If you change any of these conditions, you will have to repeat the above procedure to preserve optimum color reproduction.

11.7 Image ROI

The Image Region of Interest (ROI) feature lets you specify a portion of the sensor array. After each image is acquired, only the pixel information from the specified portion of the array is transmitted out of the camera.

The region 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 44.

The location and size of the region of interest is defined by declaring an offset X (coordinate), a width, an offset Y (coordinate), and a height. For example, suppose that you specify the offset X as 10, the width as 16, the offset Y as 6, and the height as 10. The region of the array that is bounded by these settings is shown in Figure 44.

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

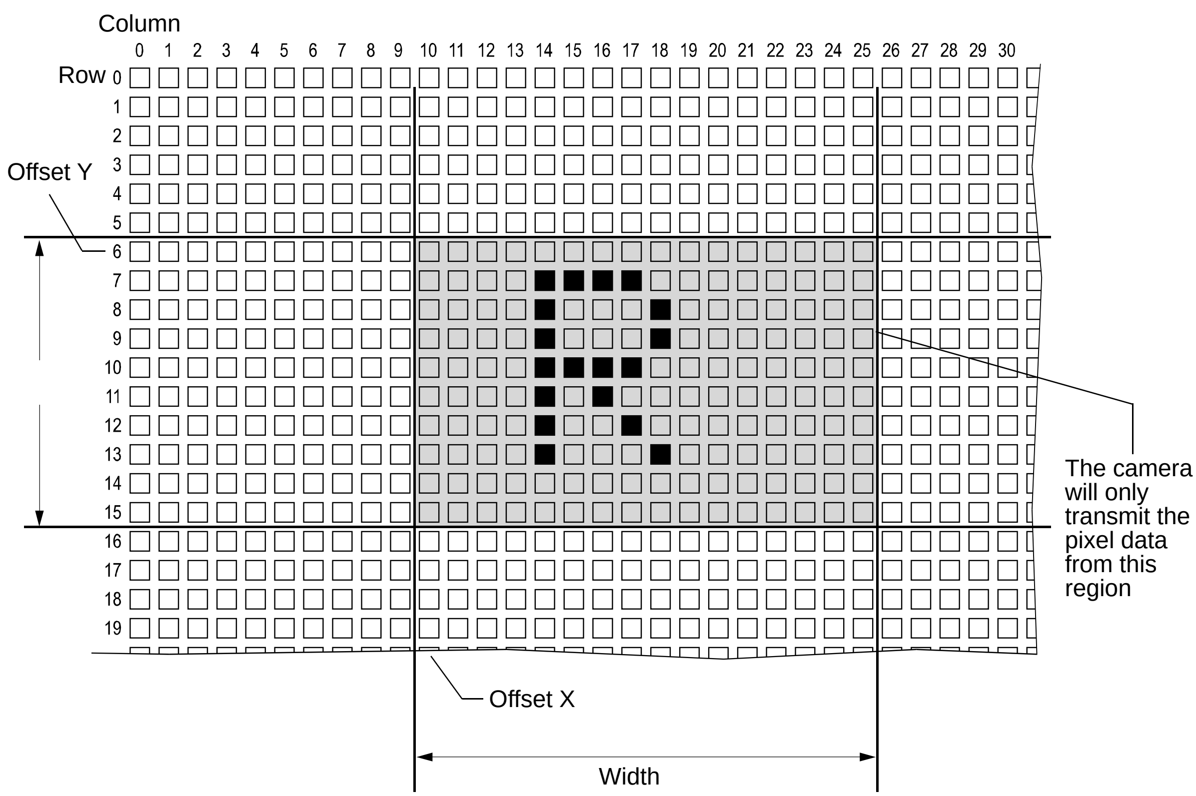


Fig. 44: Region of Interest

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

For more information about how changing the ROI height affects the maximum allowed frame rate, see Section 9.7 on page 78.

Guidelines for Setting the Image ROI

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

- OffsetX: determines the starting column for the region of interest.
- OffsetY: determines the starting row for the region of interest.
- Width: determines the width of the region of interest.
- Height: determines the height of the region of interest.

When you are setting the camera's ROI, you must follow these guidelines:

Guideline	Example
Offset X + ROI width < Width of camera sensor	daA1600-60um: Offset X: 463, ROI width: 500 Width of camera sensor: 1600 463 + 500 < 1600
Offset Y + ROI height < Height of camera sensor	daA1600-60um: Offset Y: 351, ROI height: 200 Height of camera sensor: 1200 351 + 200 < 1200

Table 34: Guidelines for Setting the Camera's ROI

ROI Parameters	Camera Model	Parameter Range	Example
OffsetX OffsetY	daA1280-54bm/bc daA1600-60bc daA2500-14bm/bc	▪ Can be set in increments of 2 ▪ Must be set to an even number	0, 2, 4, 6, 8, etc.
OffsetX	daA1600-60bm	▪ Can be set in increments of 8. ▪ Must be set to an even number .	8, 16, 24, etc.
Width	daA1280-54bm/bc daA1600-60bc	▪ Can be set in increments of 2 ▪ Minimum value is 16.	16, 18, 20, 22, 24, etc.
	daA1600-60bm	▪ Can be set in increments of 8 ▪ Minimum value is 16.	16, 24, 32, 40, etc.
	daA2500-14bm/bc	▪ Can be set in increments of 2 ▪ Minimum value is 4.	4, 6, 8, 10, 12, etc.
Height	daA1280-54bm/bc daA1600-60bm/bc	▪ Can be set in increments of 2 ▪ Minimum value is 8.	8, 10, 12, 14, 16, etc.
	daA2500-14bm/bc	▪ Can be set in increments of 2 ▪ Minimum value is 4.	4, 6, 8, 10, 12, etc.

Table 35: ROI Parameters and Parameter Ranges

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

```
int64_t i = camera.WidthMax.GetValue();
int64_t i = camera.HeightMax.GetValue();

camera.Width.SetValue(1294);
camera.Height.SetValue(964);

camera.OffsetY.SetValue(0);
camera.OffsetX.SetValue(0);
```

Changing ROI Parameters "On-the-Fly"

Making ROI 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 following parameters:

- OffsetX
- OffsetY

Changes to the ROI size are not allowed on-the-fly.

11.8 Binning

With binning, multiple sensor pixels are combined and reported out of the camera as a single pixel.

Binning Directions

You can set binning in two directions: horizontal or vertical.

- With **horizontal binning**, adjacent pixels from a specific number of **columns** are combined and are reported out of the camera as a single pixel.
- With **vertical binning**, adjacent pixels from a specific number of **rows** in the imaging sensor array are combined and are reported out of the camera as a single pixel.

You can use both horizontal and vertical binning at the same time. However, if you use a different binning factor for horizontal and vertical binning, objects will appear distorted in the image. For more information about possible image distortion, see Section 11.8.3 on page 104.

The number of binned pixels depends on the horizontal binning and the vertical binning settings. For more information about the binning settings, see Section 11.8.1 on page 101.

Binning Modes

Two modes can be used to perform binning:

- **Sum**: The values of the affected pixels are summed. This increases the camera's response to light and the signal-to-noise ratio.
- **Average**: The values of the affected pixels are averaged. This increases the signal-to-noise ratio without changing the camera's response to light.

Both modes reduce the amount of image data to be transferred. This may result in higher camera frame rates.

On some camera models, the binning modes used by the camera are preset and can't be changed.

On other camera models, the binning modes can be set. For more information, see Section 11.8.2 on page 103.

Binning on Color and Monochrome Cameras

On monochrome cameras, the values of directly adjacent pixels are summed or averaged:

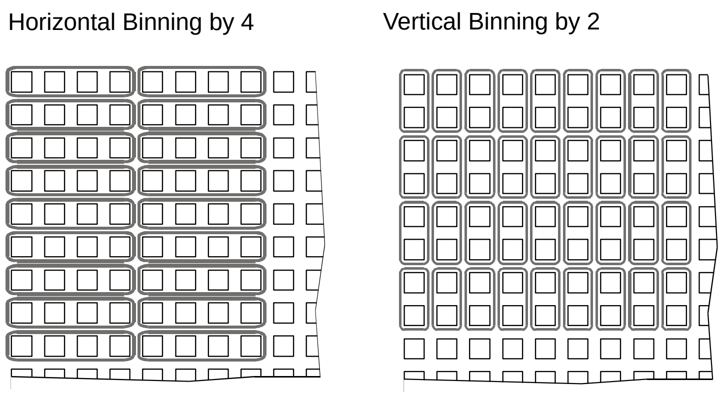


Fig. 45: Binning on Monochrome Cameras

On color cameras, the values of adjacent pixels of the same color are summed or averaged:

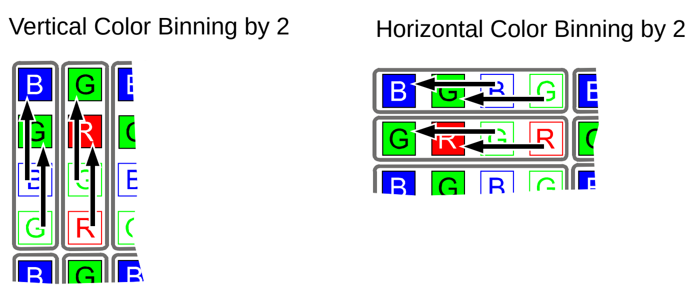


Fig. 46: Binning on Color Cameras

11.8.1 Setting Binning

You can enable

- **horizontal binning** by setting the **BinningHorizontal** parameter.
- **vertical binning** by setting the **BinningVertical** parameter.

Setting the parameter’s value to 2, 3, or 4 enables horizontal or vertical binning by 2, by 3, or by 4, respectively. Setting the parameter’s value to 1 disables horizontal or vertical binning.

Always set the BinningHorizontal parameter first, then set the BinningVertical parameter.

If the BinningHorizontal parameter is not set first, some combinations of vertical binning and horizontal binning can't be achieved.

For a list of allowed combinations, see Table 36.

Example: On daA1600-60bm/bc cameras, you can't set the BinningVertical parameter to 2 when the BinningHorizontal parameter is set to 1. Set the BinningHorizontal parameter to 2 first, then set the BinningVertical parameter to 2.

The range of allowed settings for the BinningHorizontal and the BinningVertical parameter values varies by camera model as shown in Table 36.

Camera Model	Allowed Settings BinningHorizontal	Allowed Settings BinningVertical	Allowed Combinations (H x V Binning)	Notes
daA1280-54bm/bc	1, 2	1, 2	1 x 1 2 x 1 2 x 2	
daA1600-60bm/bc	1, 2	1, 2	1 x 1 2 x 2	
daA2500-14bm/bc	1, 2, 3 (*), 4	1, 2, 3, 4	All	* Horizontal binning by 3 is not supported. Setting the parameter value to 3 is allowed, but will result in an effective horizontal binning by 2.

Table 36: Binning Horizontal and Binning Vertical Settings

You can set the Binning Horizontal and the BinningVertical parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values:

```
// Enable horizontal binning by 4
camera.BinningHorizontal.SetValue(4);
// Enable vertical binning by 2
camera.BinningVertical.SetValue(2);
// Disable horizontal and vertical binning
camera.BinningVertical.SetValue(1);
camera.BinningHorizontal.SetValue(1);
```

11.8.2 Setting the Binning Mode

If supported, you can set the

- **horizontal binning mode** by setting the **BinningHorizontalMode** parameter.
- **vertical binning mode** by setting the **BinningVerticalMode** parameter.

Always set the BinningHorizontalMode parameter first, then set the BinningVerticalMode parameter.

If the BinningHorizontalMode parameter is not set first, some combinations of the binning horizontal mode and the binning vertical mode can't be achieved.

For a list of allowed combinations, see Table 37.

Example: On daA1600-60bm/bc cameras, you can't set the BinningVerticalMode parameter to Sum when the BinningHorizontalMode parameter is set to Average. Set the BinningHorizontalMode parameter to Sum first, then set the BinningVerticalMode parameter to Sum.

The range of allowed settings for the BinningHorizontalMode and the BinningVerticalMode parameter values varies by camera model as shown in Table 37.

Camera Model	Allowed Settings BinningHorizontalMode Parameter	Allowed Settings BinningVerticalMode Parameter	Allowed Combinations (H x V Binning Mode)
daA1280-54bm/bc	Average	Average	Average x Average
daA1600-60bm/bc	Average, Sum	Average, Sum	Average x Average Sum x Sum
daA2500-14bm/bc	Average, Sum	Average	Average x Average Sum x Average

Table 37: Binning Mode Horizontal and Binning Mode Vertical Settings

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

```
// Set the horizontal binning mode to Average
camera.BinningHorizontalMode.SetValue(BinningHorizontalMode_Average);
// Set the vertical binning mode to Sum
camera.BinningVerticalMode.SetValue(BinningVerticalMode_Sum);
// Determine the vertical binning mode
e = camera.BinningVerticalMode.GetValue();
```

11.8.3 Considerations When Using Binning

Binning's Effect on ROI Settings

When you have the camera set to use binning, keep in mind that the settings for your image region of interest (ROI) will refer to the binned rows and columns in the sensor and not to the physical rows and columns in the sensor as they normally would.

For example, assume that you are using a daA1600-60um camera set for 2 by 2 binning as described above. In this case, you would act as if you were actually working with an 800 column by 600 row sensor when setting your ROI parameters.

For more information about the Image Region of Interest (ROI) feature, see Section 11.7 on page 97.

Increased Response to Light

Using binning with summed pixel values (see "Binning Modes" in Section 11.8 on page 100) can greatly increase the camera's response to light. When pixel values are summed, the acquired images may look overexposed. If this is the case, you can reduce the lens aperture, the intensity of your illumination, the camera's exposure time setting, or the camera's gain setting.

Reduced Resolution

Using binning effectively reduces the resolution of the camera's imaging sensor. For example, the sensor in the daA1600-60um camera normally has a resolution of 1600 (H) x 1200 (V). If you set this camera to use horizontal binning by 2 and vertical binning by 2, the effective resolution of the sensor is reduced to 800 (H) by 600 (V).

Possible Image Distortion

Objects will only appear undistorted in the image if the numbers of binned lines and columns are equal. With all other combinations, the imaged objects will appear distorted. If, for example, vertical binning by 2 is combined with horizontal binning by 4, the widths of the imaged objects will appear shrunk by a factor of 2 compared to the heights.

11.9 Reverse X and Reverse Y

The camera's Reverse X and Reverse Y features let you flip the captured images horizontally and/or vertically before they are transmitted from the camera.

Reverse X and Reverse Y can be enabled at the same time. This effectively rotates the image by 180 degrees.

11.9.1 Reverse X

The Reverse X feature is a horizontal mirror image feature. When Reverse X 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.

> If you are using dart model daA1600-60uc or daA2500-14uc, the effective Bayer color pixel alignment will change from GB to BG if you enable reverse X.
>
> For more information about color pixel alignments, see Section 11.6 on page 89.

Figure 47 shows a normal image on the left and an image captured with reverse X enabled on the right.

Normal Image **Mirror Image**

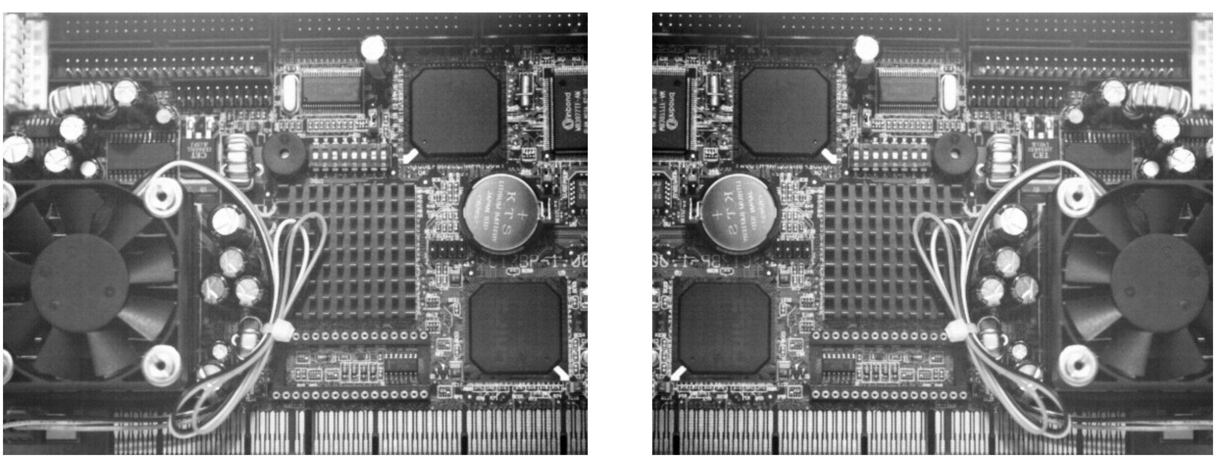


Fig. 47: Reverse X Mirror Imaging

Using ROIs with Reverse X

You can use the Image ROI and Auto Function ROI features when using the Reverse X feature. The position of an ROI relative to the sensor remains the same regardless of whether or not the Reverse X feature is enabled.

As a consequence, different regions of the image will be controlled or displayed depending on whether or not the Reverse X feature is enabled. See example in Figure 48.

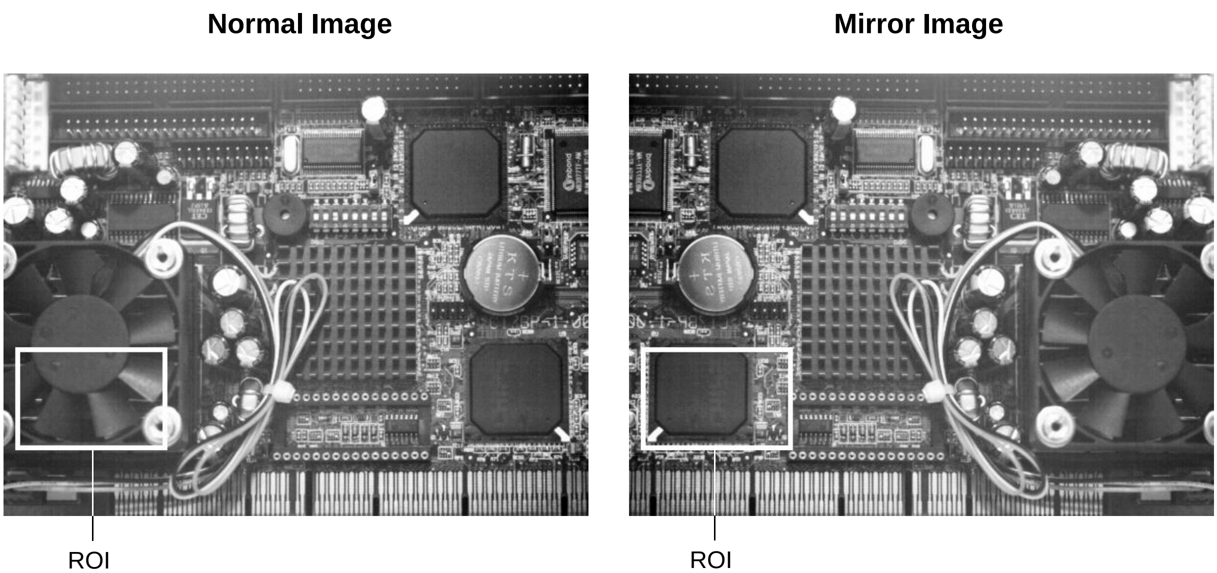


Fig. 48: Using an ROI with Reverse X Mirror Imaging

11.9.2 Reverse Y

The Reverse Y feature is a vertical mirror image feature. When the Reverse Y feature is enabled, the lines in a captured image will be swapped top-to-bottom. This means that the top line in the image will be swapped with the bottom line, the next-to-top line will be swapped with the next-to-bottom line, and so on.

> If you are using dart model daA1600-60uc or daA2500-14uc, the effective Bayer color pixel alignment will change from GB to RG if you enable reverse Y.
>
> For more information about color pixel alignments, see Section 11.6 on page 89.

Figure 49 shows a normal image on the left and an image captured with reverse Y enabled on the right.

Normal Image **Reverse Y Mirror Image**

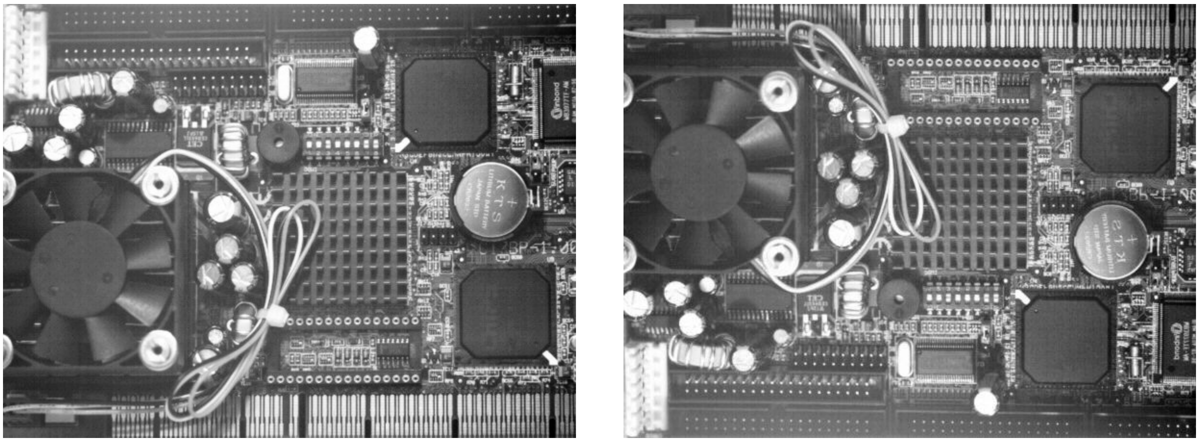


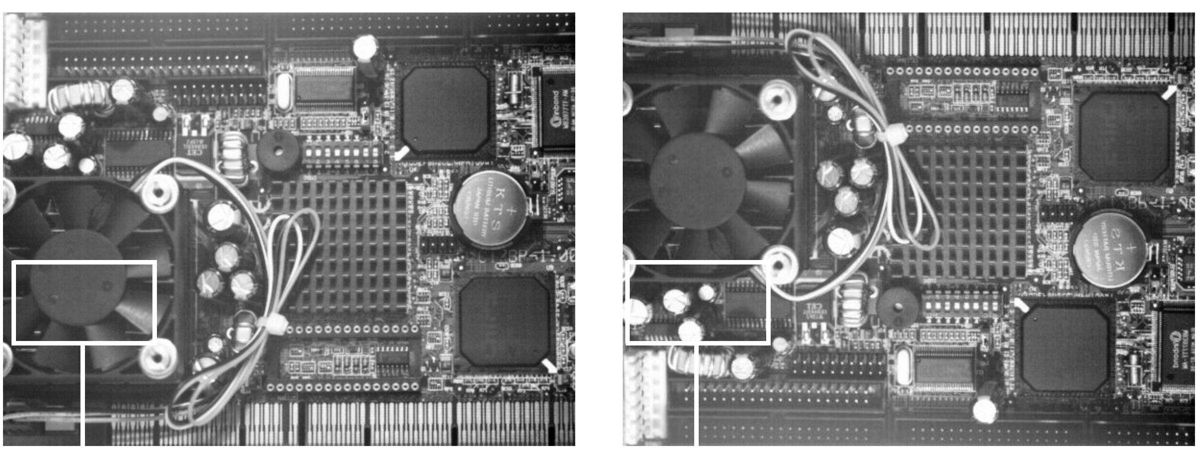
Fig. 49: Reverse Y Mirror Imaging

Using ROIs with Reverse Y

You can use the Image ROI and Auto Function ROI features when using the Reverse Y feature. The position of an ROI relative to the sensor remains the same regardless of whether or not the Reverse Y feature is enabled.

As a consequence, different regions of the image will be controlled or displayed depending on whether or not the Reverse Y feature is enabled. See example in Figure 50.

Normal Image **Reverse Y Mirror Image**



ROI ROI

Fig. 50: Using an ROI with Reverse Y Mirror Imaging

11.9.3 Enabling Reverse X and Reverse Y

You can enable the Reverse X and Reverse Y features by setting the ReverseX and the ReverseY parameter values. You can use the pylon API to set the parameter values from within your application software. The following code snippet illustrates using the API to set the parameter values:

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

// Enable reverse Y
camera.ReverseY.SetValue(true);
```

11.10 Defect Pixel Correction

When analyzing your acquired images, some pixels may appear significantly brighter or darker than the rest, even when uniform light is used ("outlier pixels"). This problem arises from sensitivity differences among the pixels due to production tolerances.

The defect pixel correction minimizes the influence of these sensitivity differences. Basler dart cameras can perform two types of pixel corrections:

- The **static** pixel correction only corrects pixels that have a significantly **lesser** intensity value than its neighboring pixels (or that are always completely black).
- The **dynamic** pixel correction only corrects pixels that have a significantly **greater** intensity value than its neighboring pixels.

You can specify which corrections should be performed by setting the DefectPixelCorrectionMode parameter:

- **On** (default): Static and dynamic pixel corrections are both enabled.
- **Off**: Static and dynamic pixel corrections are both disabled.
- **StaticOnly**: Only the static pixel correction is enabled.

You can set the DefectPixelCorrectionMode parameter by using the pylon API. The following code snippet illustrates using the API to set the pixel correction:

```
// Disable pixel correction
camera.DefectPixelCorrectionMode.SetValue(DefectPixelCorrectionMode_Off);
// Enable static and dynamic pixel correction
camera.DefectPixelCorrectionMode.SetValue(DefectPixelCorrectionMode_On);
// Enable static pixel correction only
camera.DefectPixelCorrectionMode.SetValue(DefectPixelCorrectionMode_StaticOnly);
```

11.11 Auto Functions

Auto functions control image properties and are the automatic counterparts of certain features such as the Gain feature or the Balance White feature, which require manually setting the related parameter values.

	By default, all auto functions (exposure auto, balance white auto, and gain auto) are enabled. They are set to the continuous mode of operation (see Section 11.11.1 on page 110). If you want to manually set the parameter values for gain, exposure time, or white balance, you must first set the corresponding auto function parameter (ExposureAuto, BalanceWhiteAuto, or GainAuto) to Off. After camera power up or reset, all auto functions will be re-enabled unless you define a startup set with auto functions turned off. For more information about the startup set, see Section 11.15 on page 126.

Auto functions are particularly useful to keep good image quality under frequently changing conditions, e.g. unreliable light conditions.

An auto function automatically adjusts a parameter value until the related image property reaches a target value. Each auto function uses the pixel data of the full image as the base for adjusting.

The manual setting of the parameter value is not preserved. For example, when the gain auto function adjusts the Gain parameter value, the manually set Gain parameter value is not preserved.

Generally, the different auto functions can operate at the same time. For more information, see the following sections describing the individual auto functions.

	A target value for an image property can only be reached if it is in accord with all pertinent camera settings and with the general circumstances used for capturing images. Otherwise, the target value will only be approached. For example, with a short exposure time, insufficient illumination, and a low setting for the upper limit of the Gain parameter value, the gain auto function may not be able to achieve the current target value setting for the image.

11.11.1 Auto Function Operating Modes

The following auto function modes of operation are available:

- All auto functions provide the **Once** mode of operation. When the Once mode of operation is selected, the parameter values are automatically adjusted until the related image property reaches the target value. After the automatic parameter value adjustment is complete, the auto function will automatically be set to Off and the new parameter value will be applied to the following images.

The parameter value can be changed by using the Once mode of operation again, by using the Continuous mode of operation, or by manual adjustment.

If an auto function is set to the Once operation mode and the circumstances prevent that a target value for an image property is reached, the auto function will try to reach that target value for a maximum of 50 or 100 images, depending on the frame rate. After that, the auto function is set to Off.

- All auto functions also provide a **Continuous** mode of operation where the parameter value is adjusted repeatedly while images are acquired. This is the default mode of operation.

  Depending on the current frame rate, the automatic adjustments will usually be carried out for every or every other image.

  The repeated automatic adjustment will proceed until the Once mode of operation is used or until the auto function is set to Off.
- When an auto function is set to **Off**, the parameter value resulting from the latest automatic adjustment will operate, unless the parameter is manually adjusted.

You can enable auto functions and change their settings while the camera is capturing images ("on the fly").

If you have set an auto function to Once or Continuous operation mode while the camera was continuously capturing images, the auto function will become effective with a short delay and the first few images may not be affected by the auto function.

11.11.2 Auto Function ROI

The Auto Function Region of Interest (ROI) feature lets you specify a part of the sensor array that will be used for auto function control.

The settings for the Auto Function ROI feature are not tied to the settings for the Image ROI feature. For more information about the Image ROI feature, see Section 11.7 on page 97.

All dart BCON cameras provide two Auto Function ROIs. For both Auto Function ROIs, you can specify a separate part of the sensor array.

Each Auto Function ROI serves as the base for specific auto functions:

- The pixel data from **Auto Function ROI 1** serves as the base for **Exposure Auto** and **Gain Auto**. The Auto Function ROI settings are always identical for both auto functions. This does not imply, however, that Gain Auto and Exposure Auto must always be used at the same time.
- The pixel data from **Auto Function ROI 2** serves as the base for **Balance White Auto**.

These presets are fixed and can't be changed.

The location and size of an Auto Function ROI is defined by declaring an X offset, a Y offset, a width, and a height. Auto Function ROIs are referenced to the top left corner of the sensor array (column 0, row 0).

For example, suppose that you specify the X offset as 14, the width as 5, the Y offset as 7, and the height as 6. This sets the following Auto Function ROI:

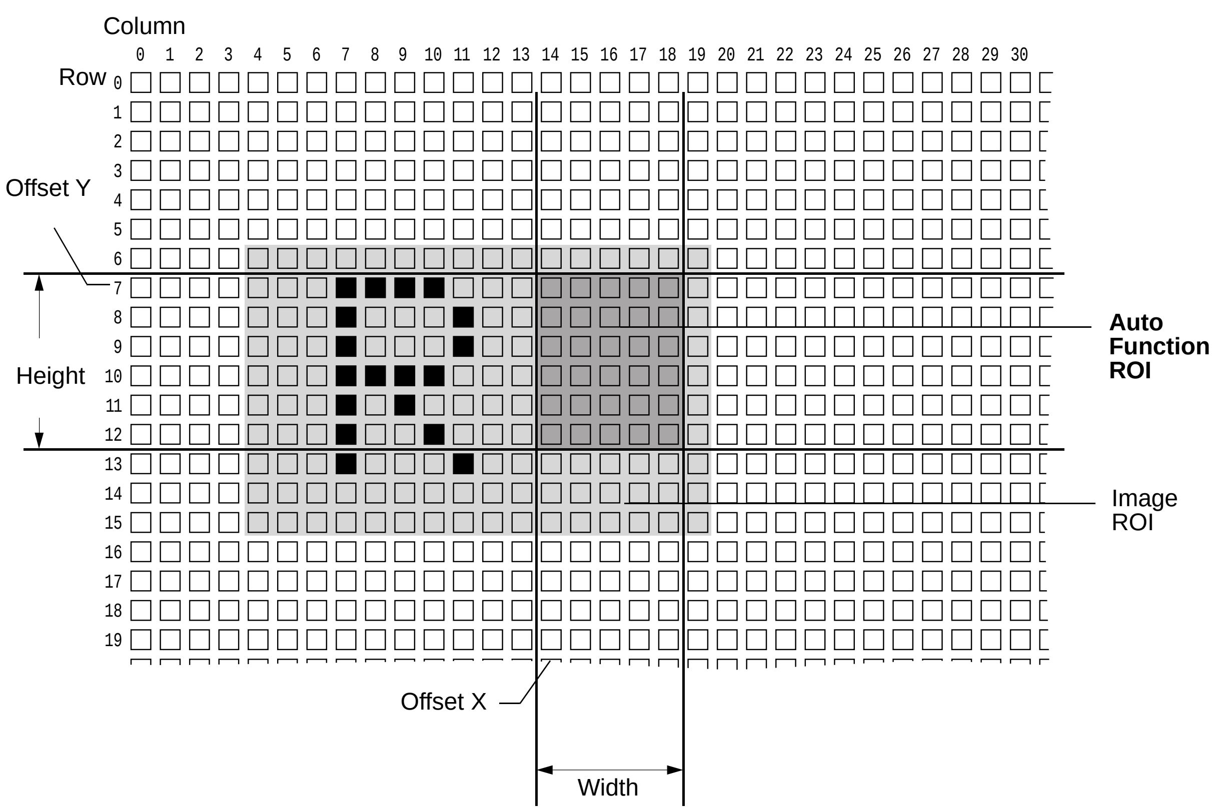


Fig. 51: Auto Function Region of Interest and Image Region of Interest

11.11.2.1 Setting an Auto Function ROI

By default, all Auto Function ROIs are set to the full resolution of the camera's sensor. However, you can change their positions and sizes as desired.

To set an Auto Function ROI:

1. Select the Auto Function ROI that you want to configure. You can do this by setting the AutoFunctionROISelector parameter to ROI1 or ROI2.
2. Set the position and size of the selected Auto Function ROI by changing the following parameters:
   - AutoFunctionROIOffsetX
   - AutoFunctionROIOffsetY
   - AutoFunctionROIWidth
   - AutoFunctionROIHeight

When you are setting an Auto Function ROI, you must follow these guidelines:

Guideline	Example
AutoFunctionROIOffsetX + AutoFunctionROIWidth ≤ Width of camera sensor	daA1600-60um: AutoFunctionROIOffsetX + AutoFunctionROIWidth ≤ 1600
AutoFunctionROIOffsetY + AutoFunctionROIHeight ≤ Height of camera sensor	daA1600-60um: AutoFunctionROIOffsetY + AutoFunctionROIHeight ≤ 1200

You can configure an Auto Function ROI from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to select an Auto Function ROI and set its size and position:

```
// Select Auto Function ROI 1
camera.AutoFunctionROISelector.SetValue(AutoFunctionROISelector_ROI1);
// Set position and size for the selected Auto Function ROI
camera.AutoFunctionROIOffsetX.SetValue(10);
camera.AutoFunctionROIOffsetY.SetValue(10);
camera.AutoFunctionROIWidth.SetValue(500);
camera.AutoFunctionROIHeight.SetValue(400);
```

11.11.2.2 Overlap Between Auto Function ROI and Image ROI

The size and position of an Auto Function ROI can be, but need not be, identical to the size and position of the Image ROI.

The overlap between Auto Function ROI and Image ROI determines whether and to what extent the auto function will control the related image property. Only the pixel data from the areas of overlap will be used by the auto function to control the image property of the entire image.

Different degrees of overlap are illustrated in Figure 52. The hatched areas in the figure indicate areas of overlap.

- If the Auto Function ROI is completely included in the Image ROI (see (a) in Figure 52), the pixel data from the Auto Function ROI will be used to control the image property.
- If the Image ROI is completely included in the Auto Function ROI (see (b) in Figure 52), only the pixel data from the Image ROI will be used to control the image property.
- If the Image ROI only partially overlaps the Auto Function ROI (see (c) in Figure 52), only the pixel data from the area of partial overlap will be used to control the image property.
- If the Auto Function ROI does not overlap the Image ROI (see (d) in Figure 52), the auto function will **not work**.

- The minimum overlap between Auto Function ROI and Image ROI is 4 × 4 pixels. If a lower overlap is set (e.g. 4 × 2 pixels), the related auto function will **not work**.
- Basler strongly recommends completely including the Auto Function ROI within the Image ROI or choosing identical positions and sizes for Auto Function ROI and Image ROI.

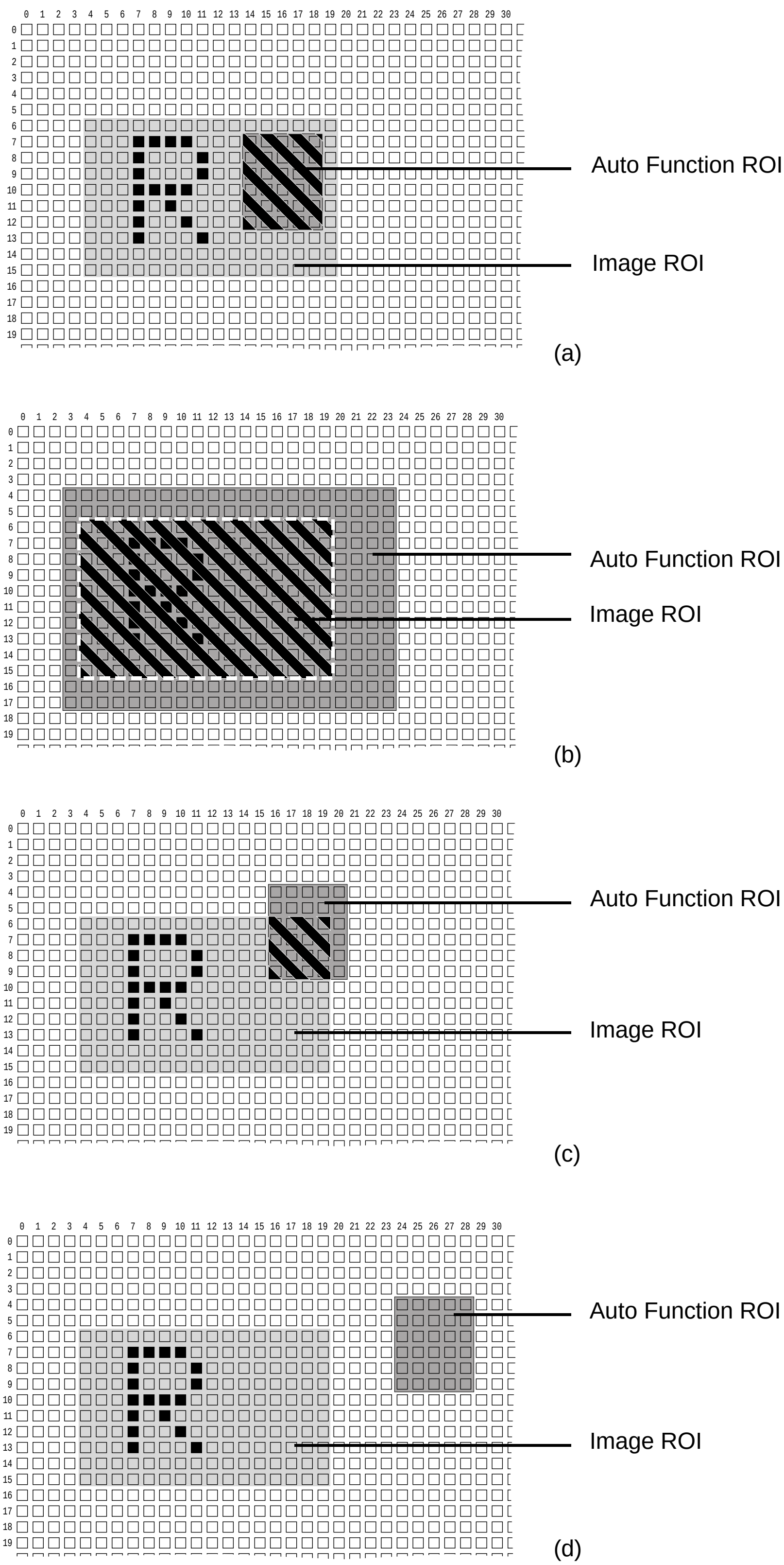


Fig. 52: Various Degrees of Overlap Between the Auto Function ROI and the Image ROI

11.11.3 Gain Auto

Gain Auto is the automatic counterpart to manually setting the Gain parameter. The gain auto function automatically adjusts the Gain parameter value within set limits until a target brightness value for the pixel data is reached.

The gain auto function can be operated in the Once and Continuous modes of operation.

The gain auto function and the exposure auto function can be used at the same time. In this case, however, you must also set the Auto Function Profile feature.

For more information about

- setting the gain manually, see Section 11.2 on page 83.
- the Auto Function Profile feature, see Section 11.11.6 on page 119.

Lower and Upper Limits

The limits within which the camera will adjust the Gain parameter are defined by the AutoGainUpperLimit and the AutoGainLowerLimit parameters.

Example: You set AutoGainLowerLimit to 2 and AutoGainUpperLimit to 6. During automatic adjustments, the Gain parameter value will never be lower than 2 and never be higher than 6.

Target Value

When the gain auto function is enabled, it adjusts the gain until a target brightness, i.e., an average gray value, is reached. You can set the target value using the AutoTargetBrightness parameter.

The parameter value range refers to the theoretically maximum available range of gray values for the set pixel format.

Example: If an 8-bit pixel format is set, the maximum gray value is 256. Therefore, a parameter value of 0.5 corresponds to a gray value of 128.

The target value calculation does not include gamma correction. Depending on the gamma correction value set, images output by the camera may have a significantly lower or higher average gray value than indicated by the AutoTargetBrightness parameter value.

For example, if you set the pixel format to YCbCr422 and AutoTargetBrightness to 0.5, images output by the camera will have a higher average gray value than 128. This is because an additional sRGB gamma correction value is applied after the target value calculation has been performed.

For more information about gamma correction, see Section 11.4 on page 85.

To set the gain auto function:

1. Set the value of the AutoGainLowerLimit and AutoGainUpperLimit parameters.
2. Set the value of the AutoTargetBrightness parameter.
3. Set the value of the GainAuto parameter for the Once or the Continuous mode of operation.

You can set the gain auto function from within your application software by using the pylon API. The following code snippets illustrate using the API to set the gain auto function:

```
// Set the lowest possible lower limit and the highest possible
// upper limit for the gain auto function
camera.AutoGainLowerLimit.SetValue(camera.AutoGainLowerLimit.GetMin());
camera.AutoGainUpperLimit.SetValue(camera.AutoGainUpperLimit.GetMax());
// Set the target average gray value to 60% of the maximum gray value
camera.AutoTargetBrightness.SetValue(0.6);
// Enable Gain Auto by setting the operation mode to Continuous
camera.GainAuto.SetValue(GainAuto_Continuous);
```

For more information about auto functions, see Section 11.11 on page 110.

11.11.4 Exposure Auto

The exposure auto function will not work if the camera's exposure mode is set to trigger width. For more information about the trigger width exposure mode, see Section 9.2.3 on page 63.

Exposure Auto is the automatic counterpart to manually setting the ExposureTime parameter. The exposure auto function automatically adjusts the ExposureTime parameter value within set limits until a target brightness value for the pixel data is reached.

The exposure auto function can be operated in the Once and Continuous modes of operation.

The exposure auto function and the gain auto function can be used at the same time. In this case, however, you must also set the Auto Function Profile feature.

For more information about setting the exposure time manually, see Section 9.3 on page 67.

For more information about the Auto Function Profile feature, see Section 11.11.6 on page 119.

Lower and Upper Limits

The limits within which the camera will adjust the AutoExposureTime parameter are defined by the AutoExposureTimeUpperLimit and the AutoExposureTimeLowerLimit parameters.

Example: You set AutoExposureTimeLowerLimit to 100 and AutoExposureTimeUpperLimit to 5000. During automatic adjustments, the exposure time will never be lower than 100 ms and never be higher than 5000 ms.

If the AutoExposureTimeUpperLimit parameter is set to a sufficiently high value, the camera's frame rate can be decreased.

Target Value

When the exposure auto function is enabled, it adjusts the exposure time until a target brightness, i.e., an average gray value, is reached. You can set the target value using the AutoTargetBrightness parameter.

The parameter value range refers to the theoretically maximum available range of gray values for the set pixel format.

For example, if an 8-bit pixel format is set, the maximum gray value is 256. Therefore, a parameter value of 0.5 corresponds to a gray value of 128.

The target value calculation does not include gamma correction. Depending on the gamma correction value set, images output by the camera may have a significantly lower or higher average gray value than indicated by the AutoTargetBrightness parameter value.

For example, if you set the pixel format to YCbCr422 and AutoTargetBrightness to 0.5, images output by the camera will have a higher average gray value than 128. This is because an additional sRGB gamma correction value is applied after the target value calculation has been performed.

For more information about gamma correction, see Section 11.4 on page 85.

To set the exposure auto function:

1. Set the value of the AutoExposureTimeLowerLimit and AutoExposureTimeUpperLimit parameters.
2. Set the value of the AutoTargetBrightness parameter.
3. Set the value of the ExposureAuto parameter for the Once or the Continuous mode of operation.

You can set the exposure auto function from within your application software by using the pylon API. The following code snippets illustrate using the API to set the exposure auto function:

```
// Set the lowest possible lower limit and the highest possible
```

```
// upper limit for the exposure auto function
camera.AutoExposureTimeLowerLimit.SetValue(camera.AutoExposureTimeLowerLimit.GetMin());
camera.AutoExposureTimeUpperLimit.SetValue(camera.AutoExposureTimeUpperLimit.GetMax());
// Set the target average gray value to 60% of the maximum gray value
camera.AutoTargetBrightness.SetValue(0.6);
// Enable Exposure Auto by setting the operation mode to Continuous
camera.ExposureAuto.SetValue(ExposureAuto_Continuous);
```

For general information about auto functions, see Section 11.11 on page 110.

11.11.5 Balance White Auto

Balance White Auto is the automatic counterpart to manually setting the white balance. The balance white auto function is only available on color models.

For more information about white balance and setting the white balance manually, see Section 11.6.3 on page 92.

To set the balance white auto function using the Basler pylon API, set the value of the BalanceWhiteAuto parameter for the Once or the Continuous mode of operation.

You can set the white balance auto functionality from within your application software by using the pylon API. The following code snippets illustrate using the API to set the balance auto functionality:

```
// Set mode of operation for balance white auto function to Once
camera.BalanceWhiteAuto.SetValue(BalanceWhiteAuto_Once);
// Set mode of operation for balance white auto function to Continuous
camera.BalanceWhiteAuto.SetValue(BalanceWhiteAuto_Continuous);
```

For general information about auto functions, see Section 11.11 on page 110.

11.11.6 Auto Function Profile

The Auto Function Profile feature will only take effect if you use the gain auto function and the exposure auto function at the same time.

The auto function profile specifies how the gain and the exposure time will be balanced when the camera is making automatic adjustments.

If you want to use this feature, you must enable both the gain auto function and the exposure auto function and set both for the continuous mode of operation.

All Basler dart BCON cameras support the following auto function profiles:

- **Minimize Gain**: Gain will be kept as low as possible during automatic adjustments.

- **Minimize Exposure**: Exposure time will be kept as low as possible during automatic adjustments.
- **Smart** (default): Gain will be kept as low as possible and the frame rate will be kept as high as possible during automatic adjustments. This is a four-step process:
  1. The camera will adjust the exposure time to achieve the target brightness value.
  2. If the exposure time must be increased to achieve the target brightness value, the camera increases the exposure time until a lowered frame rate is detected.
  3. If a lowered frame rate is detected, the camera stops increasing the exposure time and increases gain until the AutoGainUpperLimit value is reached.
  4. If the AutoGainUpperLimit is reached, the camera stops increasing gain and increases the exposure time until the target brightness value is reached. This will result in a lower frame rate.
- **Anti-Flicker 50 Hz / Anti-Flicker 60 Hz**: Gain and exposure time will be optimized to reduce flickering. If the camera is operating in an environment where the lighting flickers at a 50-Hz or a 60-Hz rate, the flickering lights can cause significant changes in brightness from image to image. Enabling the anti-flicker profile may reduce the effect of the flickering in the captured images.
  Depending on your local power line frequency (e.g. North America: 60 Hz, Europe: 50 Hz), set the auto function profile to AntiFlicker50Hz or to AntiFlicker60Hz.

To set the auto function profile:

1. Set the value of the AutoFunctionProfile parameter to
   - AutoFunctionProfile_MinimizeGain,
   - AutoFunctionProfile_MinimizeExposureTime,
   - AutoFunctionProfile_Smart,
   - AutoFunctionProfile_AntiFlicker50Hz, or
   - AutoFunctionProfile_AntiFlicker60Hz.
2. Set the value of the GainAuto parameter to the Continuous mode of operation.
3. Set the value of the ExposureAuto parameter to the Continuous mode of operation.

You can set the auto function profile from within your application software by using the pylon API. The following code snippet illustrates using the API to set the auto function profile. As an example, the MinimizeGain auto function profile is set:

```
// Keep gain as low as possible during automatic adjustments
camera.AutoFunctionProfile.SetValue(AutoFunctionProfile_MinimizeGain);
camera.GainAuto.SetValue(GainAuto_Continuous);
camera.ExposureAuto.SetValue(ExposureAuto_Continuous);
```

11.12 Backlight Compensation

i	The Backlight Compensation feature will only take effect if the gain auto function or the exposure auto function or both are enabled. For more information about auto functions, see Section 11.11 on page 110.

If a bright light comes from behind your image subject, the subject may be underexposed and appear silhouetted. The Backlight Compensation feature allows the camera to compensate for this underexposure.

You can adjust the backlight compensation by setting the value of the AutoBacklightCompensation parameter. The parameter value can range from 0 to 0.5.

When the AutoBacklightCompensation parameter value is set, a given percentage of the brightest pixels in the image (i.e. the pixels with the highest pixel values) will not be taken into account for the target value calculations. These calculations are performed by the gain auto function and the exposure auto function (see Section 11.11.3 on page 116 and Section 11.11.4 on page 117).

For example, if you set the parameter value to 0.3, then 30 % of the brightest pixels in the image will not be taken into account for the target value calculations.

This allows the camera to properly expose the darker regions of the image.

You can set the backlight compensation from within your application software by using the pylon API. The following code snippets illustrate using the API to set the AutoBacklightCompensation parameter value:

```
// Set the AutoBacklightCompensation parameter value to 0.3
camera.AutoBacklightCompensation.SetValue(0.3);
```

11.13 Test Patterns

All cameras include the ability to generate test patterns. Test patterns are used to check the camera’s basic functionality and its ability to transmit an image.

Test patterns can be used for service purposes and for failure diagnostics.

Enabling a Test Pattern

The TestPattern parameter is used to set the camera to output a test pattern. You can set the value of the TestPattern parameter to one of the test patterns or to Off.

You can set the TestPattern parameter from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to set the parameter:

```
// Set for no test pattern
camera.TestPattern.SetValue(TestPattern_Off);
// Set for the first test pattern
camera.TestPattern.SetValue(TestPattern_GreyDiagonalSawtooth8);
// Set for the second test pattern
camera.TestPattern.SetValue(TestPattern_ColorDiagonalSawtooth8);
```

Test Pattern 1: Gray Diagonal Sawtooth (8 bit)

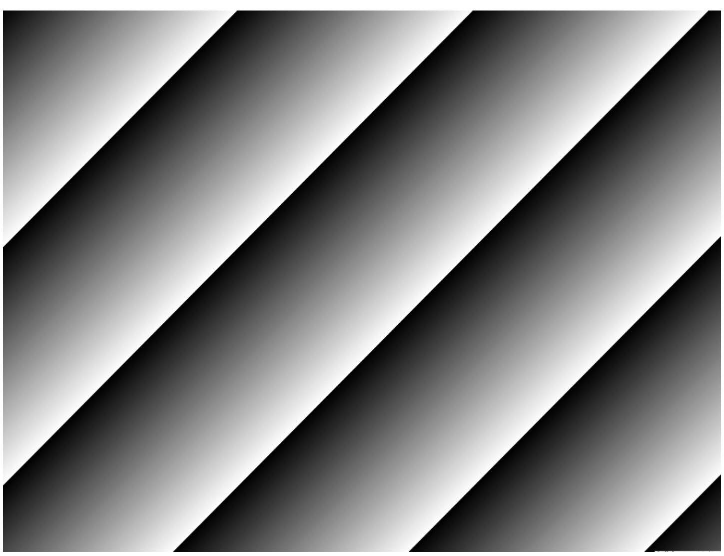


Fig. 53: Test Pattern 1

The test pattern "Gray Diagonal Sawtooth" (8 bit) is best suited for use when the camera is set for monochrome 8-bit output. The test pattern 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 pattern 1 will look similar to Figure 53.

Test Pattern 2: Color Diagonal Sawtooth

The test pattern "Color Diagonal Sawtooth" is available on color cameras only. As shown in Figure 54, the test pattern consists of diagonal color gradients.

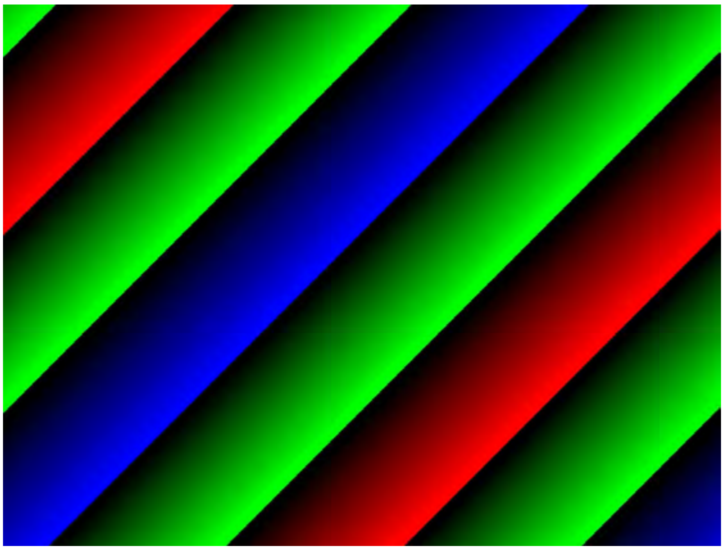


Fig. 54: Test Pattern 2

11.14 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:

- DeviceVendorName (read only) - contains the camera vendor's name.
- DeviceModelName (read only) - contains the model name of the camera.
- DeviceManufacturerInfo (read only) - can contain some information about the camera manufacturer. This string is usually empty.
- DeviceVersion (read only) - contains the device version number for the camera.
- DeviceFirmwareVersion (read only) - contains the version of the firmware in the camera.
- DeviceSerialNumber (read only) - contains the serial number of the camera.
- DeviceUserID (read / write) - is used to assign a user-defined name to a device. The name will be visible in the "friendly name" field of the device information objects returned by pylon's device enumeration procedure.
- DeviceScanType (read only) - contains the scan type of the camera, for example, area scan.
- SensorWidth (read only) - contains the physical width of the sensor in pixels.
- SensorHeight (read only) - contains the physical height of the sensor in pixels.
- MaxWidth (read only) - Indicates the camera's maximum region of interest (ROI) width setting for the current OffsetX settings.
- MaxHeight (read only) - Indicates the camera's maximum region of interest (ROI) height setting for the current OffsetY settings.

You can read the values for all of the device information parameters or set the value of the DeviceUserID parameter from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to read the parameters or set the DeviceUserID parameter:

```
// Read the DeviceVendorName parameter
String_t s = camera.DeviceVendorName.GetValue();

// Read the DeviceModelName parameter
String_t s = camera.DeviceModelName.GetValue();

// Read the DeviceManufacturerInfo parameter
String_t s = camera.DeviceManufacturerInfo.GetValue();

// Read the DeviceVersionparameter
String_t s = camera.DeviceVersion.GetValue();

// Read the DeviceFirmwareVersion parameter
String_t s = camera.DeviceFirmwareVersion.GetValue();

// Read the DeviceSerialNumber parameter
String_t s = camera.DeviceSerialNumber.GetValue();
```

```
// Set and read the DeviceUserID parameter
camera.DeviceUserID.SetValue("CAM_1");
String_t s = camera.DeviceUserID.GetValue();

// Read the DeviceScanType parameter
DeviceScanTypeEnums e = camera.DeviceScanType.GetValue();

// Read the SensorWidth parameter
int64_t i = camera.SensorWidth.GetValue();

// Read the SensorHeight parameter
int64_t i = camera.SensorHeight.GetValue();

// Read the WidthMax parameter
int64_t i = camera.WidthMax.GetValue();

// Read the HeightMax parameter
int64_t i = camera.HeightMax.GetValue();
```

11.15 Configuration Sets and User Sets

A configuration set is a group of parameter values with all the settings needed to control the camera.

There are three basic types of configuration sets: the active set, the factory set, and user sets. In addition, you can designate a startup set.

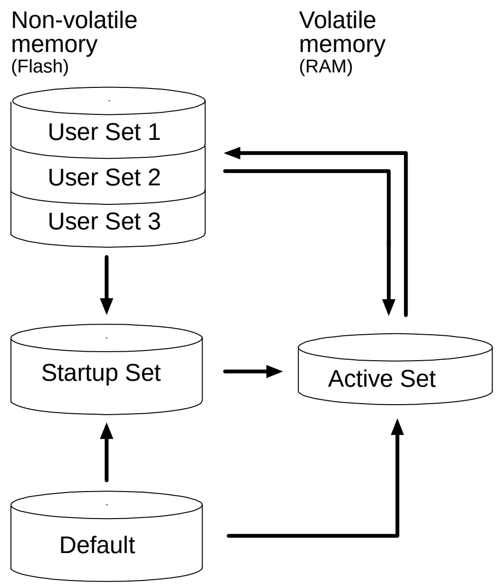


Fig. 55: Configuration Sets

Active Set

The active set contains the camera's current parameter settings and thus determines the camera's performance, that is, what your image currently looks like. When you change parameter settings using the pylon API or direct register access, you are making changes to the active set. The active set is located in the camera's volatile memory and the settings are lost, if the camera is reset or if power is switched off.

Factory Set (Default)

When a camera is manufactured, numerous tests are performed on the camera and a factory optimized setup is determined. This factory setup is stored in the Default set. It is optimized for average conditions and will provide good camera performance in many common applications.

In the Default set,

- the gain auto, exposure auto, and white balance auto functions are set to the continuous mode of operation,
- the Smart auto function profile is set.

The Default set is saved in the camera's non-volatile memory. It is not lost when the camera is reset or switched off and it can't be changed.

For more information about

- auto functions, see Section 11.11 on page 110.
- the Smart auto function profile, see Section 11.11.6 on page 119.

User Sets

There are three reserved areas in the camera’s non-volatile memory available for saving configuration sets that can be customized by the user. These sets are not lost when the camera is reset or switched off. They are commonly referred to as "user sets".

The three available user sets are called User Set 1, User Set 2, and User Set 3.

When the camera is running, a saved user set can be loaded into the active set. A saved user set can also be designated as the "startup" set, i.e., the set that will be loaded into the active set whenever the camera is powered on or reset.

Startup Set

You can designate the factory set or one of the user sets as the "startup" set. The designated startup set will automatically be loaded into the active set whenever the camera starts up at power on or after a reset.

For more information about designating the startup set, see Section 11.15.3 on page 129.

11.15.1 Saving a User Set

After setting your camera parameter values, you can save most of the settings for further use into User Set 1, User Set 2, or User Set 3. These user sets are not lost when the camera is reset or switched off.

To save the currently active set into a user set:

1. Make changes to the camera’s settings until the camera is operating in a manner that you would like to save.
2. Set the UserSetSelector parameter to UserSet1, UserSet2, or UserSet3.
3. Execute a UserSetSave command to save the active set to the selected user set.

Saving an active set to a user set

- will overwrite any parameters that were previously saved in that user 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.

You can save a user set from within your application software by using the pylon API. The following code snippet illustrates using the API to select User Set 1 and to execute the save command:

```
camera.UserSetSelector.SetValue(UserSetSelector_UserSet1);
camera.UserSetSave.Execute();
```

11.15.2 Loading a User Set or the Factory Set into the Active Set

You can load any user set (User Set 1, User Set 2, or User Set 3) or the factory set (Default) from the camera's non-volatile memory into the camera's active set.

The loaded set overwrites the parameter settings in the active set. The settings from the loaded set will now be controlling the camera.

To load a user set or the factory set into the active set:

1. Set the UserSetSelector to UserSet1, UserSet2, UserSet3, or to Default.
2. Execute a UserSetLoad command to load the selected set into the active set.

You can set the UserSetSelector and execute the UserSetLoad command from within your application software by using the pylon API. The following code snippets illustrate using the API select User Set 2 and execute the load command:

```
camera.UserSetSelector.SetValue(UserSetSelector_UserSet2);
camera.UserSetLoad.Execute();
```

Loading a user set or the factory set into the active set is only allowed when the camera is idle, i.e. when it is not acquiring images continuously or does not have a single image acquisition pending.

Loading the Default set with the standard factory setup into the active set is recommended if you have misadjusted the settings in the camera and you are not sure how to recover. The standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

11.15.3 Designating the Startup Set

You can designate any user set (User Set 1, User Set 2, or User Set 3) or the factory set (Default) as 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.

Designating a startup 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.

To designate a user set or the factory set as the startup set:

- Set the UserSetDefault parameter to UserSet1, UserSet2, UserSet3, or to Default.

You can set the UserSetDefault parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to set the selector:

```
camera.UserSetDefault.SetValue(UserSetDefault_UserSet1);
```

If you have misadjusted the settings in the cameras and you are not sure how to recover, do the following:

1. Set the UserSetDefault parameter to Default.
2. Restart the camera.

12 Troubleshooting and Support

12.1 Tech Support Resources

If you need advice about your camera or if you need assistance troubleshooting a problem with your camera, you can contact the Basler technical support team for your area. Basler technical support contact information is located in the front pages of this manual.

You will also find helpful information such as frequently asked questions, downloads, and application notes in the Support and Downloads sections of the Basler website: www.baslerweb.com

If you do decide to contact Basler technical support, please take a look at Section 12.3 on page 131 before you call. The section gives information about assembling relevant data that will help the Basler technical support team to help you with your problem.

12.2 Obtaining an RMA Number

Whenever you want to return material to Basler, you must request a Return Material Authorization (RMA) number before sending the material back. The RMA number **must** be stated in your delivery documents when you ship your material to us! Please be aware that, if you return material without an RMA number, we reserve the right to reject the material.

You can find detailed information about how to obtain an RMA number in the Support section of the Basler website: www.baslerweb.com

12.3 Before Contacting Basler Technical Support

To help you as quickly and efficiently as possible when you have a problem with a Basler camera, it is important that you collect several pieces of information before you contact Basler technical support. Basler technical support contact information is shown in the title section of this manual.

To provide data to Basler technical support, use the form given below.

To use the form:

1. Copy the form that appears below, fill it out, and send it - with sample images if appropriate - attached to your e-mail to Basler technical support
or
fax the completed form with the requested files attached to your local dealer or to Basler technical support.

1	The camera's product ID:	______________________
2	The camera's serial number:	______________________
3	Hardware 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.

If you can't access the camera, please try to state the following parameter settings:

- ☐ Image Size (ROI): ____________________________________
- ☐ Pixel Format: ____________________________________
- ☐ Exposure Time: ____________________________________
- ☐ Frame Rate: ____________________________________

11 Live image/test pattern

If you are having an image problem, try to generate and save live images that show the problem. Also generate and save test patterns. Please save the images in BMP format, zip them, and send them to Basler technical support.

Index

H

I

L

M

O

P

R

S

T

U

V

W

Revision History

Doc. ID Number	Date	Changes
AW001369**01**000	23 Jun 2016	Initial release of this document.
AW001369**02**000	01 Nov 2016	Modifications and corrections related to the development of the camera from prototype to serial production.