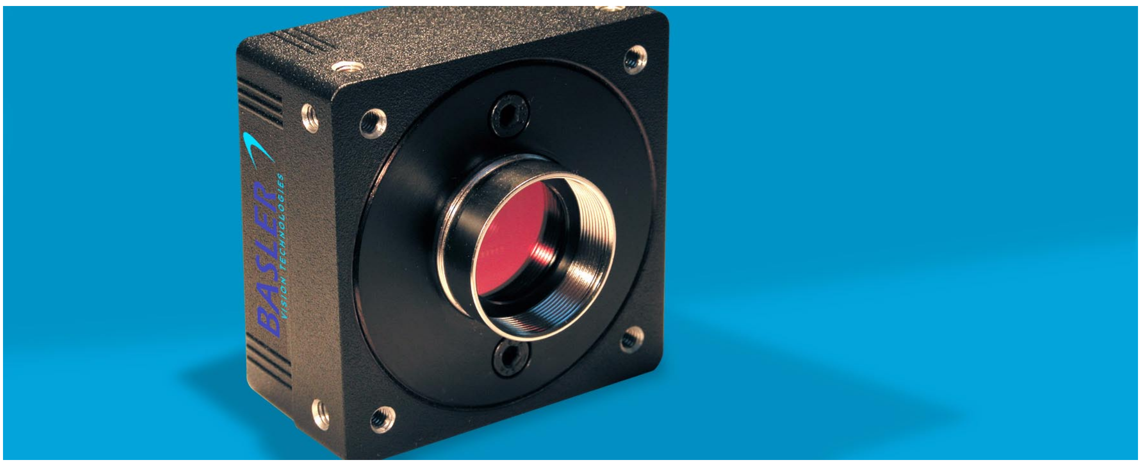
BASLER A102f



USER'S MANUAL

Document Number: DA00063003
Release Date: 22 March 2004

BASLER
VISION TECHNOLOGIES

For customers in the U.S.A.

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

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

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

For customers in Canada

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

Pour utilisateurs au Canada

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

Life Support Applications

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

Warranty Note

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

Contacting Basler Support Worldwide

Europe:

Basler AG
Ander Strusbek 60 - 62
22926 Ahrensburg
Germany

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

vc.support.europe@baslerweb.com

Americas:

Basler, Inc.
740 Springdale Drive, Suite 100
Exton, PA 19341
U.S.A.

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

vc.support.usa@baslerweb.com

Asia:

Basler Asia PTe. Ltd
25 Internat. Business Park
#04-15/17 German Centre
Singapore 609916

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

vc.support.asia@baslerweb.com

www.basler-vc.com

Table of Contents

1 Introduction

2 Camera Interface

3 Basic Operation and Standard Features

1 Introduction

1.1 Documentation Applicability

This User's Manual applies to cameras with a firmware ID number of 20.

Cameras with a lower or a higher firmware ID number may have fewer features or have more features than described in this manual. Features on cameras with a lower or a higher firmware ID number may not operate exactly as described in this manual.

An easy way to see the firmware ID number for an **A102f** camera is by using the BCAM Viewer included with the Basler BCAM 1394 driver. To see the firmware ID number:

1. Attach your camera to a computer equipped with the BCAM 1394 driver.
2. Double click the BCAM Viewer icon on your desktop or click Start ⇒ All Programs ⇒ Basler Vision Technologies ⇒ BCAM 1394 ⇒ BCAM Viewer. The viewer program window will open.
3. Find the camera name in the Bus Viewer panel that appears on the left side of the window and click on the camera name.
4. Click on the  icon in the tool bar at the top of the window.
5. A properties window similar to the one shown in Figure 1-1 will open. Use the figure as a guide to find the firmware ID number.

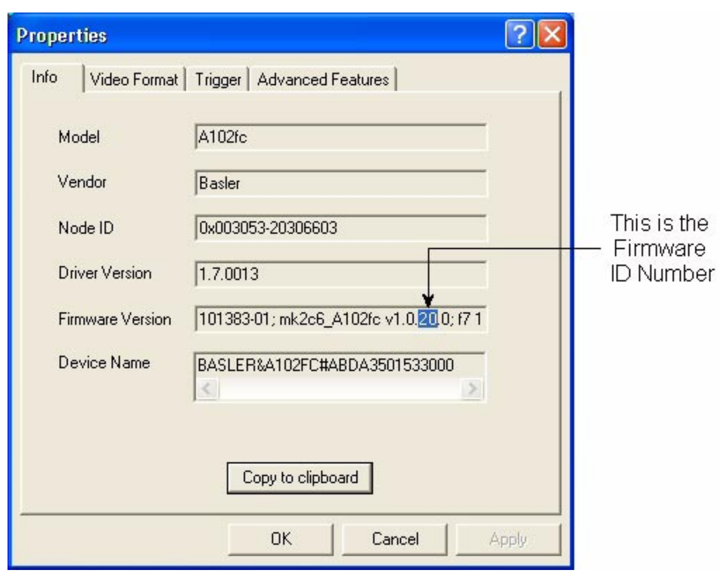


Figure 1-1: BCAM Properties Window

ⓘ	You can also access the firmware ID number by using the Extended Version Information smart feature. See Section 5.7.7 for more information.

1.2 Performance Specifications

Category	**Specification**
Sensor Type	Sony ICX-285 Progressive Scan CCD Sensor
Pixels	A102f: 1392 (H) × 1040 (V) A102fc: 1388 (H) x 1038 (V)
Pixel Size	6.45 µm (H) × 6.45 µm (V)
Anti-blooming	Yes
Max. Frame Rate (at full resolution)	15.1 frames/s (in 8 bit output modes) 11.3 frames/s (in 16 bit output modes)
Video Output Formats	A102f: Mono 8, 8 bits/pixel Mono 16, 16 bits per pixel (12 bits effective) A102fc: Mono 8, 8 bits/pixel Raw 8, 8 bits/pixel Raw 16, 16 bits/pixel (12 bits effective) YUV 4:2:2, 16 bits/pixel average
Gain and Brightness	Programmable via IEEE 1394 bus
Exposure Time Control	Programmable via IEEE 1394 bus
Synchronization	External via External Trigger signal
Power Requirements	+8.0 to +36.0 VDC (+12 VDC nominal), < 1% ripple 3.5 W max @ 12 VDC supplied via 1394 cable
I/O Electrical Characteristics	Inputs: opto-isolated, 5 VDC nominal, 10 mA nominal Outputs: opto-isolated, 2 to 35 VDC maximum forward voltage, 100 mA max collector current (See Sections 2.4.1 and 2.4.2 for more details.)
Max. Cable Lengths	1394: 4.5 m I/O: 10 m (See Section 2.2 for more details.)
Lens Adapter	C-mount
Housing Size (L x W x H)	without lens adapter: 31.5 mm x 62 mm x 62 mm with c-mount adapter: 67.3 mm x 62 mm x 62 mm
Weight	~ 240 g
Conformity	CE, FCC

Table 1-1: **A102f** Performance Specifications

1.3 Camera Models

The camera is available in a monochrome model (the **A102f**) and a color model (the **A102fc**). Throughout the manual, the camera will be called the **A102f**. Passages that are only valid for a specific model will be so indicated.

1.4 Spectral Response

The spectral response for the **A102f** monochrome cameras is shown in Figure 1-2.

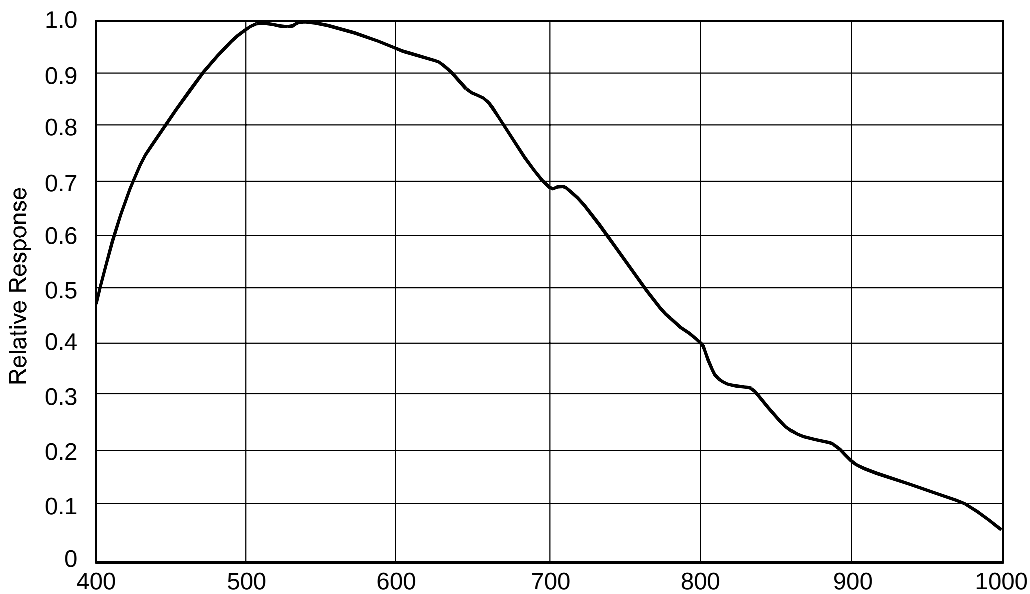


Figure 1-2: **A102f** Spectral Response - Monochrome Cameras

ⓘ	The spectral response curve excludes lens characteristics and light source characteristics.

The spectral response for **A102fc** color cameras is shown in Figure 1-3.

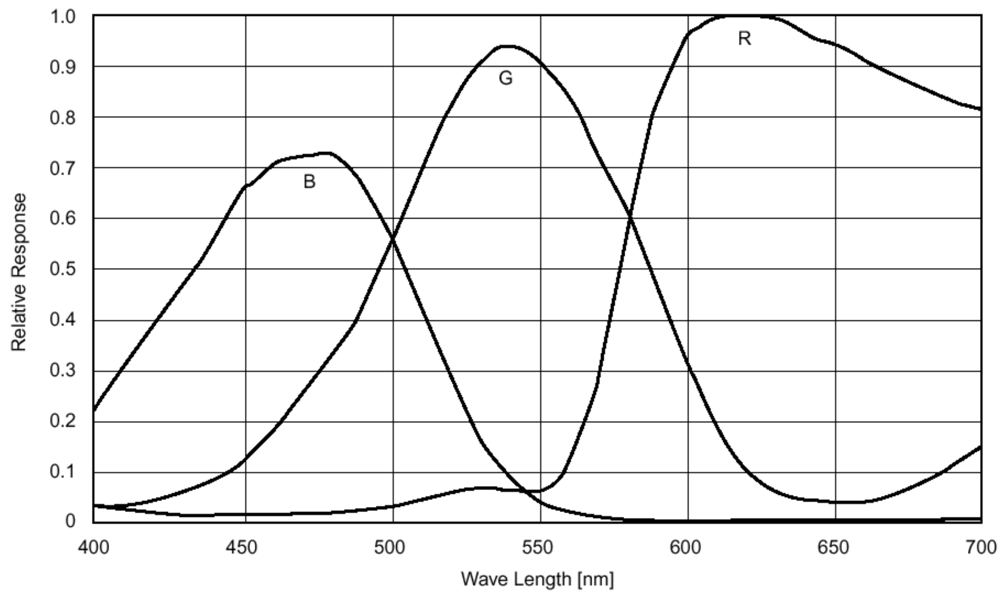


Figure 1-3: **A102f** Spectral Response - Color Cameras

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

To get the best performance from **A102fc** color cameras, use of a dielectric IR cut-off filter is recommended. The filter should transmit in a range of 400 nm to 700...720 nm, and it should cut off from 700...720 nm to 1100 nm.

A suitable filter is included in the standard C-mount adapter on **A102fc** cameras.

1.5 Environmental Requirements

1.5.1 Temperature and Humidity

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

1.5.2 Ventilation

Allow sufficient air circulation around the camera to prevent internal heat build-up in your system and to keep the camera housing temperature during operation below 50° C. Provide additional cooling such as fans or heat sinks if necessary.

1.6 Precautions

To ensure that your warranty remains in force:

Read the manual

Read the manual carefully before using the camera!

Keep foreign matter outside of the camera

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

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

Electromagnetic Fields

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

Transporting

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

Cleaning

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

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

Do not use volatile solvents such as benzine and thinners; they can damage the surface finish.

2 Camera Interface

2.1 Connections

2.1.1 General Description

The **A102f** is interfaced to external circuitry via an IEEE 1394 socket and a 10 pin RJ-45 jack located on the back of the housing. Figure 2-1 shows the location of the two connectors.

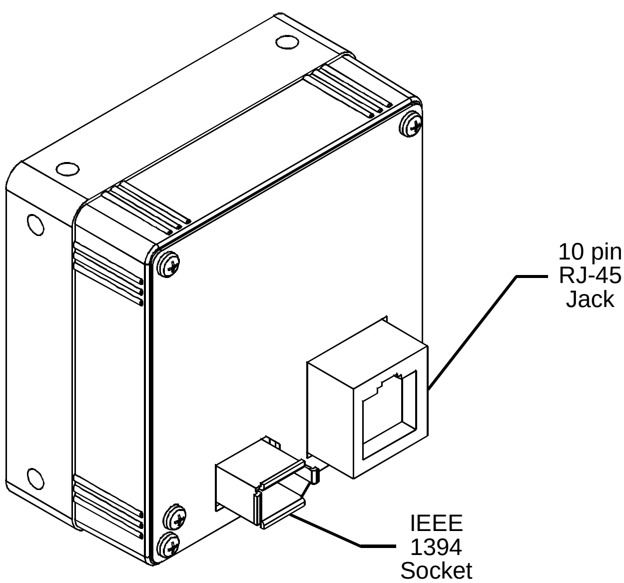


Figure 2-1: Camera Connectors

2.1.2 Pin Assignments

The IEEE 1394 socket is used to supply power to the camera and to interface video data and control signals. The pin assignments for the socket are shown in Table 2-1.

Pin	Signal
1	Power Input (+8.0 to +36.0 VDC)
2	DC Gnd
3	TPB -
4	TPB +
5	TPA -
6	TPA +

Table 2-1: Pin Assignments for the IEEE 1394 Socket

The RJ-45 jack is used to access the four physical input ports and four physical output ports on the camera. The pin assignments for the jack are shown in Table 2-2.

Pin	Designation
1	Output Port 3 -
2	Output Port 2 -
3	Output Port 1-
4	Output Port 0 -
5	Input Port 0 +
6	In Gnd Comm
7	Out VCC Comm
8	Input Port 2 +
9	Input Port 1 +
10	Input Port 3 +

Table 2-2: Pin Assignments for the RJ-45 jack

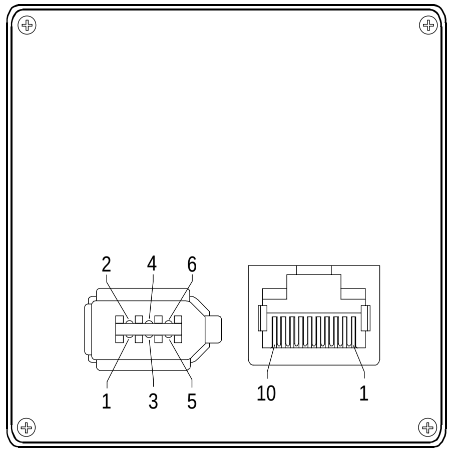


Figure 2-2: **A102f** Pin Numbering

ⓘ	The camera housing is connected to the cable shields and coupled to signal ground through an RC network (see Figure 2-3 for more details).

2.1.3 Connector Types

The 6-pin connector on the camera is a standard IEEE-1394 socket.

The 10-pin connector on the camera is an RJ-45 jack.

⚠	**Caution!** The plug on the cable that you attach to the camera's RJ-45 jack must have 10 pins. Use of a smaller plug, such as one with 8 pins or 4 pins, can damage the pins in the RJ-45 jack on the camera.

2.2 Cables

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

The maximum length of the I/O cable is at least 10 meters. The cable must be shielded and must be constructed with twisted pair wire. Close proximity to strong magnetic fields should be avoided.

2.3 Camera Power

Power must be supplied to the camera via the IEEE 1394 cable. Nominal input voltage is +12.0 VDC, however, the camera will operate properly on any input voltage from +8.0 VDC to +36.0 VDC as specified in the IEEE 1394 standard. Maximum power consumption for the **A102f** is 3.5 W at 12 VDC. Ripple must be less than 1%.

Caution!

Use only standard IEEE 1394 connectors.

The polarity of the input power to the camera must be as shown in Table 2-1. **Do not** reverse the input power polarity. Reversing the polarity will damage the camera.

2.4 Input and Output Ports

2.4.1 Input Ports

A102f cameras are equipped with four physical input ports designated as Input Port 0, Input Port 1, Input Port 2, and Input Port 3. The input ports are accessed via the 10 pin RJ-45 jack on the back of the camera. See Table 2-2 and Figure 2-2 for input port pin assignments and pin numbering.

As shown in Figure 2-3, each input port is opto-isolated. The nominal input voltage for the LED in the opto-coupler is 5.0 V (± 1.0 V). The input current for the LED is 5 to 15 mA with 10 mA recommended.

For each input port, a current between 5 and 15 mA means a logical one. A current of less than 0.1 mA means a logical zero.

By default, Input Port 0 is assigned to receive an external trigger (ExTrig) signal that can be used to control the start of exposure. For more information about the ExTrig signal and for information on assigning the ExTrig signal to a different input port, see Section 3.2.5.

ⓘ	As stated above, the nominal input voltage for the LED on each input is +5 VDC. If a 560 Ohm resistor is added to the positive line for an input, the input voltage can be 12 VDC. If a 1.2 or 1.5 kOhm resistor is added to the positive line for an input, the input voltage can be 24 VDC.

2.4.2 Output Ports

A102f cameras are equipped with four physical output ports designated as Output Port 0, Output Port 1, Output Port 2, and Output Port 3. The output ports are accessed via the 10 pin RJ-45 jack on the back of the camera. See Table 2-2 and Figure 2-2 for output port pin assignments and pin numbering.

As shown in Figure 2-3, each output port is opto-isolated. The minimum forward voltage is 2 V, the maximum forward voltage is 35 V, the maximum reverse voltage is 6 V, and the maximum collector current is 100 mA.

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

By default, Output Port 0 is assigned to transmit an integration enabled (IntEn) signal that indicates when exposure is taking place. For more information about the IntEn signal, see Section 3.4.

By default, Output Port 1 is assigned to transmit a trigger ready (TrigRdy) signal that goes high to indicate the earliest point at which exposure start for the next frame can be triggered. For more information about the TrigRdy signal, see Section 3.3.

The assignment of camera output signals to physical output ports can be changed by the user. See Section 5.7.11 for more information about configuring output ports.

ⓘ	By default, output ports 0, 1, and 2 are set to a low state after power on. Output port 3 is initially set to low but will go high approximately 100 to 300 ms after power on. Output port 3 will remain high for approximately 750 ms and will then reset to low.

2.5 Pixel Data

Pixel data is transmitted as isochronous data packets according to version 1.20 of the “1394 - based Digital Camera Specification” (DCAM) issued by the 1394 Trade Association (see the trade association’s web site: www.1394ta.org). The first packet of each frame is identified by a 1 in the sync bit of the packet header.

Pixel Data Transmission Sequence

Pixel data is transmitted in the following sequence:

- Row 0/Pixel 0, Row 0/Pixel 1, Row 0/Pixel 2 ... Row 0/Pixel 1390, Row 0/Pixel 1391.
- Row 1/Pixel 0, Row 1/Pixel 1, Row 1/Pixel 2 ... Row 1/Pixel 1390, Row 1/Pixel 1391.
- Row 2/Pixel 0, Row 2/Pixel 1, Row 2/Pixel 2 ... Row 2/Pixel 1390, Row 2/Pixel 1391.
- And so on.

2.6 IEEE 1394 Device Information

The **A102f** uses an IEEE 1394a - 2000 compliant physical layer device to transmit pixel data. Detailed spec sheets for devices of this type are available at the Texas Instruments web site (www.ti.com).

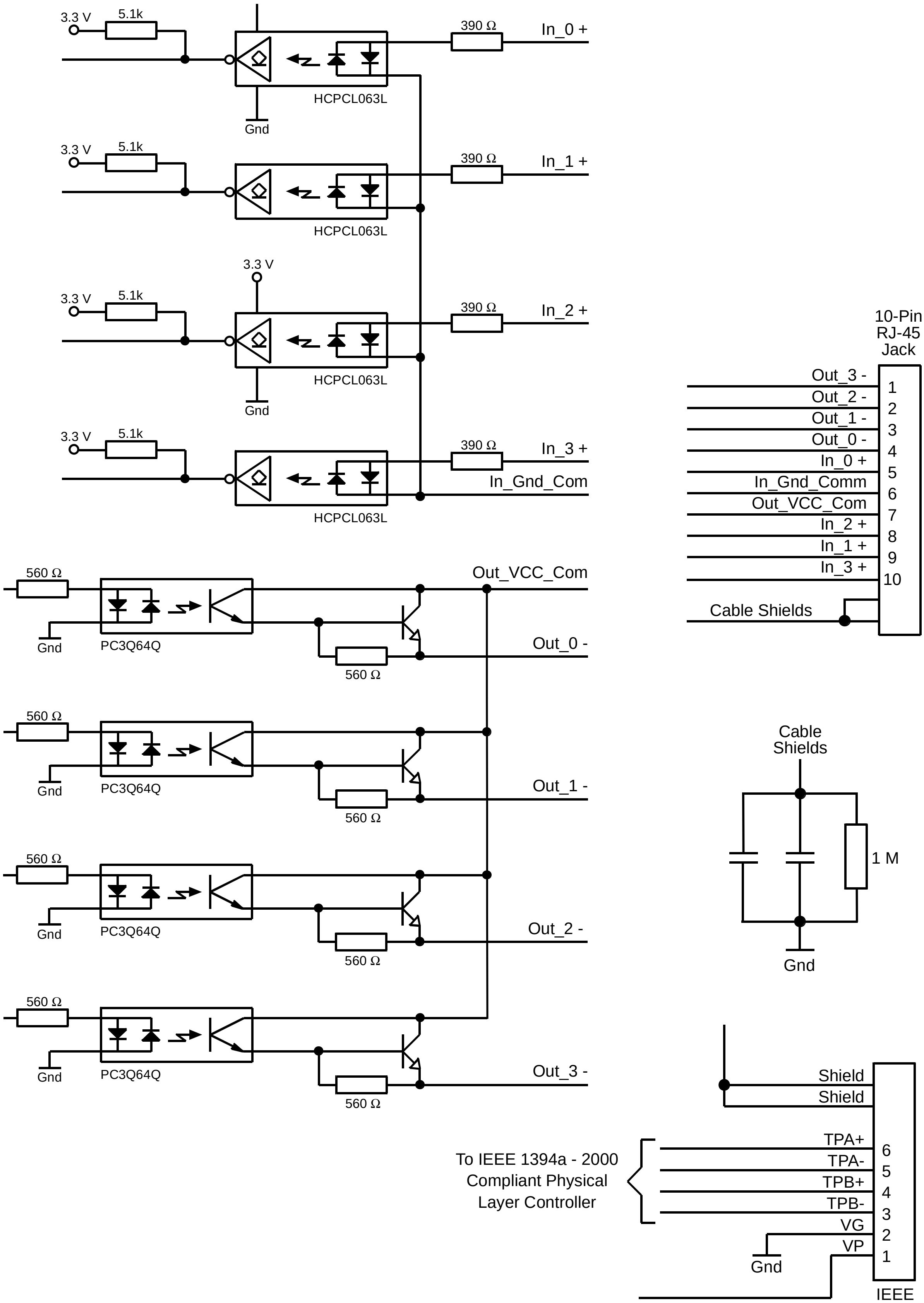


Figure 2-3: I/O Schematic

3 Basic Operation and Standard Features

3.1 Functional Description

3.1.1 Overview

A102f area scan cameras employ a CCD sensor chip which provides features such as a full frame shutter and electronic exposure time control.

Normally, exposure time and charge readout are controlled by values transmitted to the camera's control registers via the IEEE 1394 interface. Control registers are available to set exposure time and frame rate. There are also control registers available to set the camera for single frame capture or continuous frame capture.

Exposure start can also be controlled via an externally generated trigger (ExTrig) signal. The ExTrig signal facilitates periodic or non-periodic start of exposure. When exposure start is controlled by a rising ExTrig signal and the camera is set for the programmable exposure mode, exposure begins when the trigger signal goes high and continues for a pre-programmed period of time. Accumulated charges are read out when the programmed exposure time ends.

At readout, accumulated charges are transported from the sensor's light-sensitive elements (pixels) to the vertical shift registers (see Figure 3-1). The charges from the bottom line of pixels in the array are then moved into a horizontal shift register. Next, the charges are shifted out of the horizontal register through an FPGA and into an image buffer. Shifting is clocked according to the camera's internal data rate.

As the charges move out of the horizontal shift register, they are converted to voltages which are proportional to the size of each charge. The voltages are amplified by an internal Variable Gain Control (VGC) and then digitized by a 12 bit, Analog-to-Digital converter (ADC). For optimal digitization, gain and brightness can be programmed by setting command registers in the camera.

The data leaves the image buffer and passes back through the FPGA to a 1394 link layer controller where it is assembled into data packets that comply with version 1.20 of the "1394 - based Digital Camera Specification" (DCAM) issued by the 1394 Trade Association. The packets are passed to a 1394 physical layer controller which transmits them isochronously to a 1394 interface board in the host PC. The physical and link layer controllers also handle transmission and receipt of asynchronous data such as programming commands.

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

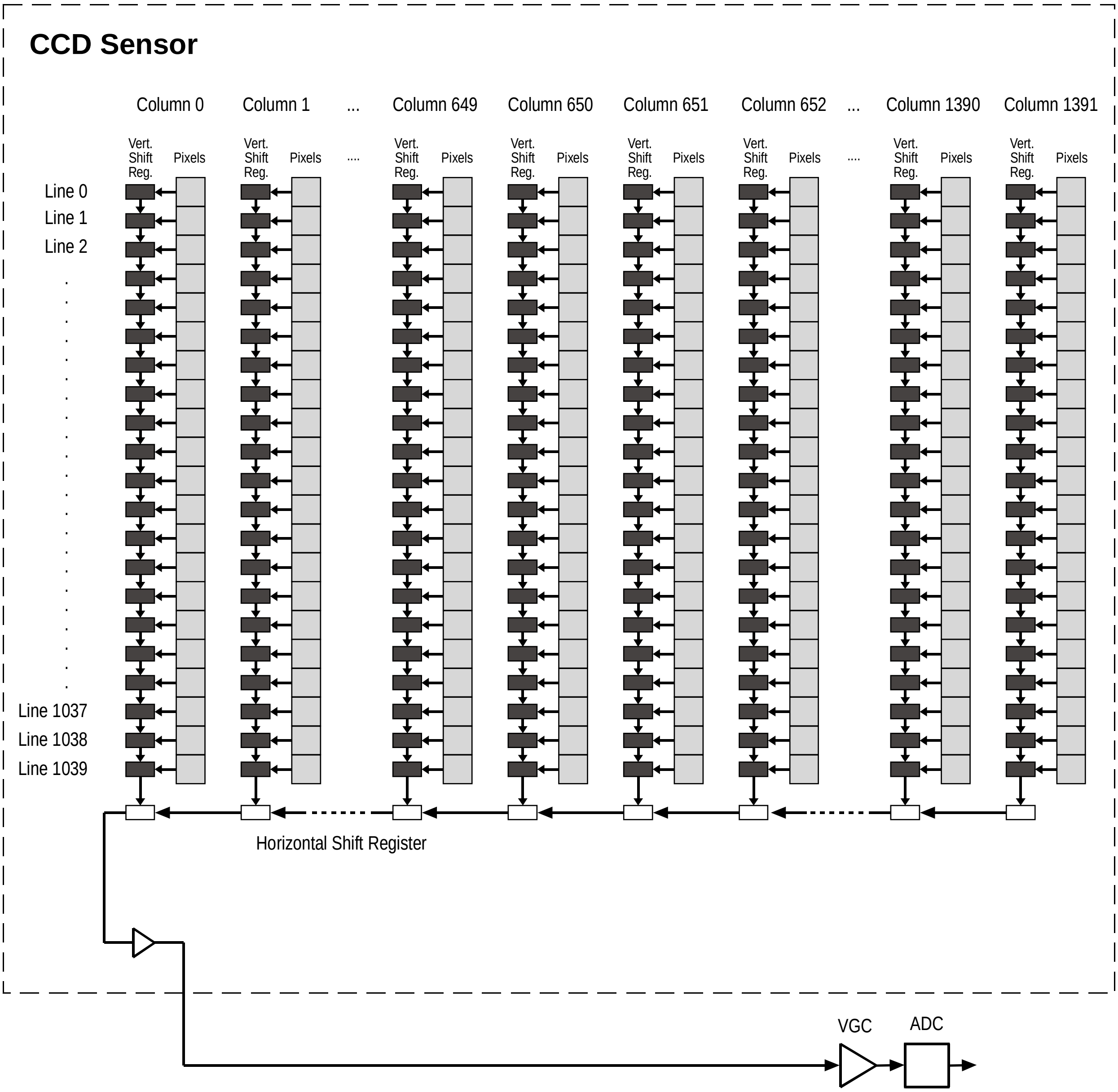


Figure 3-1: **A102f** Sensor Architecture

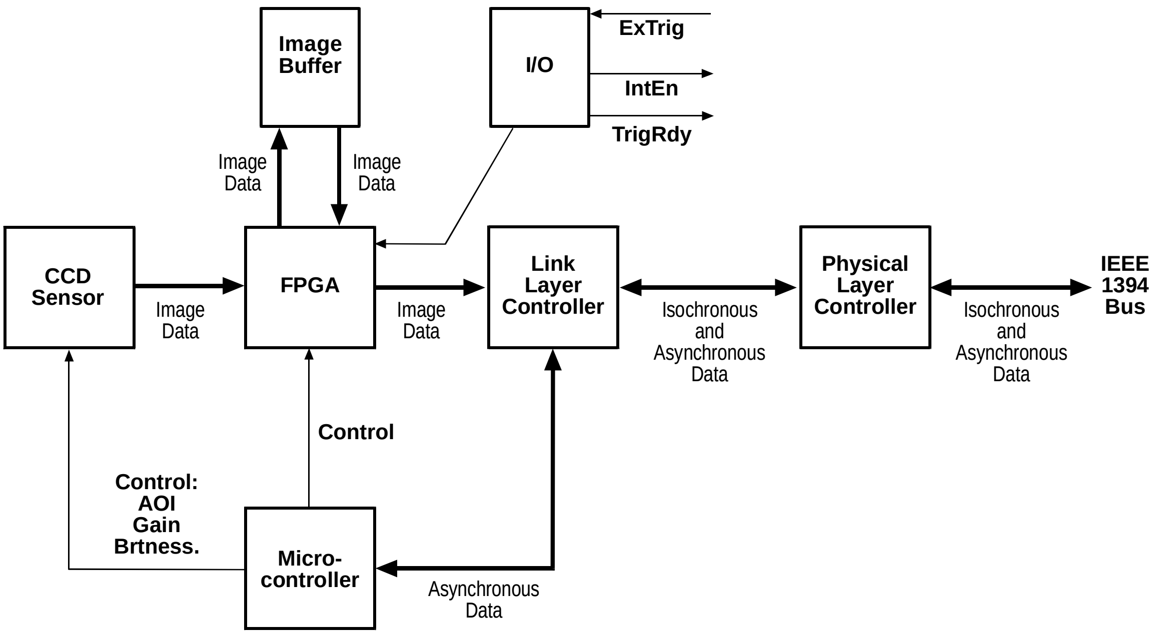


Figure 3-2: Block Diagram

3.2 Exposure Control

3.2.1 Setting the Exposure Time

Exposure time is determined by the value stored in the Shutter control register (see page 4-9). The value in the register can range from 1 to 4095 (0x001 to 0xFFF). The value in the register represents *n* in the equation: Exposure Time = *n* x 20 µs. So, for example, if the value stored in the Shutter register is 100 (0x064), the exposure time will be 100 x 20 µs or 2000 µs.

If you are operating the camera at a standard frame rate, you can determine the maximum shutter setting for that frame rate by reading the Max_Value field of the Shutter_Inq register (see page 4-6).

3.2.2 Maximum Exposure Time

The maximum exposure time for a given frame rate is determined by the following formula:

$$\frac{1}{\text{frame rate}} = \text{maximum exposure time}$$

For example, if a camera is operating at 15 fps:

$$\frac{1}{15\text{ fps}} = 0.0667\text{ s}$$

So in this case, the maximum exposure time is 66.7 ms.

ⓘ	Exceeding the maximum exposure time for your frame rate will cause the camera to slow down, i.e., it will cause the camera to operate at a lower frame rate.

3.2.3 Controlling Exposure Start with "Shot" Commands via the 1394 Interface

Exposure start can be controlled by sending "shot" commands directly to the camera via the 1394 bus. In this case, a software trigger or an external trigger (ExTrig) signal is not used. When exposure start is controlled via the 1394 bus, two modes of operation are available: one-shot and continuous-shot.

One-Shot Operation

In one-shot operation, the camera exposes and transmits a single image. Exposure begins after the One_Shot control register is set to 1 (see page 4-8). Exposure time is determined by the value field in the Shutter control register (see page 4-9).

The One_Shot control register is self cleared after transmission of the image data.

Continuous-Shot Operation

In continuous-shot operation, the camera continuously exposes and transmits images. The exposure of the first image begins after the Iso_En/Continuous_Shot control register is set to 1 (see page 4-8). The exposure time for each image is determined by the value field in the Shutter control register. The start of exposure on the second and subsequent images is automatically controlled by the camera.

If the camera is operating in video Format 0, the rate at which images will be captured and transmitted is determined by the value stored in the Cur_V_Frm_Rate / Revision control register (see page 4-8).

If the camera is operating in video Format 7, the rate at which images will be captured and transmitted is determined by the value stored in the Byte_Per_Packet control register (see Section 3.12.2 and page 4-11).

Image exposure and transmission stop after the Iso_En/Continuous_Shot control register is set to 0.

ⓘ	These explanations of exposure start are included to give the user a basic insight into the interactions of the camera's registers. Typically, IEEE 1394 cameras are used with a driver which includes an interface that allows the user to parameterize and operate the camera without directly setting registers. The Basler BCAM 1394 Camera Driver, for example, has both a simple Windows® interface and a programmer's API for parameterizing and operating the camera.

ⓘ	On **A102f** cameras, exposure of a new image can begin while the previous image is being read out. This is commonly referred to as "overlap mode." Following the recommended method for exposure start in Section 3.2.6 will allow you to overlap exposure with readout and achieve the camera's maximum frame rate.

3.2.4 Controlling Exposure Start with a Software Trigger

Exposure start can be controlled by sending a software trigger command to the camera via the 1394 bus. The Trigger_Mode control register (see page 4-9) is used to enable the ability to start image exposure with a software trigger. The Software_Trigger control register (see page 4-8) is used to set the software trigger.

If you are triggering the camera with a software trigger, only the programmable exposure mode is available. In programmable mode, exposure starts when the Software_Trigger control register is set to 1. The length of the exposure is determined by the value stored in the Shutter control register (see page 4-9). The Software_Trigger register will self clear shortly after exposure start. Figure 3-3 illustrates programmable exposure with a software trigger.

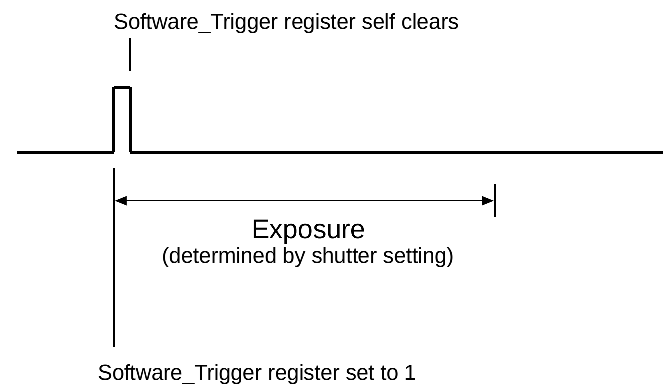


Figure 3-3: Programmable Exposure with a Software Trigger

Enabling the Software Trigger Feature

To enable the software trigger feature:

- Set the On_Off field of the Trigger_Mode control register to 1 to enable triggering.
- Set the Trigger_Source field of the Trigger_Mode control register to 7 to select software triggering.
- Set the Trigger_Mode field of the Trigger_Mode control register to 0 to select the programmable exposure mode.

Using the Software Trigger Feature

To use the software trigger feature, the camera must be set for continuous-shot operation. If more precise control of exposure start time is desired, you should also monitor the Trigger Ready signal and you must base the timing of the software trigger on the state of the Trigger Ready signal. (See Section 3.3 for more information on the Trigger Ready signal.)

The following descriptions assume that you are using a software trigger to start exposure and that you are monitoring the Trigger Ready signal.

Software Trigger / Continuous-Shot Operation

In Software Trigger/Continuous-shot operation, a "Continuous Shot Command" is used to prepare the camera to capture multiple images. In this mode, exposure will begin when the Software_Trigger register is set to 1. To use this method of operation, follow this sequence:

1. Set the Shutter control register for your desired exposure time (see page 4-9).
2. Set the Iso_En/Continuous_Shot control register to 1.
3. Check the state of the TrigRdy signal:
   a) If TrigRdy is high, you can set the Software_Trigger register to 1 when desired.
   b) If TrigRdy is low, wait until TrigRdy goes high and then set the Software_Trigger register to 1 when desired.
4. When the Software_Trigger register is set to 1, exposure will begin.
5. The Software_Trigger register will self clear shortly after exposure begins. Exposure will continue for the length of time specified in the Shutter control register.
6. At the end of the specified exposure time, readout and transmission of the captured image will take place.
7. Repeat steps 3 and 4 each time that you want to begin exposure and capture an image.
8. Image exposure and transmission will stop when the Iso_En/Continuous_Shot control register is set to 0.

> The Software_Trigger register and the Trigger_Source field in the Trigger_Mode control register are defined in version 1.31 of the IIDC specification.
>
> Because the software trigger feature is so new, the Basler BCAM 1394 Driver does not yet include a method call to access this feature. As a work-around, you can use the techniques described in the tutorial for the Basler Smart Features Framework Software to directly read from or write to the Software_Trigger and Trigger_mode registers. The Smart Features Framework Software is available on the Basler web site. To download the software go to:
>
> http://www.baslerweb.com/popups/popup_en_1825.php

Why Use the Software Trigger?

At first glance, using the software trigger feature to start image exposure appears to be equivalent to just issuing a one-shot command as described in Section 3.2.3. The difference is in the way that the camera reacts to each method. With a one-shot command, there will be some delay between the One_Shot control register being set to 1 and the actual start of exposure time. This delay is required so that the camera can be properly set up to react to the receipt of the one-shot command. With the software trigger method, there is no delay between the Software_Trigger register being set to 1 and the start of exposure. Exposure begins immediately when the register is set. So the advantage of the software trigger feature is that it gives you more precise control of the start of exposure.

> On **A102f** cameras, exposure of a new image can begin while the previous image is being read out. This is commonly referred to as "overlap mode." Following the recommended method for exposure start in Section 3.2.6 will allow you to overlap exposure with readout and achieve the camera's maximum frame rate.

3.2.5 Controlling Exposure Start with an ExTrig Signal

An external trigger (ExTrig) input signal can be used to control the start of exposure. A rising edge or a falling edge of the signal can be used to trigger exposure start. The Trigger_Mode control register (see page 4-9) is used to enable ExTrig exposure start control, to select rising or falling edge triggering, and to assign a physical input port to receive the ExTrig signal.

The ExTrig signal can be periodic or non-periodic. When the camera is operating under control of an ExTrig signal, the period of the ExTrig signal determines the camera's frame rate:

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

For example, if you are operating a camera with an ExTrig signal period of 110 ms (0.110 sec.):

$$\frac{1}{0.110} = 9.1 \text{ fps}$$

So in this case, the frame rate is 9.1 fps.

The minimum high time for a rising edge trigger (or low time for a falling edge trigger) is 1 µs.

Exposure Modes

If you are triggering the camera with an ExTrig signal, two exposure modes are available, programmable mode and level controlled mode.

Programmable Exposure Mode

When programmable mode is selected, the length of the exposure is determined by the value stored in the Shutter control register (see page 4-9). If the camera is set for rising edge triggering, exposure starts when the ExTrig signal rises. If the camera is set for falling edge triggering, exposure starts when the ExTrig signal falls. Figure 3-4 illustrates programmable exposure with the camera set for rising edge triggering.

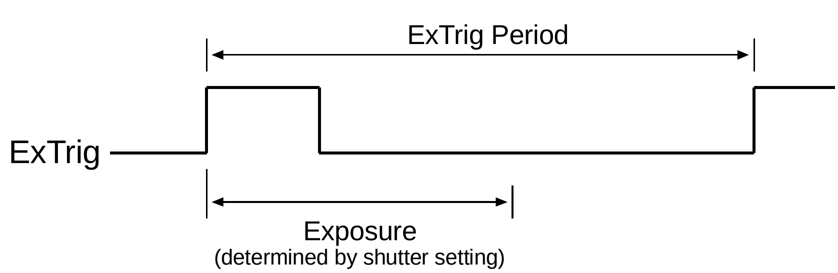


Figure 3-4: Programmable Exposure with Rising Edge Triggering

Level Controlled Exposure Mode

When level controlled mode is selected, the length of the exposure will be determined by the ExTrig signal alone. If the camera is set for rising edge triggering, exposure begins when the ExTrig signal rises and continues until the ExTrig signal falls. If the camera is set for falling edge triggering, exposure begins when the ExTrig signal falls and continues until the ExTrig signal rises. Figure 3-5 illustrates level controlled exposure with the camera set for rising edge triggering.

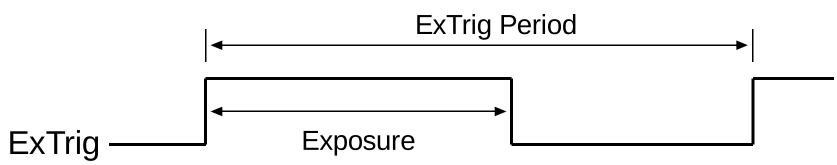


Figure 3-5: Level Controlled Exposure with Rising Edge Triggering

Enabling the External Trigger Feature

To enable the external trigger feature:

- Set the On_Off field of the Trigger_Mode control register to 1 to enable triggering.
- Set the Trigger_Polarity field of the Trigger_Mode control register to 0 to select falling edge triggering or 1 to select rising edge triggering.
- Set the Trigger_Mode field of the Trigger_Mode control register to 0 to select the programmable exposure mode or 1 to select the level controlled exposure mode.
- Set the Trigger_Source field in the Trigger_Mode control register to select which one of the four physical input ports on the camera will be used to receive the external trigger signal:

  - Set the Trigger_Source field to 0 to select physical input port 0 to receive the ExTrig signal.

  - Set the Trigger_Source field to 1 to select physical input port 1 to receive the ExTrig signal.

  - Set the Trigger_Source field to 2 to select physical input port 2 to receive the ExTrig signal.

  - Set the Trigger_Source field to 3 to select physical input port 3 to receive the ExTrig signal.

The default setting is for physical input port 0 to receive the ExTrig signal. Refer to Sections 2.1.2 and 2.4.1 for a description of the physical and electrical characteristics of the physical input ports.

ⓘ	The Trigger_Source field in the Trigger_Mode register is defined in version 1.31 of the IIDC specification. Because the Trigger_Source feature is so new, the Basler BCAM 1394 Driver does not yet include a method call to access this feature. As a work-around, you can use the techniques described in the tutorial for the Basler Smart Features Framework Software to directly read from or write to the Trigger_Source field. The Smart Features Framework Software is available on the Basler web site. To download the software go to: http://www.baslerweb.com/popups/popup_en_1825.php

The ExTrig signal must be used in combination with a one-shot or a continuous-shot command. If precise control of exposure start time is desired, you must also monitor the Trigger Ready signal and you must base the timing of the ExTrig signal on the state of the Trigger Ready signal. (See Section 3.2.6 for recommended methods for using the signal)

The following descriptions assume that the ExTrig signal is set for rising edge triggering and the programmable exposure mode.

ExTrig / One-Shot Operation

In ExTrig/One-shot operation, a “One-shot” Command is used to prepare the camera to capture a single image. When the ExTrig signal rises, exposure will begin. To use this method of operation, follow this sequence:

1. Set the Shutter control register for your desired exposure time (see Section 3.2.1).
2. Set the One_Shot control register to 1.
3. Check the state of the TrigRdy signal:
   a) If TrigRdy is high, you can toggle ExTrig when desired.
   b) If TrigRdy is low, wait until TrigRdy goes high and then toggle ExTrig when desired. (See Section 3.3 for more about TrigRdy.)
4. When ExTrig rises, exposure will begin. Exposure will continue for the length of time specified in the Shutter control register.
5. At the end of the specified exposure time, readout and transmission of the captured image will take place.

The One_Shot control register is self cleared after image transmission.

ExTrig / Continuous-Shot Operation

In ExTrig/Continuous-shot operation, a “Continuous-shot” command is used to prepare the camera to capture multiple images. In this mode, exposure will begin on each rising edge of the ExTrig signal. To use this method of operation, follow this sequence:

1. Set the Shutter control register for your desired exposure time (see Section 3.2.1).
2. Set the Iso_En/Continuous_Shot control register to 1.
3. Check the state of the TrigRdy signal:
   a) If TrigRdy is high, you can toggle ExTrig when desired.
   b) If TrigRdy is low, wait until TrigRdy goes high and then toggle ExTrig when desired. (See Section 3.3 for more about TrigRdy.)
4. When ExTrig rises, exposure will begin. Exposure will continue for the length of time specified in the Shutter control register.
5. At the end of the specified exposure time, readout and transmission of the captured image will take place.
6. Repeat steps 3 and 4 each time that you want to begin exposure and capture an image.
7. Image exposure and transmission stop when the Iso_En/Continuous_Shot control register is set to 0.

ⓘ	These explanations of exposure start are included to give the user a basic insight into the interactions of the camera's registers. Typically, IEEE 1394 cameras are used with a driver which includes an interface that allows the user to parameterize and operate the camera without directly setting registers. The Basler BCAM 1394 Camera Driver, for example, has both a simple Windows® interface and a programmer's API for parameterizing and operating the camera.

ⓘ	On **A102f** cameras, exposure of a new image can begin while the previous image is being read out. This is commonly referred to as "overlap mode." Following the recommended method for exposure start in Section 3.2.6 will allow you to overlap exposure with readout and achieve the camera's maximum frame rate.

3.2.6 Recommended Method for Controlling Exposure Start

The camera can be programmed to begin exposure on a rising edge or on a falling edge of an ExTrig signal. Also, two modes of exposure control are available: programmable and level controlled (see Section 3.2.5). For this illustration, we are assuming that a rising edge trigger and the programmable exposure mode are used.

If a camera user requires close control of exposure start, there are several general guidelines that must be followed:

- The camera should be placed in continuous shot mode.
- The user must use an external trigger (ExTrig) signal to start exposure.
- The user must monitor the trigger ready (TrigRdy) signal.
- A rising edge of the ExTrig signal must only occur when the TrigRdy signal is high.

Assuming that these general guidelines are followed, the reaction of the camera to a rising external trigger signal will be as shown in Figure 3-6:

- The start of exposure will typically occur between 0 µs and 63 µs after the rise of the ExTrig signal.
- The integrate enabled (IntEn) signal will rise between 5 and 20 µs after the start of exposure.
- The actual length of exposure will be equal to the programmed exposure time.
- The IntEn signal will fall between 30 and 100 µs after the end of exposure.

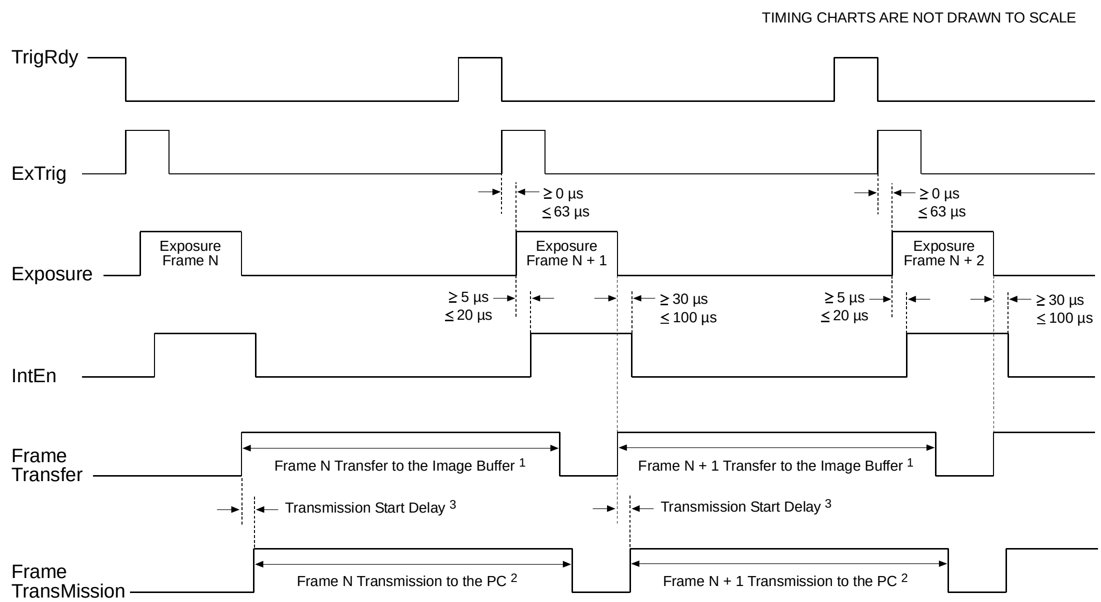


Figure 3-6: Exposure Start Controlled with an ExTrig Signal

[1] Frame Transfer Time = (AOI Height x 51.2281 µs) + 9941.0 µs

[2] Frame Transmission Time = Packets/frame x 125 µs

[3] If the transmission time is greater than the transfer time:

Start Delay = 125 µs.

If the transmission time is less than the transfer time:

Start Delay = (Transfer Time - Transmission Time) + 125 µs

3.3 Trigger Ready Signal

ⓘ	The trigger ready signal is not defined in the 1394 Trade Association Digital Camera Specification. Trigger ready is a patented feature of Basler cameras that allows our cameras to have optimized timings.

The maximum frame rate for the camera can be limited by any one of three factors:

- The amount of time it takes to transfer a captured image from the CCD sensor to the frame buffer.
- The amount of time it takes to transfer an image from the frame buffer to the PC via the IEEE 1394 bus.
- The exposure time setting.

The camera automatically recalculates the maximum frame rate any time a setting that effects one or more of these factors is changed. For example, the camera will recalculate the maximum frame rate if you change the exposure time, the size of the area of interest, or the packet size.

The camera will use the calculated maximum frame rate to generate a “trigger ready” (TrigRdy) signal. The trigger ready signal indicates the earliest moment that each exposure can begin without exceeding the maximum frame rate for the current conditions. The trigger ready signal will go low when each exposure is started and will go high when it is safe for the next exposure to begin (see Figure 3-6).

By default, the TrigRdy signal is assigned to physical output port 1 on the camera. See Section 2.4.2 for a description of the electrical characteristics of the camera’s physical output ports.

The assignment of the TrigRdy signal to a physical output port can be changed. See Section 5.7.11 for more information on changing the assignment of camera output signals to physical output ports.

ⓘ	If you signal the camera to start an exposure when trigger ready is low, the camera will delay the start of exposure until the next rise of the trigger ready signal. This prevents you from running the camera faster than the maximum rate and avoids dropping frames. If the camera is in continuous shot mode and external triggering is disabled, the trigger ready output signal will not be present.

3.4 Integrate Enabled Signal

The Integrate Enabled (IntEn) signal goes high when exposure begins and goes low when exposure ends. 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 IntEn signal to know when exposure is taking place and thus know when to avoid moving the camera.

By default, the IntEn signal is assigned to physical output port 0 on the camera. See Section 2.4.2 for a description of the electrical characteristics of the camera's physical output ports.

The assignment of the IntEn signal to a physical output port can be changed. See Section 5.7.11 for more information on changing the assignment of camera output signals to physical output ports.

ⓘ	When you use the integrate enabled signal, be aware that there is a delay in the rise and the fall of the signal in relation to the start and the end of exposure. See Figure 3-6 for details.

3.5 Gain and Brightness

The major components in the **A102f** electronics include: a CCD sensor, one VGC (Variable Gain Control), and one ADC (Analog to Digital Converter). The pixels in the CCD sensor output voltage signals when they are exposed to light. These voltages are amplified by the VGC and transferred to the ADC which converts the voltages to digital output signals.

Two parameters, gain and offset are associated with the VGC. As shown in Figures 3-7 and 3-8, increasing or decreasing the gain increases or decreases the amplitude of the signal that is input to the ADC. Increasing or decreasing the offset moves the signal up or down the measurement scale but does not change the signal amplitude.

For most applications, black should have a gray value of 1 and white should have a gray value of 255 (in modes that output 8 bits per pixel) or 4095 (in modes that output 12 effective bits per pixel). Attempt to achieve this by varying exposure and illumination rather than changing the camera's gain. The default gain is the optimal operating point (minimum noise) and should be used if possible.

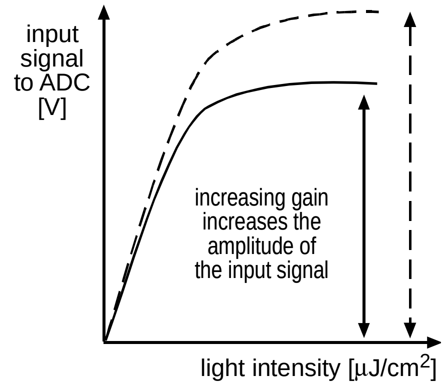


Figure 3-7: Gain

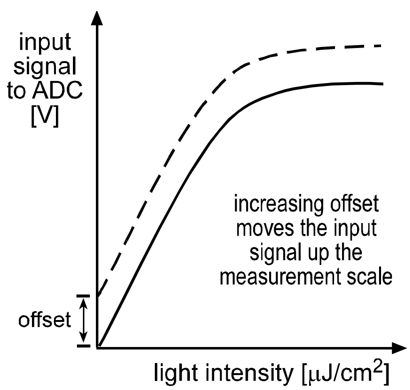


Figure 3-8: Offset

ⓘ	Because increasing gain increases both signal and noise, the signal to noise ratio does not change significantly when gain is increased.

3.5.1 Setting Gain

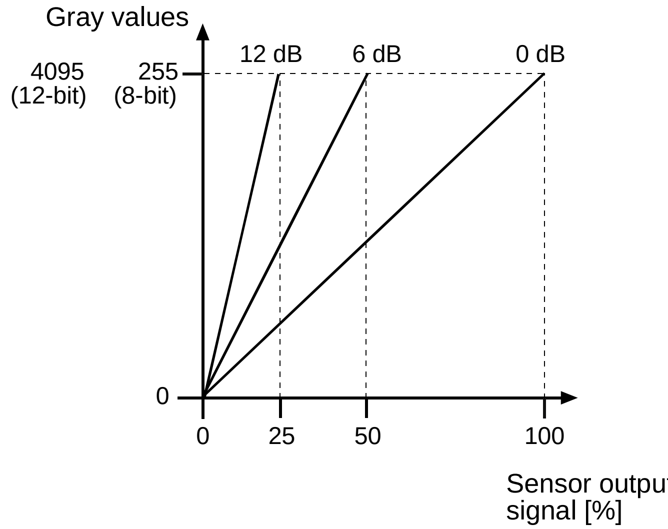


Figure 3-9: Gain Settings in dB

When the gain is set to default, the sensor's linear output range directly matches the input voltage range of the ADC. Thus, with the default gain of 0 dB, a gray value of 1 is produced when the pixels are exposed to no light and a gray value of 255 (in modes that output 8 bits per pixel) or 4095 (in modes that output 12 effective bits per pixel) is produced when the pixels are exposed to bright light.

0 dB of gain is achieved when gain is programmed to a decimal value of 192. Increasing the gain setting to more than 192 maps a smaller portion of the sensor's linear output range to the ADC's input.

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 effective bits per pixel) is reached. For example, if you found that at your brightest exposure your gray values were no higher than 127 (8-bit mode), you could increase the gain to 6 dB (amplification factor of 2) and thus reach gray values of 254 (see Figure 3-9).

Gain is adjustable and can be programmed on a decimal scale that ranges from 192 to 1023 (0x0C0 to 0x3FF). The camera's gain setting is determined by the value field in the Gain control register (see 4-9).

If you know the decimal number (DN) setting for the gain on your camera, the equivalent decibel value can be calculated using one of the following equations:

When DN setting = 192 to 511

$$dB = 20 \times \log_{10}\left(\frac{658 + DN}{658 - DN}\right) - 5.221$$

When DN setting = 512 to 1023

$$dB = 0.0354 \times DN - 5.221$$

ⓘ	When the camera is set for Mono 16 output, only settings from 192 to 511 are valid. Settings above 511 should not be used with the camera set for Mono 16 output. In all other output modes, the entire 192 to 1023 range of gain settings is valid. In normal operation, a gain setting lower than 192 should not be used. When the gain setting is lower than 192, the sensor output signal that is mapped to the input of the ADC will not be linear.

The gain settings result in the following amplifications:

Decimal Number (DN)	Hexadecimal	dB	Factor
192	0x0C0	0	X1
374	0x176	6	X2
499	0x1F3	12	X4
656	0x290	18	X8
825	0x339	24	X16
1023	0x3FF	31	X35.5

Table 3-1: Examples of Gain Settings in dB

3.5.2 Setting Brightness

The camera's brightness is changed by setting the value field in the Brightness control register (see page 4-8). The brightness setting can be programmed on a decimal scale that ranges from 0 to 255, (0x000 to 0x0FF).

If the camera is operating in any output mode other than Mono 16, a brightness setting of around 8 (decimal) will result in an offset of 0 in the digital values output for the pixels. (Due to tolerances in the electronic components in your camera, you may find that the default brightness is set to a slightly different setting.) An increase of 16 (decimal) in the brightness setting will result in a positive offset of 1 in the digital values output for the pixels. For example, a brightness setting of around 24 (8 + 16, decimal) would be required to reach a positive offset of 1. A brightness setting of around 40 (8 + 16 + 16, decimal) would be required to reach a positive offset of 2, and so on.

If the camera is set for Mono 16 output, a brightness setting of around 0 (decimal) will result in an offset of 0 in the digital values output for the pixels. An increase of 1 (decimal) in the brightness setting will result in a positive offset of 1 in the digital values output for the pixels.

3.6 Area of Interest (AOI)

The area of interest (AOI) feature allows you to specify a portion of the CCD array and during operation, only the pixel information from the specified portion of the array is transmitted out of the camera.

The area of interest is referenced to the top left corner of the CCD array. The top left corner is designated as column 0 and row 0 as shown in Figure 3-10.

The location and size of the area of interest is defined by declaring a left-most column, a width, a top row and a height. For example, suppose that you specify the left column as 10, the width as 16, the top row as 4 and the height as 10. The area of the array that is bounded by these settings is shown in Figure 3-10.

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

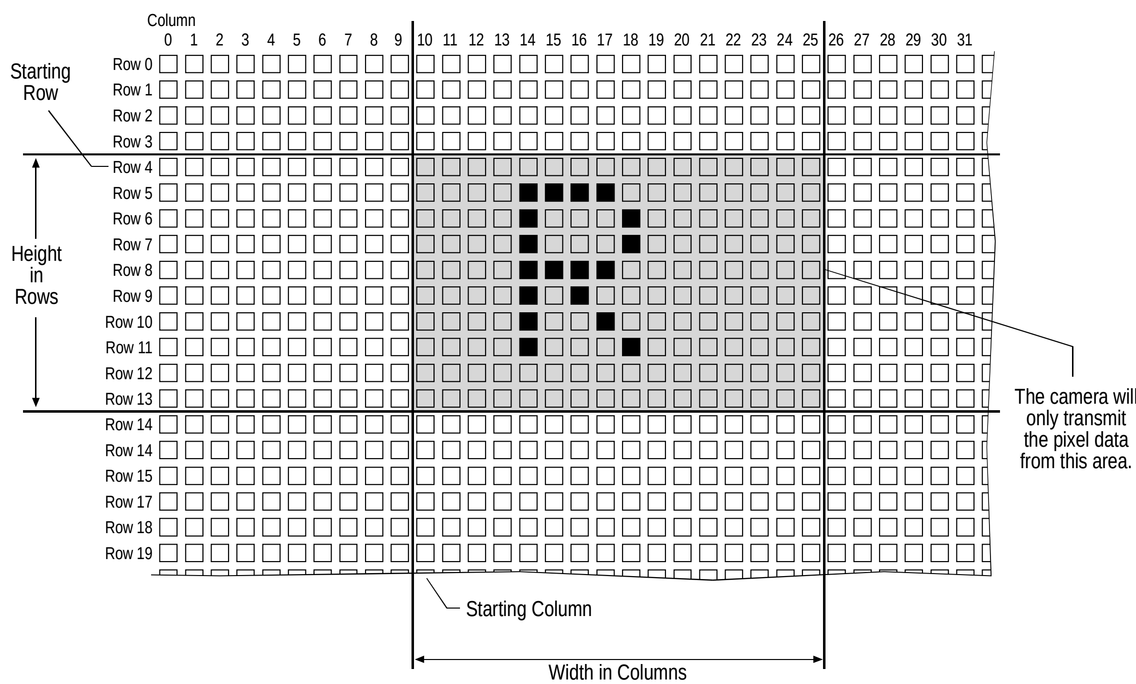


Figure 3-10: Area of Interest

The AOI feature is enabled by setting the camera to operate in Format_7, Mode_0. This is accomplished by setting the Cur_V_Format control register (see page 4-8) to 7 and the Cur_V_Mode control register to 0. The location of the area of interest is defined by setting a value for the “left” field and a value for the “top” field within the Image_Position control register for Format_7, Mode_0 (see page 4-10). The size of the area of interest is defined by setting a value for the “width” field and a value for the “height” field within the Image_Size control register for Format_7, Mode_0.

To use the entire CCD array in the **A102f** monochrome cameras, set the value for “left” to 0, the value for “top” to 0, the value for “width” to 1392 and the value for “height” to 1040.

To use the entire CCD array in the **A102fc** color cameras, set the value for “left” to 0, the value for “top” to 0, the value for “width” to 1388 and the value for “height” to 1038.

The sum of the setting for *Left* plus the setting for *Width* must not exceed 1392 for **A102f** monochrome cameras or 1388 for **A102fc** color cameras.

The sum of the setting for *Top* plus the setting for *Height* must not exceed 1040 for **A102f** monochrome cameras or 1038 for **A102fc** color cameras.

When you are setting the AOI on an **A102fc** color camera:

- The setting for *Width* must be divisible by 2.
- The setting for *Height* must be divisible by 2.
- The setting for *Left* must be zero or be divisible by 2.
- The setting for *Top* must be zero or be divisible by 2.

3.6.1 Changing AOI Parameters “On-the-Fly”

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

The camera’s response to an on-the-fly change in the AOI position will vary depending on the way that you are operating the camera:

- If the exposure time is ≥ 100 µs, the changes will take effect on the next trigger after the changes are received by the camera.
- If the exposure time is < 100 µs and the camera is running in non-overlapped mode[1], the changes will take effect on the next trigger after the changes are received by the camera.
- If the exposure time is < 100 µs and the camera is running in overlapped mode[2], when the changes are received by the camera, the camera will delay the triggering of the next image until transmission of the current image is complete. When transmission of the current image is complete, the camera will change the AOI position, will trigger the next image, and will resume running in overlapped mode.

[1] The term “non-overlapped” mode means that image capture is triggered in the following manner: the camera captures (exposes) an image and completely transmits that image out of the camera before the next image capture is triggered. In other words, exposure and transmission of image N are both completed before exposure of image N+1 begins.

[2] The term “overlapped” mode means that image capture is triggered in the following manner: the camera captures (exposes) an image and while this image is being transmitted out of the camera, capture of the next image is triggered. In other words, capture of image N+1 begins while transmission of image N is still in progress.

3.6.2 Changes to the Frame Rate With AOI

In general, the maximum frame rate for the camera increases as the size of the AOI decreases. However, the maximum frame rate can also be limited by any one of three factors:

- The amount of time it takes to transfer a captured image from the image sensor to the frame buffer.
- The amount of time it takes to transfer an image from the frame buffer to the PC via the IEEE 1394 bus.
- The exposure time setting.

To determine the maximum frame rate for a given AOI, use your AOI and exposure time settings to calculate a result in each of the three formulas below. These formulas take your AOI size into account plus the other factors that can limit the frame rate. The formula that returns the lowest value will determine the maximum frame rate for the given AOI.

Formula 1:

$$\text{Max Frames/s} = \frac{1}{(\text{AOI Height} \times 51.2281\ \mu\text{s}) + 9941.0\ \mu\text{s}}$$

Formula 2:

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

Formula 3:

$$\text{Max. Frames/s} = \frac{1}{\text{Exposure Time in }\mu\text{s} + 181.1\ \mu\text{s}}$$

Example

Assume that your AOI is set for 100 columns wide and 110 rows high and that your exposure time is set for 12000 µs. Also assume that after making all camera settings, you check the Packer_Per_Frame_Inq register in the control and status registers for Format 7. You find that the packets per frame with the current settings is 3.

ⓘ	The number of packets per frame depends on the setting of the Format 7 Byte_Per_Packet control register. In this example, we assume that the bytes per packet is set to the maximum. See Sections 3.12.2 and 3.13.2 for more information.

Formula 1:

$$\text{Max Frames/s} = \frac{1}{(110 \times 51.2281\ \mu s) + 9941.0\ \mu s}$$

Max. Frames/s = 64.2

Formula 2:

$$\text{Max. Frames/s} = \frac{1}{3 \times 125\ \mu s}$$

Max. Frames/s = 2666.7

Formula 3:

$$\text{Max. Frames/s} = \frac{1}{12000\ \mu s + 181.1\ \mu s}$$

Max. Frames/s = 82.1

Formula one returns the lowest value. So in this case, the limiting parameter is the frame transfer time and the maximum frame rate would be 64.2 frames per second.

3.7 Low Smear

In applications where a CCD sensor is under constant illumination, high-contrast images may show an unwanted effect that converts dark pixels into brighter ones. This effect is commonly called “smearing“.

With the help of the Low Smear feature on the **A102k**, smearing is reduced in the upper part of the image. The effect of the Low Smear feature is illustrated in Figure 3-11.

The left image was captured without the low smear feature. There is smearing both in the upper and lower part of the image.

The right image was captured with low smear active. There is no smearing in the upper part of the image.

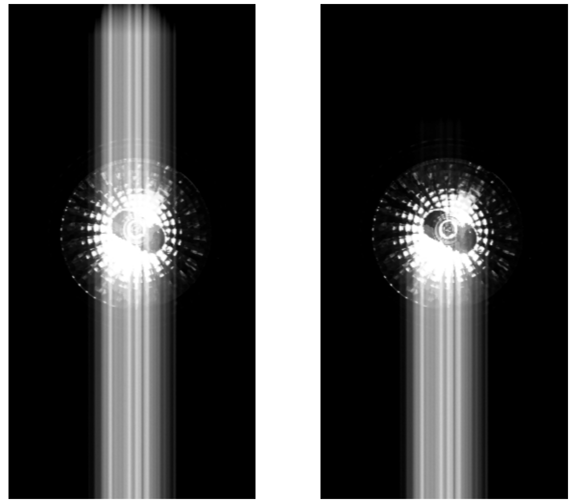


Figure 3-11: Full Smear (left), Low Smear (right)

Smearing is caused by two things:

- An unwanted post-exposure of the pixels when they are being moved out through the vertical shift registers. Only those pixels located above the area of exposure on the CCD array which must pass the light source during shift-out are subject to post-exposure. For this reason, post-exposure only produces smearing in the lower part of the image. (Remember that the lens causes the image on the sensor to be inverted, so the lower part of the image is at the top of the sensor.)
- An unwanted existing accumulation of charges in those shift registers which have passed points of constant illumination during the previous frame transfer and have thus been exposed before they receive the next pixels. These unwanted charges add to the next pixels when these pixels are shifted from the sensor cells into the vertical shift registers. This causes smearing in the upper part of the image.

The amount of unwanted charges accumulated in the shift registers grows with the amount of exposure. For that reason, smearing does not appear under short-term illumination such as flash light. It only appears under constant illumination.

The Low Smear feature cannot be activated or deactivated. It is active all of the time. To use this feature to its best advantage, the frame rate must not exceed a maximum setting. The setting can be calculated using the below equations.

First, you need to calculate the frame transfer time based on the height of the area of interest (AOI) using this formula:

$$T(f) = (AOIH \times 51.2281\ \mu s) + 9941.0\ \mu s$$

where: T(f) = frame transfer time

AOIH = number of lines in the AOI

Second, you need to calculate the maximum recommended frame rate for low smear using this formula:

$$\text{Frames/s} \le \frac{1}{T(f) + \left[\left(\text{AOIH} - \frac{T(e)}{63.16}\right) \times 5.24\right] + T(e)}$$

where: T(f) = frame transfer time in µs
AOIH = number of lines in the AOI
T(e) = exposure time in µs

If you use an example with a 2000 µs exposure time and a 1392 (H) x 600 (V) area of interest, the calculations look like this:

T(f) = (600 x 51.2281 µs) + 9941.0 µs

T(f) = 40677.86 µs

and:

$$\text{Frames/sec.} \le \frac{1}{40677.86\ \mu s + \left[\left(600 - \frac{2000}{63.16}\right) \times 5.24\ \mu s\right] + 2000\ \mu s}$$

Frames/sec. ≤ 21.90

If the camera's actual frame rate is higher than the maximum recommended frame rate, the smearing will come back. When you exceed the maximum recommended frame rate by a small amount, the upper part of the image will show partial smearing (Figure 3-12). As the frame rate is increased, the smearing will become worse.

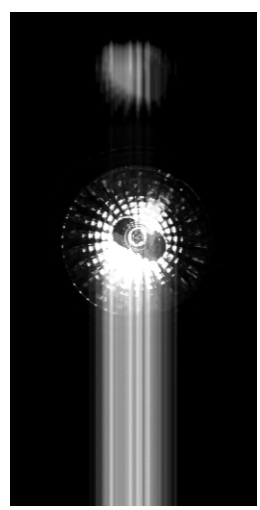


Figure 3-12: Partial Smear

3.8 Color Creation in the A102fc

The CCD sensor used in the **A102fc** is equipped with an additive color separation filter known as a Bayer filter. With the Bayer filter, each individual pixel is covered by a micro-lens which allows light of only one color to strike the pixel. The pattern of the Bayer filter used in the **A102fc** is shown in Figure 3-13. As the figure illustrates, within each block 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.)

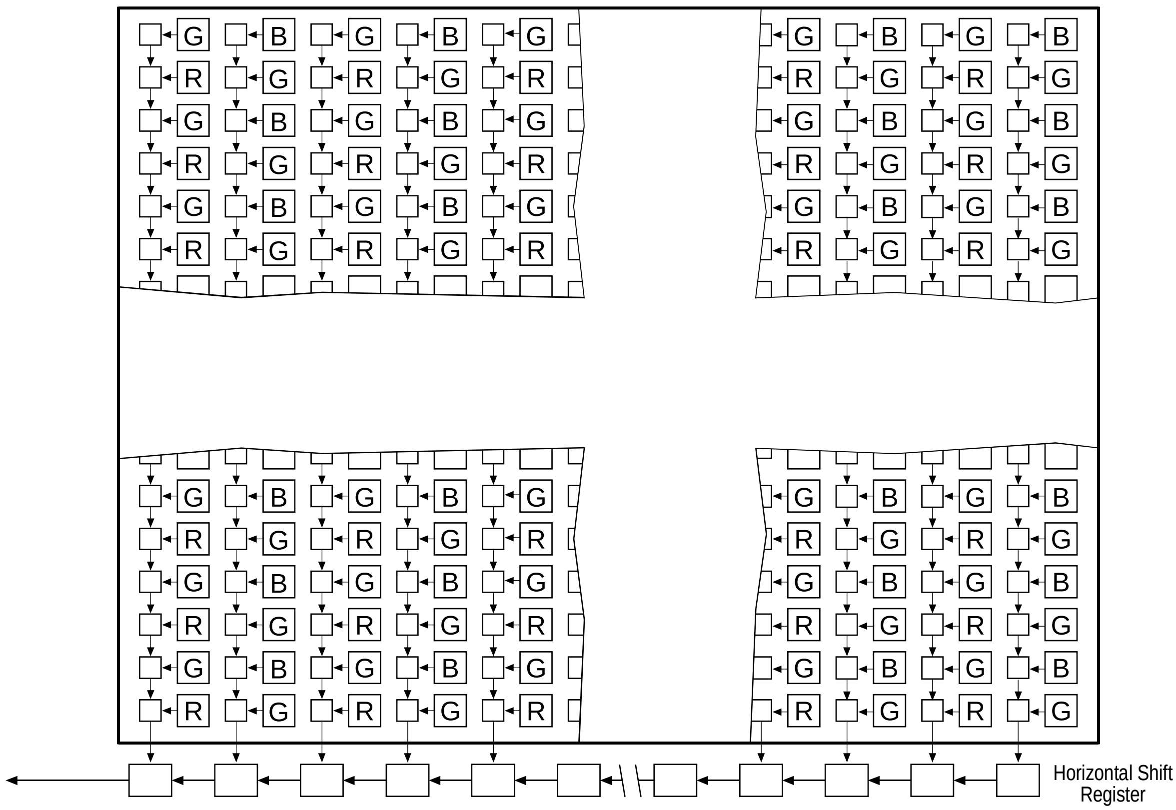


Figure 3-13: Bayer Filter Pattern on the **A102fc**

When an **A102fc** is operating in a YUV color output mode, each pixel goes through a two step conversion process as it exits the sensor and passes through the camera's electronics.

In the first step of the process an interpolation algorithm is performed to get full RGB data for the pixel. (Because each individual pixel gathers information for only one color, an interpolation must be made from the surrounding pixels to get full RGB data for an individual pixel.)

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

$Y = 0.257\ R + 0.504\ G + 0.098\ B + 16$

$U = -0.148\ R - 0.291\ G + 0.439\ B + 128$

$V = 0.439\ R - 0.368\ G - 0.071\ B + 128$

Once the conversion to YUV is complete, pixels are transmitted from the camera in the YUV (4:2:2) format as defined in Section 2.1.3 *Video Data Payload Structure* in the 1394-based Digital Camera Specification Version 1.20.

ⓘ	The values for U and for V normally range from -128 to +127. Because the 1394 Digital Camera specification requires that U values and V values be transmitted with unsigned integers, 128 is added to each U value and to each V value before the values are transmitted from the camera. This process allows the values to be transmitted on a scale that ranges from 0 to 255 (for more information, see Section 2.1.4 *Data Structure* in the 1394-based Digital Camera Specification Version 1.20). When an **A102fc** is operating in a YUV (4:2:2) mode, the average number of bits per pixel is 16. This means that the camera will require twice the bandwidth of a camera operating in an 8 bit monochrome mode. The **A102fc** can operate in YUV (4:2:2) color mode as well as several monochrome 8 bit modes (see Section 3.13).

3.8.1 White Balance

White balance capability has been implemented on the **A102fc**. With white balancing, a correction factor can be applied to the U values and to the V values transmitted from the camera.

The U_Value_or_B_Value field of the White_Balance control register (see page 4-8) can be used to vary the amount of blue in the captured images. The setting for the field can range from 16 (0x10) to 255 (0xFF). If the field is set to 64 (0x40), it will have no effect on the image. If the field is set to a value lower than 64, the image will be less blue. If the field is set to a value higher than 64, the image will be more blue. The default setting for the U_Value_or_B_Value is 112 (0x70).

To determine the effect of this field on the captured images, use this formula:

$$\text{Blue Gain} = \frac{\text{U\_Value\_or\_B\_Value field Setting}}{64}$$

The V_Value_or_R_Value field of the White_Balance control register can be used to vary the amount of red in the captured images. The setting for the field can range from 16 (0x10) to 255 (0xFF). If the field is set to 64 (0x40), it will have no effect on the image. If the field is set to a value lower than 64, the image will be less red. If the field is set to a value higher than 64, the image will be more red. The default setting for the V_Value_or_R_Value is 64 (0x40).

To determine the effect of this field on the captured images, use this formula:

$$\text{Red Gain} = \frac{\text{V\_Value\_or\_R\_Value field Setting}}{64}$$

3.8.2 Color Filter ID

The Color_Filter_ID field of the Format 7 register (see page 4-10) has been implemented on **A102f** cameras. This field can be used to determine the order of the pixel colors being output from a color camera. To use this feature, make all desired changes to the camera's parameter settings and then read the Color_Filter_ID field. The field will return the appropriate filter ID for the current settings.

The Bayer filter in the **A102fc** is an RGB primary color filter and the color filter ID field will return an ID of 0, 1, 2, or 3 as appropriate. Each ID identifies the color of the first two pixels in the first row transmitted by the camera and the color of the first two pixels in the second row transmitted. Due to the repetitive nature of the color filter, this information is all that you need to determine the order of the pixel colors in your current AOI. Table 3-2 shows the meaning of each of the color IDs.

	ID 0	ID 1	ID 2	ID 3
First pixel in the first row	R	G	G	B
Second pixel in the first row	G	B	R	G
First pixel in the second row	G	R	B	G
Second pixel in the second row	B	G	G	R

Table 3-2: Color Filter IDs

The Color_Filter_ID field of the Format 7 register is defined in version 1.31 of the IIDC specification.

Because the Color_Filter_ID feature is so new, the Basler BCAM 1394 Driver does not yet include a method call to access this feature. As a work-around, you can use the techniques described in the tutorial for the Basler Smart Features Framework Software to directly read from or write to the Color_Filter_ID field. The Smart Features Framework Software is available on the Basler web site. To download the software go to:

http://www.baslerweb.com/popups/popup_en_1825.php

3.8.3 Integrated IR Cut Filter on C-Mount Equipped Cameras

A102fc color cameras are equipped with an IR cut filter as standard equipment. The filter is mounted in the lens adapter. Cameras without an IR cut filter are available on request.

Caution!

The location of the filter limits the thread length of the lens that can be used on the camera. The thread length on your lens must be less than 7.5 mm. If a lens with a longer thread length is used, the camera will be damaged and will no longer operate. See Section 6.3 for more details.

3.9 Selectable 8 or 12 Bit Pixel Depth

When an **A102f** camera is operating in Format 7, it can be set to output pixel data at either 8 bit or 12 bit depth.

3.9.1 A102f Monochrome Cameras

For 8 Bit Depth

Set the Color_Coding_ID field of the Format_7, Mode_0 register to Mono 8 (see Section 3.12.2 and page 4-10). With this ID set, the camera outputs 8 bits per pixel.

For 12 Bit Depth

Set the Color_Coding_ID field of the Format_7, Mode_0 register for Mono 16. With this ID set, the camera outputs 16 bits per pixel but only 12 bits are effective. The effective pixel data fills from the LSB and the unused bits are filled with zeros. Pixel data is stored in the PC memory in little endian format, i.e., the low byte for each pixel is stored at the lower address and the high byte is stored at the neighboring higher address.

3.9.2 A102fc Color Cameras

For 8 bit depth

Set the Color_Coding_ID field of the Format_7, Mode_0 register to Mono 8 or Raw 8 (see Section 3.13.2 and page 4-10). With either of these IDs set, the camera outputs 8 bits per pixel.

For 12 bit depth

Set the Color_Coding_ID field of the Format_7, Mode_0 register for Raw 16. With this ID set, the camera outputs 16 bits per pixel but only 12 bits are effective. The effective pixel data fills from the LSB and the unused bits are filled with zeros. Pixel data is stored in the PC memory in little endian format, i.e., the low byte for each pixel is stored at the lower address and the high byte is stored at the neighboring higher address.

ⓘ	When a camera is operating in a mode that outputs 16 bits per pixel, the maximum frame rate at full resolution is 11.3 fps.

3.10 Strobe Control Output Signals

A102f cameras include a feature designed to help you control strobe lighting. The feature allows a user to enable and parameterize up to four strobe control output signals. The signals are designated as Strobe_0, Strobe_1, Strobe_2, and Strobe_3.

Enabling the Strobe Control Feature

The Strobe Signal Function control registers (see page 4-15) are used to enable and parameterize the strobe output signals. To enable and parameterize the strobe_0 signal:

- Set the On_Off field of the Strobe_0_Cnt register to 1 to enable the strobe_0 signal.
- Set the Signal_Polarity field to 0 for a low active signal or 1 for a high active signal as desired.
- Set the Delay_Value field. The setting in this field will determine the time between the start of image exposure and when the strobe signal changes state as shown in Figure 3-14.

  The delay can be set to a value between 0 and 4095. The units for the delay are 1/1024 ms. For example, if the delay was set to 100, the delay time would be 100/1024 ms (or approximately 97.7 µs).
- Set the Duration_Value field. The setting in this field will determine the duration of the strobe signal as shown in Figure 3-14.

  The duration can be set to a value between 0 and 4095. The units for the duration are 1/1024 ms.

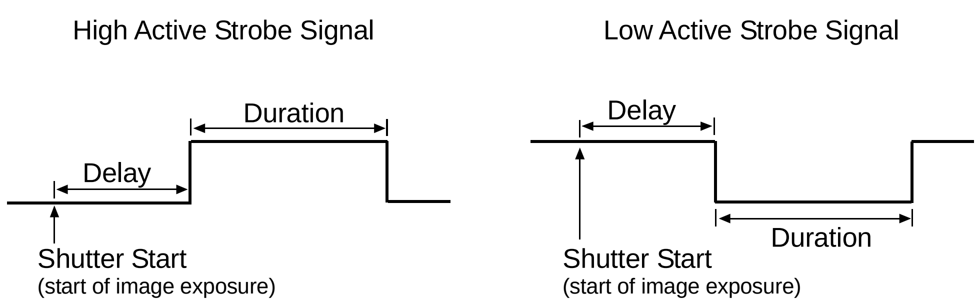


Figure 3-14: Strobe Signal

Once the Strobe_0 output signal has been enabled and parameterized, **it must be assigned to a physical output port on the camera**. The Strobe_0 signal can only be assigned to physical output port 0. See Section 5.7.11 for information on assigning output signals to physical output ports.

The Strobe_1_Cnt register is used to enable and parameterize the Strobe_1 output signal. The register is used in similar fashion as described for Strobe_0. Once the Strobe_1 output signal has been enabled and parameterized, it must be assigned to a physical output port on the camera. The Strobe_1 signal can only be assigned to physical output port 1.

The Strobe_2 and Strobe_3 output signals are also enabled, parameterized, and assigned to a physical output port in similar fashion to Strobe_0.

If you start an image exposure and the strobe signal for the previously captured image is still running, the running strobe signal ends immediately and the next delay and duration begin.

The Strobe Control Output Signal registers are defined in version 1.31 of the IIDC specification.

Because the strobe control output signal feature is so new, the Basler BCAM 1394 Driver does not yet include a method call to access this feature. As a work-around, you can use the techniques described in the tutorial for the Basler Smart Features Framework Software to directly read from or write to the strobe control output signal registers. The Smart Features Framework Software is available on the Basler web site. To download the software go to:

http://www.baslerweb.com/popups/popup_en_1825.php

3.11 Parallel Input/Output Control

A parallel I/O control feature is available on **A102f** cameras. The feature allows a user to set the state of the four physical output ports on the camera and to read the state of the four physical input ports.

To set state of the four physical output ports, write the contents of the PIO_Output register (see page 4-14). The bits are interpreted as follows:

- Bit 31 sets the state of physical output port 0.
- Bit 30 sets the state of physical output port 1.
- Bit 29 sets the state of physical output port 2.
- Bit 28 sets the state of physical output port 3.
- All other bits are ignored.

Writing to the PIO_Output register will only set the state of physical output ports that are configured as “User set.” For any output ports not configured as user set, the bit settings in this register will be ignored . See Section 5.7.11 for information on configuring physical output ports.

To determine the current state of the four physical input ports, read the contents of the PIO_Input register (see page 4-14). The bits are interpreted as follows:

- Bit 31 indicates the current state of physical input port 0.
- Bit 30 indicates the current state of physical input port 1.
- Bit 29 indicates the current state of physical input port 2.
- Bit 28 indicates the current state of physical input port 3.
- All other bits are ignored.

ⓘ	The PIO Control registers are defined in version 1.31 of the IIDC specification. Because the PIO control feature is so new, the Basler BCAM 1394 Driver does not yet include a method call to access this feature. As a work-around, you can use the techniques described in the tutorial for the Basler Smart Features Framework Software to directly read from or write to the PIO registers. The Smart Features Framework Software is available on the Basler web site. To download the software go to: http://www.baslerweb.com/popups/popup_en_1825.php

3.12 Available Video Formats, Modes, and Frame Rates on Monochrome Cameras

3.12.1 Standard Formats, Modes, and Frame Rates

The following standard video formats, modes, and frame rates are available on all **A102f** monochrome cameras:

Format_2, Mode_2, FrameRate_3 (Mono, 8 bits/pixel, 1280 x 960 pixels at 15 fps)

Format_2, Mode_6, FrameRate_2 (Mono, 16 bits/pixel, 1280 x 960 pixels at 7.5 fps)

ⓘ	When the camera is operating in Format_2, Mode_6, it outputs 16 bits per pixel but only 12 bits are effective. The effective pixel data fills from the LSB and the unused bits are filled with zeros. Pixel data is stored in the PC memory in little endian format, i.e., the low byte for each pixel is stored at the lower address and the high byte is stored at the neighboring higher address.

3.12.2 Customizable Formats and Modes

Formet_7, Mode_0 is available on **A102f** monochrome cameras.

Format_7, Mode_0

Format_7, Mode_0 is used to enable and set up the area of interest (AOI) feature described in Section 3.6. Format_7, Mode _0 is parameterized by using the Format_7, Mode_0 control and status registers (see page 4-10).

When the camera is operating in Format_7, Mode_0, the frame rate can be adjusted by setting the number of bytes transmitted in each packet. The number of bytes per packet is set by the BytePerPacket field of the Byte_Per_Packet register.

The value that appears in the MaxBytePerPacket field of the Packet_Para_Inq control register will show the maximum allowed bytes per packet setting given the current AOI settings. When the bytes per packet is set to the maximum, the camera will transmit frames at its maximum specified rate. By default, the AOI is set to use the full sensor area and the bytes per packet is set to 4092.

If you set the bytes per packet to a value lower than the maximum, the camera will transmit frames at a lower rate. The rate is calculated by the formula:

$$\text{Frames/s} = \frac{1}{\text{Packets per Frame x 125 µs}}$$

Keep in mind that when you lower the bytes per packet setting, the number of bytes needed to transmit a frame (the packets per frame) will increase. Due to limitations in the DCAM structure, a maximum of 4095 packets per frame is allowed. If you set the bytes per packet too low, the number of packets per frame will exceed the 4095 packet limit and the camera will not transmit frames properly.

ⓘ	When the camera is operating in Format_7, the Cur_V_Frame_Rate control register is not used and has no effect on camera operation.

Color Codings

In Format_7, Mode_0, the Mono 8 and Mono 16 color codings are available.

When the **Mono 8** ID is set in the Color_Coding_ID field of the Format_7, Mode_0 register, the camera outputs 8 bits per pixel.

When the **Mono 16** ID is set in the Color_Coding_ID field of the Format_7, Mode_0 register, the camera outputs 16 bits per pixel but only 12 bits are effective. The effective pixel data fills from the LSB and the unused bits are filled with zeros. Pixel data is stored in the PC memory in little endian format, i.e., the low byte for each pixel is stored at the lower address and the high byte is stored at the neighboring higher address.

ⓘ	When the camera is set for the Mono 16 color coding, the maximum frame rate at full resolution is 11.3 frames/s. Color code definitions can very from camera model to camera model. This is especially true for older models of Basler cameras.

3.13 Available Video Formats, Modes, and Frame Rates on Color Cameras

3.13.1 Standard Formats, Modes, and Frame Rates

The following standard video formats, modes, and frame rates are available on **A102fc** color cameras:

Format_2, Mode_0, FrameRate_2 (YUV 4:2:2, 16 bits/pixel ave., 1280 x 960 pixels at 7.5 fps)

Format_2, Mode_2, FrameRate_3 (Mono, 8 bits/pixel, 1280 x 960 pixels at 15 fps)

ⓘ	When the **A102**fc is operating in Format_2, Mode_2 (Mono), it outputs the raw data for each pixel, not the Y component as indicated in the DCAM specification. This is done so that the **A102**fc will have a standard output mode equivalent to the **A101**fc. This type of output is sometimes called "Bayer 8."

3.13.2 Customizable Formats and Modes

Format_7, Mode_0 and Format_7, Mode_1 are available on **A102fc** color cameras.

Format_7, Mode_0

Format_7, Mode_0 is used to enable and set up the area of interest (AOI) feature described in Section 3.6. Format_7, Mode _0 is parameterized by using the Format_7, Mode_0 control and status registers (see page 4-10).

When the camera is operating in Format_7, Mode_0, the frame rate can be adjusted by setting the number of bytes transmitted in each packet. The number of bytes per packet is set by the BytePerPacket field of the Byte_Per_Packet control register.

The value that appears in the MaxBytePerPacket field of the Packet_Para_Inq control register will show the maximum allowed bytes per packet setting given the current AOI settings. When the bytes per packet is set to the maximum, the camera will transmit frames at its maximum specified rate. By default, the AOI is set to use the full sensor area and the bytes per packet is set to 4092.

If you set the bytes per packet to a value lower than the maximum, the camera will transmit frames at a lower rate. The rate is calculated by the formula:

$$\text{Frames/Sec.} = \frac{1}{\text{Packets per Frame x } 125\ \mu\text{s}}$$

Keep in mind that when you lower the bytes per packet setting, the number of bytes needed to transmit a frame (the packets per frame) will increase. Due to limitations in the DCAM structure, a maximum of 4095 packets per frame is allowed. If you set the bytes per packet too low, the number of packets per frame will exceed the 4095 packet limit and the camera will not transmit frames properly.

ⓘ	When the camera is operating in Format_7, the Cur_V_Frame_Rate control register is not used and has no effect on camera operation.

Color Codings

In Format_7, Mode_0, the Mono 8, Raw 8, Raw 16, and YUV 4:2:2 color codings are available.

When the **Mono 8** ID is set in the Color_Coding_ID field of the Format_7, Mode_0 register, the camera outputs 8 bits per pixel and outputs only the Y component (brightness) of the YUV format.

When the **Raw 8** ID is set in the Color_Coding_ID field of the Format_7, Mode_0 register, the camera outputs 8 bits per pixel and outputs the raw data for each pixel. The pixel data is not processed to account for the color filter. (This type of output is sometimes called “Bayer 8.”)

When the **Raw 16** ID is set in the Color_Coding_ID field of the Format_7, Mode_0 register, the camera outputs 16 bits per pixel but only 12 bits are effective. The effective pixel data fills from the LSB and the unused bits are filled with zeros. Pixel data is stored in the PC memory in little endian format, i.e., the low byte for each pixel is stored at the lower address and the high byte is stored at the neighboring higher address. The camera outputs raw data for each pixel. The pixel data is not processed to account for the color filter.

When the **YUV 4:2:2** ID is set in the Color_Coding_ID field of the Format_7, Mode_0 register, the camera outputs image data in the YUV 4:2:2 format at an average of 16 bits per pixel.

When the camera is set for Format_7, Mode_0 and the Mono 8 color coding is selected, the available gray values are limited to a range from 16 to 234.

When the camera is set for the Raw 16 or the YUV 4:2:2 color coding, the maximum frame rate at full resolution is 11.3 frames/s.

The Raw 8 and Raw 16 color codings are defined in version 1.31 of the IIDC specification.

Color code definitions can very from camera model to camera model. This is especially true for older models of Basler cameras.

Format_7, Mode_1

Format_7, Mode_1 is very similar to Format_7, Mode_0 with the exception of the available color codings and how the color codings work.

Format_7, Mode_1 is parameterized by using the Format_7, Mode_1 control and status registers (see page 4-12).

Color Codings

In Format_7, Mode_1, only the Mono 8 color coding is available and the color coding works differently than the Mono 8 color coding available in Format_7, Mode_0.

When the **Mono 8** ID is set in the Color_Coding_ID field of the Format_7, Mode_1 register, the camera outputs 8 bits per pixel and outputs the raw data for each pixel. The pixel data is not processed to account for the color filter. (This type of output is sometimes called “Bayer 8.”)

This mode of operation is included on the **A102fc** for compatibility with versions 1.6 and below of the Basler BCAM 1394 driver software. These early versions of the BCAM software require a Mono 8 color coding that results in the output of raw pixel data.

4 Configuring the Camera

The **A102f** is configured by setting status and control registers as described in the “1394-Based Digital Camera Specification” issued by the 1394 Trade Association. (The specification is available at the 1394 Trade Association’s web site: www.1394ta.org.) Except where noted, all registers conform to version 1.31 of the specification.

If you are creating your own driver to operate the camera, Sections 4.1 through 4.4 provide the basic information you will need about the registers implemented in the camera along with some information about read/write capabilities.

A fully functional driver is available for Basler IEEE 1394 cameras such as the **A102f**. The Basler BCAM 1394 Driver/Software Development Kit includes an API that allows a C++ programmer to easily integrate camera configuration and operating functions into your system control software. The driver also includes a Windows® based viewer program that provides camera users with quick and simple tools for changing camera settings and viewing captured images.

The BCAM 1394 Driver/SDK comes with comprehensive documentation including a programmer’s guide and code samples. For more information, visit the Basler web site at: www.basler-vc.com.

4.1 Block Read and Write Capabilities

The camera supports block reads and block writes. If you do a single read or a block read, the camera will return a 0 for all non-existent registers. If you do a single write to a non-existent register or a block write that includes non-existent registers, the writes to non-existent registers will have no effect on camera operation.

Block reads or writes are limited to a payload of 32 quadlets.

4.2 Changing the Video Format setting

Whenever the Current Video Format setting is changed, you must also do the following:

If the Cur_V_Format is changed from Format 7 to Format 0, you must also write the Cur_V_Mode and the Cur_V_Frm_Rate.

If the Cur_V_Format is changed from Format 0 to Format 7, you must also write the Cur_V_Mode, the Image_Position, the Image_Size and the Bytes_Per_Packet. (See Section 3.12.2 for more information on setting the Bytes per Packet in Format 7).

4.3 Configuration ROM

The configuration ROM in the **A102f** is compliant with the DCAM specification V 1.30

4.4 Implemented Standard Registers

A list of all standard registers implemented in A102f appears below.

The base address for all camera control registers is:

Bus_ID, Node_ID, FFFF F0F0 0000

This address is contained in the configuration ROM in the camera unit directory. The offset field in each of the tables is the byte offset from the above base address.

4.4.1 Inquiry Registers

Camera Initialize Register

Offset	Name	Notes
000h	Initialize	Camera will reset itself, break any state lock, and re-initialize itself to factory settings.

Inquiry Register for Video Format

Offset	Name	Notes
100h	V_Format_Inq	The A102f and A102fc support: • Format_2 • Format_7.

Inquiry Registers for Video Mode

Offset	Name	Notes
188h	V_Mode_Inq_2 (Format 2)	In format_2, the A102f supports: • Mode_2 (1280 x 960, mono, 8 bits/pixel) • Mode_6 (1280 x 960, mono, 16 bits/pixel) In format_2, the A102fc supports: • Mode_0 (1280 x 960, YUV 4:2:2, 16 bits/pixel avg.) • Mode_2 (1280 x 960, mono, 8 bits/pixel)
19Ch	V_Mode_Inq_7 (Format 7)	In format_7, the A102f supports: • Mode_0 (1392 x 1040) In format_7, the A102fc supports: • Mode_0 (1388 x 1038) • Mode_1 (1388 x 1038)

Inquiry Registers for Video Frame Rate

Offset	Name	Notes
240h	V_Rate_Inq_2_0 (Format_2, Mode_0)	In format 2, mode 0, the A102fc supports the following frame rates: • 7.5 fps (frame rate 2)
248h	V_Rate_Inq_2_2 (Format_2, Mode_2)	In format 2, mode 2, the A102f and A102fc support the following frame rates: • 15 fps (frame rate 3)
258h	V_Rate_Inq_2_6 (Format_2, Mode_6)	In format 2, mode 6, the A102f supports the following frame rates: • 7.5 fps (frame rate 2)

CSR Inquiry Register for Format 7

Offset	Name	Notes
2E0h	V_CSR_Inq_7_0	The A102f and A102fc implement the CSR for: • Format_7, mode_0
2E4h	V_CSR_Inq_7_1	The A102fc implements the CSR for: • Format_7, mode_1

Inquiry Register for Basic Functions

Offset	Name	Notes
400h	Basic_Func_Inq	The A102f and A102fc support the following basic functions: • Advanced_Feature_Inq • Vmode_Error_Status_Inq • Feature_Control_Error_Status • One_Shot_Inq

Inquiry Registers for Feature Presence

Offset	Name	Notes
404h	Feature_Hi_Inq	The A102f and A102fc support the following features: • Brightness • Shutter • Gain • Trigger TheA102fc also supports: • White_Balance
408h	Feature_Lo_Inq	= 0
40Ch *	Opt_Function_Inq	The A102f and A102fc support the following features: • Parallel Input/Output Control • Strobe Signal
480h	Advanced_Feature_Inq	The A102f and A102fc implement a Basler advanced feature set.
484h *	PIO_Control_CSR_Inq	The A102f and A102fc support this feature.
48Ch *	Strobe_Output_CSR_Inq	The A102f and A102fc support this feature.

***** These registers are defined in version 1.31 of the IIDC specification.

Inquiry Registers for Feature Elements

Offset	Name	Notes
500h	Brightness_Inq	The A102f and A102fc support the following sub-features for brightness: • Presence_Inq • ReadOut_Inq • Manual_Inq • Min_Value = 0x000 • Max_Value = 0x0FF
50Ch	White_Bal_Inq	The A102fc supports the following sub-features for white balance: • Presence_Inq • ReadOut_Inq • Manual_Inq • Min_Value = 0x10 • Max_Value = 0xFF
51Ch	Shutter_Inq	The A102f and A102fc support the following sub-features for shutter: • Presence_Inq • ReadOut_Inq • Manual_Inq • Min_Value = 0x001 • Max_Value = depends on format and mode For format 2 or format 6, the max shutter setting depends on the frame rate: At 3.75 fps, max = 4095 (0xFFF) At 7.5 fps, max = 4095 (0xFFF) At 15 fps max = 3331 (0xD03) For format 7, the max shutter setting is 0xFFF
520h	Gain_Inq	The A102f and A102fc support the following sub-features for gain: • Presence_Inq • ReadOut_Inq • Manual_Inq • Min_Value = 0x0C0 • Max_Value = 0x3FF for all output modes except Mono 16 0x1FF for Mono 16

Offset	Name	Notes
530h	Trigger_Inq	The A102f and A102fc support the following sub-features for trigger: • Presence_Inq • ReadOut_Inq • OnOff_Inq • Polarity_Inq • Value_Read_Inq * • Trigger_Source_0_Inq * • Trigger_Source_1_Inq * • Trigger_Source_2_Inq * • Trigger_Source_3_Inq * • Software_Trigger_Inq * • Trigger_Mode0_Inq • Trigger_Mode1_Inq

***** These fields are defined in version 1.31 of the IIDC specification.

4.4.2 Control and Status Registers

Control and Status Registers for the Camera

Offset	Name	Notes
600h	Cur_V_Frame_Rate / Revision	Default = 0
604h	Cur_V_Mode	Default = 0
608h	Cur_V_Format	Default = Format 7
60Ch	Iso_Channel / Iso_Speed	Default Channel =0 Default Speed = S400
614h	Iso_En / Continuous Shot	Default = 0
61Ch	One_Shot / Multi_Shot	Default = 0 Multi_Shot is not supported
62Ch *	Software_Trigger	Write a 1 to set the trigger. Self cleared when Trigger Mode = 0
630h *	Data_Depth	Default = 8 Read only

* These registers are defined in version 1.31 of the IIDC specification.

Control and Status Registers for Features

Offset	Name	Notes
800h	Brightness	Brightness settings range from 0 to 255 (0x000 to 0x0FF). Default = 16 (0x10) for the A102f = 32 (0x20) for the A102fc See Section 3.5.2 for more information on setting the brightness.
80Ch	White_Balance	The U_Value_or_B_Value setting adjusts the blue level in the captured images. The setting ranges from 16 (0x10) to 255 (0xFF). When set to 64 (0x40), this register has no effect on the image. Settings lower than 64 make the images less blue. Settings greater than 64 make the images more blue. U_Value _or_B_Value default = 112 (0x70) The V_Value_or_R_Value setting adjusts the red level in the captured images. The setting ranges from 16 (0x10) to 255 (0xFF). When set to 64, this register has no effect on the image. Settings lower than 64 make the images less red. Settings greater than 64 make the images more red. V_Value_or_R_Value default = 64 (0x40)

Offset	Name	Notes
81Ch	Shutter	Shutter settings range from 1 (0x001) to 4095 (0xFFF). Exposure time = (shutter register value) x 20 µs Default = 500 (0x1F4) = 10 ms for the A102f = 562 (0x232) = 11.24 ms for the A102fc
820h	Gain	Gain settings range from 192 (0x0C0) to 1023 (0x3FF) for all output modes except Mono 16. For Mono 16, gain settings range from 192 (0x0C0) to 511 (0x1FF). A setting of 0x0C0 results in a gain of 1x (0dB) A setting of 0x3FF results in a gain of 35.5x (31 db) Default = 210 (0x0D2) for the A102f = 224 (0x0E0) for the A102fc
830h	Trigger_Mode	Setting On_Off to: 0 = disable trigger use {= default} 1 = enable trigger use Setting Trigger_Polarity to: 0 = low active {= default} 1 = high active Setting Trigger_Source * to: 0 = ExtTrig on physical input port 0 {= default} 1 = ExtTrig on physical input port 1 2 = ExtTrig on physical input port 2 3 = ExtTrig on physical input port 3 7 = Software trigger Setting Trigger_Mode to: 0 = mode 0 (programmable mode) 1 = mode 1 (level mode) {= default} When an external trigger is used, trigger mode 0 and trigger mode 1 are both valid. When a software trigger is used, only trigger mode 0 is valid. The parameter portion of the register is ignored.

* This field is defined in version 1.31 of the IIDC specification.

Control and Status Registers for Format_7, Mode_0

The base address for each Format_7, Mode_0 camera control register is:

Bus_ID, Node_ID, FFFF F1F0 0000

The offset field in the table is the byte offset from the above base address.

Offset	Name	Notes
000h	Max_Image_Size_Inq	A102f: Hmax = 1392, Vmax = 1040 A102fc: Hmax = 1388, Vmax = 1038
004h	Unit_Size_Inq	A102f: Hunit = 1, Vunit = 1 A102fc: Hunit = 2, Vunit = 2
008h	Image_Position	Default = (0, 0)
00Ch	Image_Size	Default = (Hmax, Vmax)
010h	Color_Coding_ID	A102f default = Mono 8 (ID = 0) A102fc default = YUV 4:2:2 (ID = 2)
014h	Color_Coding_Inq	The A102f supports the following color codes in this mode: • Mono 8 (ID = 0) • Mono 16 (ID = 5) The A102fc supports the following color codes in this mode: • Mono 8 (ID = 0) • Raw 8 * (ID = 9) • Raw 16 * (ID = 10) • YUV 4:2:2 (ID = 2)
034h	Pixel_Number_Inq	The value of this register depends on the following registers: • Image_Size
038h	Total_Bytes_Hi_Inq	The value of this register depends on the following registers: • Image_Size • Color_Coding_ID The value covers the following data: • Image data • Padding bytes
03Ch	Total_Bytes_Lo_Inq	See TOTAL_BYTES_HI_INQ register
040h	Packet_Para_Inq	UnitBytePerPacket = 4 MaxBytePerPacket depends on: • Image_Size • Color_Coding_ID

* The Raw 8 and Raw 16 Color_Coding_IDs are defined in version 1.31 of the IIDC specification.

Offset	Name	Notes
044h	Byte_Per_Packet	BytePerPacket = set by user * RecBytePerPacket = 4092 (0xFFC)
048h	Packet_Per_Frame_Inq	The value of this register depends on the following registers: • Image_Size • Color_Coding_ID • Byte_Per_Packet
04Ch	Unit_Position_Inq	A102f: Hposunit = 1, Vposunit = 1 A102fc: Hposunit = 2, Vposunit = 2
054h **	Data_Depth	Indicates the effective data depth of the pixel information being transmitted from the camera. The value of this register depends on the following registers: • Color_Coding_ID In Format _7, Mode_0: *Color_Coding_ID* — *Effective Data Depth* — *Bits Transmitted* ID = 0 (Mono 8) — 8 — 8 ID = 9 (Raw 8) ** — 8 — 8 ID = 5 (Mono 16) — 12 — 16 ID = 10 (Raw 16) ** — 12 — 16 ID = 2 (YUV 4:2:2) — 8 — 16
058 **	Color_Filter_ID	Identifies the order of the pixel colors transmitted out of the camera on cameras equipped with an RGB primary color filter (such as the Bayer filter used on Basler A102fc cameras). The A102fc supports the following color filter IDs: • ID 0 • ID 1 • ID 2 • ID 3

* When you lower the bytes per packet setting, the number of bytes needed to transmit a frame (the packets per frame) will increase. Due to limitations in the DCAM structure, a maximum of 4095 packets per frame is allowed. If you set the bytes per packet too low, the number of packets per frame will exceed the 4095 packet limit and the camera will not transmit frames properly.

** The Data_Depth register, Color_Filter_ID register, Raw 8 Color_Coding_ID, and Raw 16 Color_Coding_ID are defined in version 1.31 of the IIDC specification.

Control and Status Registers for Format_7, Mode_1

Format_7, Mode_1 is available on color **A102f**c cameras only. The base address for each Format_7, Mode_1 camera control register is:

Bus_ID, Node_ID, FFFF F1F0 0100

The offset field in the table is the byte offset from the above base address.

Offset	Name	Notes
000h	Max_Image_Size_Inq	Hmax = 1388, Vmax = 1038
004h	Unit_Size_Inq	Hunit = 2, Vunit = 2
008h	Image_Position	Default = (0, 0)
00Ch	Image_Size	Default = (Hmax, Vmax)
010h	Color_Coding_ID	A102fc default =Mono 8
014h	Color_Coding_Inq	The A600fc supports the following color codes in this mode: • Mono 8 (Raw data)
034h	Pixel_Number_Inq	The value of this register depends on the following registers: • Image_Size
038h	Total_Bytes_Hi_Inq	The value of this register depends on the following registers: • Image_Size • Color_Coding_ID The value covers the following data: • Image data • Padding bytes
03Ch	Total_Bytes_Lo_Inq	See TOTAL_BYTES_HI_INQ register
040h	Packet_Para_Inq	UnitBytePerPacket = 4 MaxBytePerPacket depends on: • Image_Size • Color_Coding_ID

Offset	Name	Notes
044h	Byte_Per_Packet	BytePerPacket = set by user * RecBytePerPacket = 4092 (0xFFC)
048h	Packet_Per_Frame_Inq	The value of this register depends on the following registers: • Image_Size • Color_Coding_ID • Byte_Per_Packet
048h	Packet_Per_Frame_Inq	The value of this register depends on the following registers: • Image_Size • Color_Coding_ID • Byte_Per_Packet
04Ch	Unit_Position_Inq	Hposunit = 2 Vposunit = 2
054h **	Data_Depth	Indicates the effective data depth of the pixel information being transmitted from the camera. The value of this register depends on the following registers: • Color_Coding_ID In Format _7, Mode_0: *Color_Coding_ID* — *Effective Data Depth* — *Bits Transmitted* ID = 0 (Mono 8) — 8 — 8
054 **	Color_Filter_ID	Identifies the order of the pixel colors transmitted out of the camera on cameras equipped with an RGB primary color filter (such as the Bayer filter used on Basler A102fc cameras). The A102fc supports the following color filter IDs: • ID 0 (RG/GB) • ID 1 (GB/RG) • ID 2 (GR/BG) • ID 3 (BG/GR)

* When you lower the bytes per packet setting, the number of bytes needed to transmit a frame (the packets per frame) will increase. Due to limitations in the DCAM structure, a maximum of 4095 packets per frame is allowed. If you set the bytes per packet too low, the number of packets per frame will exceed the 4095 packet limit and the camera will not transmit frames properly.

** The Data_Depth register and Color_Filter_ID register are defined in version 1.31 of the IIDC specification.

Control and Status Registers for the PIO Control Function

The base address for the PIO Control Function control and status registers is:

Bus_ID, Node_ID, FFFF F2F0 00C8

The offset field in the table is the byte offset from the above base address.

Offset	Name	Notes
000h *	PIO_Output	Bit 31 sets the state of physical output port 0. Bit 30 sets the state of physical output port 1. Bit 29 sets the state of physical output port 2. Bit 28 sets the state of physical output port 3.
004h *	PIO_Input	Bit 31 indicates the current state of physical input port 0. Bit 30 indicates the current state of physical input port 1. Bit 29 indicates the current state of physical input port 2. Bit 28 indicates the current state of physical input port 3.

* These registers are defined in version 1.31 of the IIDC specification.

ⓘ	The PIO_Output register will only set the state of physical output ports that are configured as “User set” (see Section 5.7.11). For any output ports not configured as user set, the bit settings in this register will be ignored .

Control and Status Registers for the Strobe Signal Output Function

The base address for the Strobe Signal Output Function control and status registers is:

Bus_ID, Node_ID, FFFF F2F0 0300

The offset field in the table is the byte offset from the above base address.

Offset	Name	Notes
000h *	Srobe_CTRL_Inq	The A102f and A102fc support the following strobe signals: • Strobe_0 • Strobe_1 • Strobe_2 • Strobe_3
100h *	Strobe_0_Inq	The A102f and A102fc support the sub-features for Strobe_0: • Presence_Inq • ReadOut_Inq • On/Off_Inq • Polarity_Inq • Min_Value • Max_Value
104h *	Strobe_1_Inq	Same definition as Strobe_0_Inq.
108h *	Strobe_2_Inq	Same definition as Strobe_0_Inq.
10Ch *	Strobe_3_Inq	Same definition as Strobe_0_Inq.
200h *	Strobe_0_Cnt	Setting On_Off to: 0 = disable Strobe_0 {= default} 1 = enable Strobe_0 Setting Signal_Polarity to: 0 = low active 1 = high active {= default} The delay value can range from 0 (0x000) to 4095 (0xFFF). The units for the delay are 1/1024 ms. For example, if the delay setting is 100 (0x064), the delay will be 100/1024 ms (~ 97.7 µs). The duration value can range from 0 (0x000) to 4095 (0xFFF). The units for the delay are 1/1024 ms.

* These registers are defined in version 1.31 of the IIDC specification.

Offset	Name	Notes
204h *	Strobe_1_Cnt	Same definition as Strobe_0_Cnt.
208h *	Strobe_2_Cnt	Same definition as Strobe_0_Cnt.
20Ch *	Strobe_3_Cnt	Same definition as Strobe_0_Cnt.

* These registers are defined in version 1.31 of the IIDC specification.

If a strobe signal is enabled, the signal will only be present on the associated output port when the output port is configured for “strobe.” For example, if the Strobe_0 signal is enabled, the signal will only be present on physical output port 0 when the port is configured for “strobe.” If the Strobe_1 signal is enabled, the signal will only be present on physical output port 1 when the port is configured for “strobe.” Etc.

See Section 5.7.11 for more information about configuring the output ports.

4.5 Basler Advanced Features Registers

ⓘ	The functionally of the advanced features control and status registers has been made part of Basler's Smart Features Framework (SFF). We strongly recommend that you use the SFF registers (see Section) rather than the advanced features registers. To ensure backward compatibility, the existing advanced features registers described below will continue to be supported, but their functionality will not be extended in the future.

The base address for the advanced features registers is:

Bus_ID, Node_ID, FFFF F2F0 0000

This address is contained in the Advanced_Feature_Inq register of the "Inquiry register for feature presence" section.

The offset field in each of the tables is the byte offset from the above base address.

4.5.1 Advanced Features Access Control Register

Advanced Features Access Control Register

Offset	Name	Notes
0000h	Access_Control_Register	See DCAM Spec. V1.30 page 26.

4.5.2 Advanced Features Inquiry Registers

Inquiry Register for Advanced Features (High)

Offset	Name	Field	Bit	Description
0008h	Adv_Inq_Hi	Presence	[0]	Presence of this feature
		Test Image	[1]	Presence of test images
		Shading	[2]	Presence of shading correction
		---	[3]	Reserved
		Extd. Versions	[4]	Presence of extended versions
		Firmware Upload	[5]	Presence of firmware upload capability
		---	[6..31]	Reserved

Inquiry Register for Advanced Features (Low)

Offset	Name	Field	Bit	Description
000Ch	Adv_Inq_Lo	Presence	[0]	Presence of this feature
		---	[1..31]	Reserved

Inquiry Register for Extended Version Information

Offset	Name	Field	Bit	Description
1010h	Extd_Versions_Inq	Presence	[0]	Presence of this feature
		---	[1..7]	Reserved
		Length	[8..15]	Specifies the length in quadlets of the “String” field in the Extended Versions Information Register (see below).
		---	[16..31]	

Extended Versions Information Register

<table>
<tr><th>Offset</th><th>Name</th><th>Field</th><th>Bit</th><th>Description</th></tr>
<tr><td>1014h</td><td>Extd_Versions</td><td>String
(Read only)</td><td>[n Bytes]</td><td>An ASCII character string that includes the software version numbers for the camera. The length of this string field is equal to the number of quadlets given in the “Length” field of the Inquiry Register for Extended Version Information (see above).</td></tr>
<tr><td colspan="5">This string contains the camera’s “firmware ID” number. You can read the string to determine your camera’s firmware ID. The ID number’s position in the string is described in Section 1.1.
For troubleshooting purposes, Basler technical support may ask you to read this register and to supply the results.</td></tr>
</table>

Status and Control Register for Test Images

This advanced features register can be used to control the operation of the camera's test image feature (see Section 5.7.6 for a description of the available test images).

Offset	Name	Field	Bit	Description
0098h	Test_Image	Presence_Inq (Read only)	[0]	Presence of this feature 0: N/A 1: Available
		---	[1..7]	Reserved
		Image_Inq_1 (Read only)	[8]	Presence of test image 1 0: N/A 1: Available
		Image_Inq_2 (Read only)	[9]	Presence of test image 2 0: N/A 1: Available
		Image_Inq_3 (Read only)	[10]	Presence of test image 3 0: N/A 1: Available
		Image_Inq_4 (Read only)	[11]	Presence of test image 4 0: N/A 1: Available
		Image_Inq_5 (Read only)	[12]	Presence of test image 5 0: N/A 1: Available
		Image_Inq_6 (Read only)	[13]	Presence of test image 6 0: N/A 1: Available
		Image_Inq_7 (Read only)	[14]	Presence of test image 7 0: N/A 1: Available
		---	[15]	Reserved
		Image_On (Read/write)	[16..18]	0: No test image active 1: Test image 1 active 2: Test image 2 active 3: Test image 3 active
		---	[19..31]	Reserved

5 Smart Features and the Smart Features Framework

5.1 What are Smart Features

Smart features are features unique to Basler cameras. Test Images, the Cycle Time Stamp, and the CRC Checksum are examples of Basler smart features.

In some cases, enabling a smart feature will simply change the behavior of the camera. The Test Image feature is a good example of this type of smart feature. When the Test Image feature is enabled, the camera outputs a test image rather than a captured image.

When certain smart features are enabled, the camera actually develops some sort of information about each image that it acquires. In these cases, the information is added to each image as trailing data when the image is transmitted from the camera. Examples of this type of smart feature are the Cycle Time Stamp feature and the CRC Checksum. When the Cycle Time Stamp feature is enabled, after an image is captured, the camera determines when the acquisition occurred and develops a cycle time stamp for the image. And if the CRC Checksum feature is enabled, the camera calculates a checksum for the image. The cycle time stamp and checksum are added as trailing data to each image as the image is transmitted from the camera.

5.2 What is the Smart Features Framework

The first component of the Smart Features Framework (SFF) is a mechanism that allows you to enable and to parametrize smart features. This mechanism is essentially an extension of the register structure defined in the DCAM specification for use with “Advanced Features.” The SFF establishes a register for each smart feature. By setting bits within the register for a particular smart feature, you can enable the feature and control how the feature operates.

When certain smart features are enabled, the camera actually develops some sort of data about each image that it acquires. For example, when the Cycle Time Stamp feature is enabled, the camera creates a time stamp for each image based on when the image exposure started. In the cases where a smart feature develops some sort of data about a captured image, the smart feature’s data is added as trailing data to each image as the image is transmitted from the camera.

The SFF provides a mechanism for parsing the smart features data added to images transmitted out of the camera by assigning a unique identifier (GUID) to each smart feature. Whenever the camera adds data for a smart feature to an image, it includes the GUID for the smart feature as part of the added data. The GUIDs are especially useful when you enable several smart features that add data to the image stream. The GUIDs make it possible to identify which portion of the added data is the result of each enabled smart feature. Refer to Sections 5.6 and 5.7 for detailed information about getting smart features results.

5.3 What do I Need to Use Smart Features

To use smart features you will need:

- A camera that supports smart features. Not all camera models support smart features. And with some camera models that do support smart features, you may find that older cameras may not support all available smart features or may not support smart features at all. Section 5.5 contains information about checking a camera to see if it supports smart features.
- A method of accessing the camera's DCAM register structure. We strongly recommend that you use the Basler BCAM 1394 Driver (v1.7 or higher) along with the Basler Smart Features Framework software to access the registers. (See Section 5.4 for more information about the SFF Software.)

> We strongly recommend that you use the Basler BCAM 1394 driver. However, any driver that can get images in format 7 and that provides access to the DCAM registers can be used to work with smart features. If you do use a different driver, you can adapt the access techniques described in the SFF Software tutorial (see Section 5.4) to the driver you are using.
>
> You should be aware that drivers other than the Basler BCAM driver have not been tested with smart features.

5.4 What is the Smart Features Framework Software?

A Smart Features Framework Software (SFF Software) package is available from Basler. The SFF Software has two major components:

- An SFF Viewer. The viewer is a Windows® based tool that allows you to easily enable and disable smart features, parameterize the camera, capture and view images, and view smart features results.
- An SFF Tutorial. The tutorial explains how to access the cameras smart features from within your own applications. The tutorial is based on the assumption that you are using the Basler BCAM 1394 driver with your camera.

The SFF software package is available for download at the Basler web site. To download the software, go to:

http://www.baslerweb.com/popups/popup_en_1825.php

> The SFF Viewer will only work on PCs that have the BCAM driver v1.7 or higher installed.

5.5 Enabling and Parameterizing Smart Features

The camera provides a control and status register (CSR) for each smart feature (see Sect 5.7 for details of each feature and its CSR). To enable and parameterize a smart feature, the following steps must be performed:

1. Check to see if the camera supports smart features.
2. Ask the camera for the address of the CSR for the desired smart feature.
3. Enable and parameterize the desired smart features.

The next two sections describe steps 1 and 2. The layout of the registers used to enable and parameterize the smart features is described in section 5.7.

5.5.1 Checking to see if the Camera Supports Smart Features

Smart features are vendor unique. Such features are referred to in the 1394 Trade Association DCAM standard as advanced features. The DCAM standard specifies how vendors should implement advanced features. According to the standard, advanced features must be unlocked (that is, enabled) by writing an advanced features set identifier (Feature ID) and a time-out value to the Advanced Features Access Control Register. From the point of view of the DCAM standard, smart features are a set of advanced DCAM features. The Feature ID associated with Basler smart features is 0x0030 533B 73C3.

For Basler cameras, unlocking advanced features is not strictly necessary because any implemented smart features are always available. However, the unlock mechanism is also used to check to see if a camera supports vendor unique features such as smart features. If a device doesn’t recognize a Feature ID written to the Access Control Register, a value of 0xFFFF FFFF FFFF FFFF will be read back from the ACR. This value indicates that the device does not implement the feature set associated with that Feature ID.

Assuming that the address of the Advanced Features Access Control Register is 0xFFFF F2F0 0000, perform the following steps to see if a camera is smart features capable:

1. Write the quadlet data 0x0030 533B to 0xFFFF F2F0 0000
2. Write quadlet data 0x73C3 F000* to 0xFFFF F2F0 0004
3. Read quadlet data from 0xFFFF F2F0 0000 and 0xFFFF F2F0 0004. If at least one of the read operations returns a value that is not equal to 0xFFFF FFFF, the camera supports smart features. If both read operations return 0xFFFF FFFF, the camera does not support smart features.

Note that instead of performing two single quadlet write operations, a block write can be performed.

* The last three zeros in this quadlet data represent a timeout value. (Refer to the section of the IIDC specification that describes “CSRs for Advanced Features.”)

5.5.2 Determining the Address a Smart Feature's CSR

The control and status register (CSR) for each smart feature is identified by a 128 bit Globally Unique Identifier (GUID). GUIDs are also known as UUIDs (Universal Unique Identifier).

A GUID consists of:

- One 32 bit number (D1)
- Two 16 bit numbers (D2, D3)
- A sequence of 8 bytes (D4[0] - D4[7])

GUID example:

CA8A916A	-	14A4	-	4D8E	-	BBC9	-	93DF50495C16
(D1)		(D2)		(D3)		(D4[0] - D4[1])		(D4[2] - D4[7])

Section 5.7 describes the standard smart features available on **A102f** cameras. Each smart feature description includes the GUID assigned to the feature's CSR.

To determine the starting address of a smart feature's CSR, the feature's CSR GUID must be written to the Smart Features Inquiry register (SF_Inq_Register). The SF_Inq_Register's offset relative to the Access Control Register is 0x10. If the camera recognizes the GUID as the CSR GUID for an implemented smart feature, the address of CSR for the feature can be read from the Smart Features Address Register (SF_Addr_Register) at offset 0x20. If the feature isn't supported by the device, a value of 0x0 will be read from the SF_Addr_Register.

SF_Inq_Register Layout

	Bit			
Offset	0-7	8-15	16-23	24-31
10h	D1			
14h	D3		D2	
18h	D4[3]	D4[2]	D4[1]	D4[0]
1ch	D4[7]	D4[6]	D4[5]	D4[4]

SF_Addr_Register Layout

	Bit			
Offset	0-7	8-15	16-23	24-31
20h	Address_Lo			
24h	Address_Hi			

Example

Determine the address of the "CRC Checksum" smart feature which has a CSR GUID of:

3B34004E - 1B84 - 11D8 - 83B3 - 00105A5BAE55

D1: 0x3B34 004E

D2: 0x1B84

D3: 0x11D8

D4[0]: 0x83

D4[1]: 0xB3

D4[2]: 0x00

D4[3]: 0x10

D4[4]: 0x5A

D4[5]: 0x5B

D4[6]: 0xAE

D4[7]: 0x55

Step 1: Write the CSR GUID to the SF_Inq_Register

Assuming that the address for the ACR is 0xFFFF F2F0 0000, perform the following quadlet write operations to the SF_Inq_Register

a. Write quadlet data 0x3B34 004E to 0xFFFF F2F0 0010 (D1)

b. Write quadlet data 0x11D8 1B84 to 0xFFFF F2F0 0014 (D3 | D2)

c. Write quadlet data 0x1000 B383 to 0xFFFF F2F0 0018 (D4[3] | D[0])

d. Write quadlet data 0x55AE 5B5A to 0xFFFF F2F0 001C (D4[7] | D[4])

Instead of performing four quadlet write operations, one block write operation can be performed.

Step 2: Read the start address for the smart feature from the SF_Addr_register

a. Read quadlet data from 0xFFFF F2F0 0020 (Address_Lo)

b. Read quadlet data from 0xFFFF F2F0 0024 (Address_Hi)

If both Address_Lo and Address_Hi return zero, the camera doesn't support the CRC checksum feature. Assuming the read operations yielded Address_Lo = 0xF2F0 0038 and Address_Hi = 0x0000 FFFF, the CRC Checksum feature CSR's address is 0xFFFF F2F0 0038.

5.5.3 Enabling and Parameterizing a Smart Feature

Once you have determined the starting address of the control and status register (CSR) for your desired smart feature, you are ready to enable and parameterize the feature by setting bits within the CSR.

Section 5.7 describes the standard smart features available on **A102f** cameras. Each smart features description includes an explanation of what the feature does and an explanation of the parameters associated with the feature. The descriptions also include a detailed layout of how the bits contained within the feature's CSR relate to the parameters for the feature. After reading the description of your desired smart feature, you can enable and parameterize the feature by setting the appropriate bits within the CSR.

5.6 Getting Smart Features Results

In many cases, activating a smart feature results in additional data that must be transmitted by the camera, i.e., the results of the smart feature. The results of a smart feature will be appended to the image data so that each frame contains both image data and smart features results.

Before using any of the smart features that add information to the image data, the extended data stream feature must be enabled. The extended data stream is in itself a smart feature. When the extended data stream feature is enabled, information such as the height of the image, the width of the image, and the AOI size is added to each image's basic pixel data. Disabling the extended data stream feature switches off all smart features that add information to the image data stream.

The extended data stream feature and any other smart features which add information to the image data stream will only work when the camera is set for video format 7. For other video formats, enabling the extended data stream feature or any of the other smart features that normally add data to the image stream does not affect the image data stream; the camera only sends the basic image data without any added information.

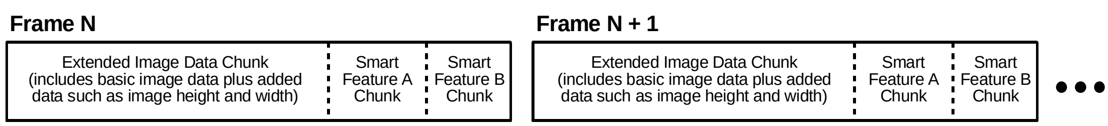


Figure 5-1: Image Data Stream with Smart Features Enabled

As illustrated in Figure 5-1, when smart features are enabled, each image frame consists of "chunks." For example, the frame may include a chunk which contains the extended image data (the basic image data plus the added height, width, etc. information), a chunk which contains the results for the frame counter smart feature, a chunk which contains the results for the cycle time stamp smart feature, etc. Table 5-1 describes the general structure of a chunk.

Position	Name	Description
0	Data [K Bytes]	The data that the chunk is transporting.
K	Chunk GUID [16 Bytes]	Identifies the type of chunk and the smart feature associated with the chunk. (Note that a smart feature's chuck GUID is not the same as its CSR GUID.)
K+16	Length [4 Bytes]	The chunk's total length in bytes.
K+20	Inverted Length [4 Bytes]	The bitwise complement of the length.

Table 5-1: General Structure of a Chunk

Each chunk ends with a four byte unsigned integer indicating the length of the chunk and four bytes which indicate the bitwise complement of the length. Transferring both the chunk length and the bitwise complement of the length serves as a mechanism to detect transmission errors. If the last four bytes of a chunk aren't the bitwise complement of the preceding four bytes, the chunk's length information isn't valid and this indicates that a transmission error occurred.

There are different types of chunks, for example, the chunk that is added when the cycle time stamp smart feature is enabled and the chunk that is added when the frame counter smart feature is enabled. Although most chunks follow the general structure described in Table 5-1, each type of chunk has unique aspects to its layout. To allow you to distinguish between the chunks, each chunk carries a “chunk GUID”. The GUID for each chunk is transferred just before the chunk’s length information. If you look through the descriptions of the smart features in Section 5.7, you will notice that for smart features which add a chunk to the image data stream, there is a description of the layout of the chunk and the chunk GUID associated with the chunk.

A chunk’s length field contains the chunk’s total length in bytes. The GUID, the length, and the inverted length are included as part of the total chunk length.

By appending length information and a chunk GUID to each chunk, the camera sends a self-describing data stream and allows easy navigation through the individual chunks that make up a complete image data frame.

Don’t confuse CSR GUIDs with chunk GUIDs:

- Each smart feature has a control and status register (CSR) associated with it and each CSR has a unique “CSR GUID” assigned to the register. The CSR GUIDs are used to help you keep track of which CSR is associated with each smart feature.
- Any smart feature that adds a “chunk” of data to the image data stream also has a unique “chunk GUID” assigned to the feature. The chunk GUID will be included the chunk of data that a smart feature adds to the image data. The chunk GUIDs let you determine which smart feature is associated with each added chunk in the image data stream.

The CRC Checksum is an exception to the general structure of a chunk. See Section 5.7.5 for more information.

5.6.1 How Big a Buffer Do I Need?

When smart features that add data to the image are enabled, the size of each transmitted frame will be larger than you would normally expect for a frame which contains only image data. To determine the size of the buffer that you will need to hold an image with appended smart features data, check the Total_Bytes_Hi_Inq and Total_Bytes_Lo_Inq fields of the Format_7 register for the mode you are currently using. Make sure to check these fields after all smart features have been enabled and all other settings affecting the image size have been completed. The size information in these fields will allow you to properly set up buffers to receive the transmitted images.

5.7 Smart Features on the A102f

5.7.1 Extended Data Stream

The extended data stream feature has two functions:

- When it is enabled, information such as image height, image width, and AOI size is added to the basic pixel data for each image.
- It must be enabled before you can use any other smart feature that adds information to the image data stream.

With the extended data stream feature enabled, the basic pixel data for each image and the added information such as the image height and width are included in an "extended data chunk". Refer to the extended data chunk layout below for a complete description of the information included in the extended data chunk.

> The extended data stream feature must be enabled in order to use any of the other smart feature that adds information to the image data stream. Disabling the extended data stream feature switches off all smart features that add information to the image data stream.
>
> The extended data stream feature and any other smart features which add information to the image data stream will only work when the camera is set for video format 7.

Control and Status Register for the Extended Data Stream Feature

Name	Extended_Data_Stream	
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.	
CSR GUID	4E7ABCB0 - 1B84 - 11D8 - 9651 - 00105A5BAE55	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
----	[1 ... 30]	Reserved
Enable (Read / write)	[31]	Enable / Disable this feature 0: Disable 1: Enable

Extended Data Chunk Layout

Position	Name	Description
0	Pixel Data [K Bytes]	The pixel data from the captured image
K	Gap [M Bytes]	For technical reasons, there might be a gap between the pixel data and the other data in the extended image data.

K + M	Stride [4 Bytes]	Signed integer. Indicates the number of bytes needed to advance from the beginning of one row in an image to the beginning of the next row.
K + M + 4	Reserved [3 Bytes]	-----
K + M + 7	Data Depth [1 Byte]	Effective data depth in bits of the pixels in the image.
K + M + 8	Top [2 Bytes]	Y coordinate of the top left corner of the current area of interest (AOI).
K + M + 10	Left [2 Bytes]	X coordinate of the top left corner of the current AOI.
K + M + 12	Height [2 Bytes]	Height in pixels of the current AOI.
K + M + 14	Width [2 Bytes]	Width in pixels of the current AOI.
K + M + 16	Reserved [3 Bytes]	-----
K + M + 19	Color Coding ID [1 Byte]	Color coding ID which describes the pixel data format. See DCAM V1.31 page 38. See also Sections 3.12.2 and 3.13.2 of this document.
K + M + 20	Reserved [3 Bytes]	-----
K + M + 23	Color Filter ID [1 Byte]	For color cameras, describes the orientation of the color filter to the current AOI. See DCAM V1.31 page 39. See also Section 3.8.2 of this document.
K + M + 24	Chunk GUID [16 Bytes]	94ED7C88 - 1C0F - 11D8 - 82E0 - 00105A5BAE55
K + M + 40	Chunk Length [4 Bytes]	This chunk's total length in bytes.
K + M + 44	Inverted Chunk Length [4 Bytes]	The bitwise complement of the chunk length.

5.7.2 Frame Counter

The frame counter feature numbers images sequentially as they are captured. The counter starts at 0 and wraps at 4294967296. The counter increments by one for each captured frame. Whenever the camera is powered off, the counter will reset to 0.

Note that if the camera is in continuous shot mode and continuous capture is stopped, up to two numbers in the counting sequence may be skipped. This happens due to the internal image buffering scheme used in the camera.

ⓘ	The extended data stream feature (see Section 5.7.1) must be enabled in order to use the frame counter feature or any of the other smart feature that adds information to the image data stream. Disabling the extended data stream feature switches off all smart features that add information to the image data stream. The frame counter feature will only work when the camera is set for video format 7.

Control and Status Register for the Frame Counter Feature

Name	Frame_Counter	
Address	See “Determining the Address of Smart Features CSRs” on page 5-4.	
CSR GUID	4433C4A4 - 1B84 - 11D8 - 86B2 - 00105A5BAE55	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
----	[1 ... 30]	Reserved
Enable (Read / write)	[31]	Enable / Disable this feature 0: Disable 1: Enable

Frame Counter Chunk Layout

Position	**Name**	**Description**
0	Counter [4 Bytes]	The frame counter.
4	Chunk GUID [16 Bytes]	8C5DB844 - 1C0F - 11D8 - 965F - 00105A5BAE55
20	Chunk Length [4 bytes]	This chunk’s total length in bytes.
24	Inverted Chunk Length [4 bytes]	The bitwise complement of the chunk length.

5.7.3 Cycle Time Stamp

The cycle time stamp feature adds a chunk to each image frame containing the value of the counters for the IEEE 1394 bus cycle timer. The counters are sampled at the start of exposure of each image.

> The extended data stream feature (see Section 5.7.1) must be enabled in order to use the cycle time stamp feature or any of the other smart feature that adds information to the image data stream. Disabling the extended data stream feature switches off all smart features that add information to the image data stream.
>
> The cycle time stamp feature will only work when the camera is set for video format 7.

Control and Status Register for the Cycle Time Stamp Feature

Name	Cycle_Time_Stamp	
Address	See “Determining the Address of Smart Features CSRs” on page 5-4.	
CSR GUID	5590D58E - 1B84 - 11D8 - 8447 - 00105A5BAE55	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
----	[1 ... 30]	Reserved
Enable (Read / write)	[31]	Enable / Disable this feature 0: Disable 1: Enable

Cycle Time Stamp Chunk Layout

Position	**Name**	**Description**		
0	Cycle Time Stamp [4 Bytes]	**Field**	**Bit**	**Description**
		Second Count	[0 ... 6]	Counts the seconds. Wraps to zero after 127 seconds.
		Cycle Count	[7 ... 19]	Counts the 125 µs isochronous bus cycles. Wraps to zero after counting to 7999.
		Cycle Offset	[20 ... 31]	Counts at 24.576 MHz and wraps to zero after counting to 3071 (resulting in a 125 µs cycle)
4	Chunk GUID [16 Bytes]	994DD430 - 1C0F - 11D8 - 8F6B - 00105A5BAE55		
20	Chunk Length [4 Bytes]	This chunk’s total length in bytes.		
24	Inverted Chunk Length [4 Bytes]	The bitwise complement of the chunk length.		

5.7.4 DCAM Values

The DCAM values feature adds a chunk to each image frame containing the current settings for some standard DCAM features. The settings are sampled at the start of exposure of each image.

ⓘ	The extended data stream feature (see Section 5.7.1) must be enabled in order to use the DCAM values feature or any of the other smart feature that adds information to the image data stream. Disabling the extended data stream feature switches off all smart features that add information to the image data stream. The DCAM values feature will only work when the camera is set for video format 7.

Control and Status Register for the DCAM Values Feature

Name	DCAM_Values	
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.	
CSR GUID	494DE528 - 1B84 - 11D8 - 8A0C - 00105A5BAE55	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
----	[1 ... 30]	Reserved
Enable (Read / write)	[31]	Enable / Disable this feature 0: Disable 1: Enable

DCAM Values Chunk Layout

Position	Name	Description		
0	Gain CSR [4 Bytes]	Content of the DCAM Gain CSR		
		Field	**Bit**	**Description**
		Presence_Inq	[0]	Presence of this feature If 0, the DCAM feature is not available and all of its values should be ignored
		Abs_Control	[1]	Absolute control mode If 1, the DCAM feature is in absolute control mode and the current value can be read from the Absolute Value CSR. Otherwise, the Value field holds the current raw value setting.
		-----	[2 ... 4]	reserved
		One_Push	[5]	If 1, a one push operation was in progress.
		ON_OFF	[6]	0: The feature was disabled, ignore the value 1: The feature was enabled
		A_M_Mode	[7]	0: The feature was in manual control mode 1: The feature was in auto control mode
		--	[8 ... 19]	Reserved
		Value	[20 ... 31]	Value of the feature
4	Gain Absolute Value CSR [4 Bytes]	**Field**	**Bit**	**Description**
		Abs Value	[0 ... 31]	Floating point value with IEEE/real*4 format Unit: dB
8	Shutter CSR [4 bytes]	Content of the DCAM Shutter CSR Same layout as the GAIN CSR		
12	Shutter Absolute Value CSR [4 Bytes]	**Field**	**Bit**	**Description**
		Abs Value	[0 ... 31]	Floating point value with IEEE/real*4 format Unit: sec
16	Gamma CSR [4 bytes]	Content of the DCAM Gamma CSR Same layout as the Gain CSR		
20	Gamma Absolute Value CSR [4 Bytes]	**Field**	**Bit**	**Description**
		Abs Value	[0 ... 31]	Floating point value with IEEE/real*4 format Unit: dB

<table>
<tr><td rowspan="10">24</td><td rowspan="10">White Balance CSR
[4 Bytes]</td><td colspan="3">Content of the DCAM White Balance CSR</td></tr>
<tr><td>Field</td><td>Bit</td><td>Description</td></tr>
<tr><td>Presence_Inq</td><td>[0]</td><td>Presence of this feature
If 0, the DCAM feature is not available and all of its values should be ignored</td></tr>
<tr><td>Abs_Control</td><td>[1]</td><td>Absolute control mode
If 1, the DCAM feature is in absolute control mode and the current value can be read from the Absolute Value CSR. Otherwise, the Value field holds the current raw value setting.</td></tr>
<tr><td>-----</td><td>[2 ... 4]</td><td>reserved</td></tr>
<tr><td>One_Push</td><td>[5]</td><td>If 1, a one push operation was in progress.</td></tr>
<tr><td>ON_OFF</td><td>[6]</td><td>0: The feature was disabled, ignore the value
1: The feature was enabled</td></tr>
<tr><td>A_M_Mode</td><td>[7]</td><td>0: The feature was in manual control mode
1: The feature was in auto control mode</td></tr>
<tr><td>U_Value /
B_Value</td><td>[8 ... 19]</td><td>U value / B value</td></tr>
<tr><td>V_Value /
R_Value</td><td>[20 ... 31]</td><td>V_Value / R_Value</td></tr>
<tr><td rowspan="2">28</td><td rowspan="2">White Balance
Absolute Value CSR
[4 Bytes]</td><td>Field</td><td>Bit</td><td>Description</td></tr>
<tr><td>Abs Value</td><td>[0 ... 31]</td><td>Floating point value with IEEE/real*4 format
Unit: K</td></tr>
<tr><td>32</td><td>Brightness CSR
[4 bytes]</td><td colspan="3">Content of the DCAM Brightness CSR
Same layout as the Gain CSR</td></tr>
<tr><td rowspan="2">36</td><td rowspan="2">Brightness Absolute Value CSR
[4 Bytes]</td><td>Field</td><td>Bit</td><td>Description</td></tr>
<tr><td>Abs Value</td><td>[0 ... 31]</td><td>Floating point value with IEEE/real*4 format
Unit: %</td></tr>
<tr><td>40</td><td>Chunk GUID
[16 Bytes]</td><td colspan="3">911C8982 - 1C0F - 11D8 - 8AF0 - 00105A5BAE55</td></tr>
<tr><td>56</td><td>Chunk Length
[4 bytes]</td><td colspan="3">This chunk’s total length in bytes.</td></tr>
<tr><td>60</td><td>Inverted Chunk Length
[4 bytes]</td><td colspan="3">The bitwise complement of the chunk length.</td></tr>
</table>

5.7.5 CRC Checksum

The CRC Checksum feature adds a chunk to each image frame containing a 16 bit CRC checksum calculated using the Z-modem method. The CRC Checksum chunk is always the last chunk added to the image data stream and the chunk is always 32 bits in size. As shown in Figure 5-2, the checksum is calculated using all of the image data and all of the appended chunks except for the checksum itself.

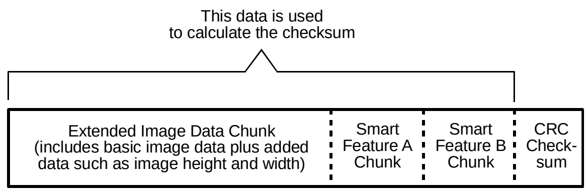


Figure 5-2: Data Used for the Checksum Calculation

The extended data stream feature (see Section 5.7.1) must be enabled in order to use the CRC Checksum feature or any of the other smart feature that adds information to the image data stream. Disabling the extended data stream feature switches off all smart features that add information to the image data stream.

The CRC Checksum feature will only work when the camera is set for video format 7.

Control and Status Register for the CRC Checksum Feature

Name	CRC_Checksum	
Address	See “Determining the Address of Smart Features CSRs” on page 5-4.	
CSR GUID	3B34004E - 1B84 - 11D8 - 83B3 - 00105A5BAE55	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
----	[1 ... 30]	Reserved
Enable (Read / write)	[31]	Enable / Disable this feature 0: Disable 1: Enable

CRC Checksum Chunk Layout

The CRC checksum is an exception to the normal chunk structure. The CRC chunk is always 32 bits wide and is always the last chunk appended to the image data. The lower 16 bits of the chunk are filled with the checksum and the upper 16 bits of the chunk are filled with zeros.

Bit	Description
[0 ... 7]	CRC Checksum low byte
[8 ... 15]	CRC Checksum high byte
[16 ... 23]	0x00
[24 ... 31]	0x00

Using the Checksum to Check the Data Integrity

When the checksum smart feature is enabled, the following two C functions can be used to check if an acquired frame contains a valid CRC checksum. The user must pass the acquired image buffer and the buffer's length in bytes to the CheckBuffer() function. The CheckBuffer() function uses the CRC16() function to calculate the checksum.

These two samples are intended to aid you in developing the code for your application. They are provided solely as examples.

```
/** \brief Calculates a 16 bit CRC checksum
  * \param pData Pointer to the data buffer
  * \param nbyLength Size of the buffer in bytes
  * \return The CRC checksum
  */
unsigned short CRC16(const unsigned char *pData, unsigned long nbyLength )
{
    unsigned long i, j, c, bit;
    unsigned long crc = 0;
    for (i=0; i<nbyLength; i++) {
        c = (unsigned long)*pData++;
        for (j=0x80; j; j>>=1) {
            bit = crc & 0x8000;
            crc <<= 1;
            if (c & j) bit^= 0x8000;
            if (bit) crc^= 0x1021;
        }
    }
    return (unsigned short) (crc & 0xffff);
}
```

```
/** \brief Verifies a frame buffer's CRC checksum
  * \param pData Pointer to the frame
  * \param nbyLength Size of frame in bytes
  * \return 1, if the check succeeds, 0 otherwise
  */
int CheckBuffer(const unsigned char* pData, unsigned long nbyLength )
{
       unsigned long nCurrentCRC, nDesiredCRC;
       /* Calculate the CRC checksum of the buffer. Don't take the last four bytes
           containing the checksum into account */
       nCurrentCRC = CRC16(pData, nbyLength - sizeof( unsigned long ) );
       /* Retrieve the desired CRC value from the data buffer */
       nDesiredCRC =  ((unsigned long*) pData)[ nbyLength / sizeof ( unsigned long ) - 1];
       /* Return TRUE if they are equal */
       return nCurrentCRC == nDesiredCRC;
}
```

5.7.6 Test Images

A102f cameras include a test image mode as a smart feature. The test image mode is used to check the camera's basic functionality and its ability to transmit an image via the video data cable. The test image mode can be used for service purposes and for failure diagnostics. In test mode, the image is generated with a software program and the camera's digital devices and does not use the optics, the CMOS pixel array, or the ADCs. Three test images are available on **A102f** cameras.

When a test image is active, the gain, brightness, and exposure time have no effect on the image.

ⓘ	The test image smart feature does not add information to the image data stream and can be enabled even when the extended data stream feature (see Section 5.7.1) is disabled. The test image feature will work when the camera is set for any valid video format.

Test Image one

Test image one is designed for use with monochrome, 8 bit output modes. As shown in Figure 5-3, test image one consists of rows with several gray scale gradients ranging from 0 to 255. Assuming that the camera is operating at full 1392 x 1040 resolution and is set for a monochrome, 8 bit output mode, when the test images are generated:

- Row 0 starts with a gray value of 0 for the first pixel,
- Row 1 starts with a value of 1 for the first pixel,
- Row 2 starts with a gray vale of 2 for the first pixel, and so on.

(If the camera is operating at a lower resolution when the test images are generated, the basic appearance of the test pattern will be similar to Figure 5-3, but the starting pixel values on each row will not be as described above.)

The mathematical expression for test image one is:

Gray Value = [x + y] MOD 256

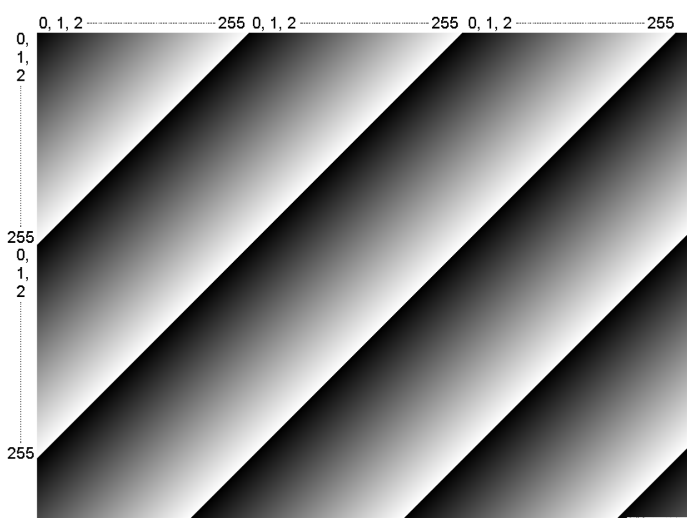


Figure 5-3: Test Image One

Test Image Two

Test image two is designed for use with monochrome, 16 bit output modes. Test image two consists of rows with several gray scale gradients ranging from 0 to 4095. Assuming that the camera is operating at full 1392 x 1040 resolution and is set for a monochrome, 16 bit output mode, when the test images are generated:

- Row 0 starts with a gray value of 0 for the first pixel, 1 for the second pixel, 2 for the third pixel, 3 for the fourth pixel, ...
- Row 1 starts with a gray value of 4 for the first pixel, 5 for the second pixel, 6 for the third pixel, 7 for the fourth pixel, ...
- Row 2 starts with a gray value of 8 for the first pixel, 9 for the second pixel, 10 for the third pixel, 11 for the fourth pixel, ...
- Row 3 starts with a gray value of 12 for the first pixel, 13 for the second pixel, 14 for the third pixel, 15 for the fourth pixel, ...

(If the camera is operating at a lower resolution when the test images are generated, the basic appearance of the test pattern will be similar to the description above, but the staring pixel values on each row will not be as described.)

The mathematical expression for test image two is:

Gray Value =[x + 4y] Mod 4096

ⓘ	Test image two is designed as a 16 bit test pattern. A 16 bit test pattern can not be properly displayed on standard 8 bit monitors. If you set the camera for 16 bit output and enable test pattern two, the pixel values in the test pattern will not be as described above when viewed on an 8 bit monitor. If you set the camera for 8 bit output and enable test pattern two, the pixel values in the test pattern will not be as described above.

Test Image Three

Test image three is similar to test image one but it is not stationary. The image moves by 1 pixel from right to left whenever a one-shot or a continuous-shot command signal is sent to the camera.

Control and Status Register for the Test Image Feature

Name	Test_Images	
Address	See “Determining the Address of Smart Features CSRs” on page 5-4.	
CSR GUID	2A411342 - C0CA - 4368 - B46E - EE5DEEBF0548	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
-----	[1 ... 7]	Reserved
Image_Inq_1 (Read only)	[8]	Presence of test image 1 0: Not Available 1: Available
Image_Inq_2 (Read only)	[9]	Presence of test image 2 0: Not Available 1: Available
Image_Inq_3 (Read only)	[10]	Presence of test image 3 0: Not Available 1: Available
Image_Inq_4 (Read only)	[11]	Presence of test image 4 0: Not Available 1: Available
Image_Inq_5 (Read only)	[12]	Presence of test image 5 0: Not Available 1: Available
Image_Inq_6 (Read only)	[13]	Presence of test image 6 0: Not Available 1: Available
Image_Inq_7 (Read only)	[14]	Presence of test image 7 0: Not Available 1: Available
-----	[15]	Reserved
Image_On (Read / write)	[16 ... 18]	0: No test image active 1: Test image 1 active 2: Test image 2 active 3: Test image 3 active
-----	[19 ... 31]	Reserved

5.7.7 Extended Version Information

A102f cameras include a register that contains version numbers for the camera’s internal software. For troubleshooting purposes, Basler technical support may ask you to read this register and to supply the results.

ⓘ	The extended version information smart feature does not add information to the image data stream and can be accessed even when the extended data stream feature (see Section 5.7.1) is disabled. The extended version feature will work when the camera is set for any valid video format.

Control and Status Register for the Extended Version Information Feature

Name	Extended_Version_Information	
Address	See “Determining the Address of Smart Features CSRs” on page 5-4.	
CSR GUID	2B2D8714 - C15E - 4176 - A235 - 6EF843D747B4	
Field	**Bit**	**Description**
Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
-----	[1 ... 7]	Reserved
Length	[8 ... 15]	Specifies the length in quadlets of the “string” field.
-----	[16 ... 31]	Reserved
Version_Info	[n Bytes]	An ASCII character string that includes the version numbers for the camera’s internal software. The length of this string field is equal to the number of quadlets given in the “length” field above.

ⓘ	The ASCII character string in the Version_Info field contains the camera’s “firmware ID” number. You can read the string to determine your camera’s firmware ID. The ID number’s position in the string is described in Section 1.1.

5.7.8 Lookup Table

The **A102f** camera has a sensor that reads pixel values at a 12 bit depth, however, the camera can be set to output pixel values at an 8 bit depth. When set for 8 bit output, the camera normally uses an internal process to convert the 12 bit pixel values from the sensor to the 8 bit values transmitted out of the camera. **A102f** cameras also include a smart feature that allows you to use a custom lookup table to map the 12 bit sensor output to 8 bit camera output rather than using the internal process.

The lookup table is essentially just a list of 4096 values, however, not every value is the table is actually used. If we number the values in the table from 0 through 4095, the table works like this:

- The number at location 0 in the table represents the 8 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 0.
- The numbers at locations 1 through 7 are not used.
- The number at location 8 in the table represents the 8 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 8.
- The numbers at locations 9 through 15 are not used.
- The number at location 16 in the table represents the 8 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 16.
- The numbers at locations 17 through 23 are not used.
- The number at location 24 in the table represents the 8 bit value that will be transmitted out of the camera when the sensor reports that a pixel has a value of 24.
- And so on.

As you can see, the table does not include an 8 bit output value for every pixel value that the sensor can report. So what does the camera do when the sensor reports a pixel value that is between two values that have a defined 8 bit output? In this case, the camera performs a straight line interpolation to determine the 8 bit value that it should transmit. For example, assume that the sensor reports a pixel value of 12. In this case, the camera would perform a straight line interpolation between the values at location 8 and location 16 in the table. The result of the interpolation would be reported out of the camera at an 8 bit depth.

Another thing to keep in mind about the table is that location 4088 is the last location that will have a usable 8 bit value associated with it. (Locations 4089 to 4095 are not used.) If the sensor reports a value above 4088, the camera will not be able to perform an interpolation. In cases where the sensor reports a value above 4088, the camera simply transmits the 8 bit value from location 4088 in the table.

Please look at page 5-25 and examine the layout of the control and status register for the lookup table smart feature. You will notice that the first two quadlets of the register include bits that allow you to check for this feature's presence and to enable or disable the feature. These initial two quadlets are followed by 4096 quadlets. The 4096 quadlets contain the values that make up the customized lookup table.

The advantage of the lookup table feature is that it allows the user to customize the response curve of the camera. The graphs below represent the contents of two typical lookup tables. The first graph is for a lookup table where the values are arranged so that the output of the camera increases linearly as the sensor output increases. The second graph is for a lookup table where the values are arranged so that the camera output increases quickly as the sensor output moves from 0 through 2048 and increases gradually as the sensor output moves from 2049 through 4096.

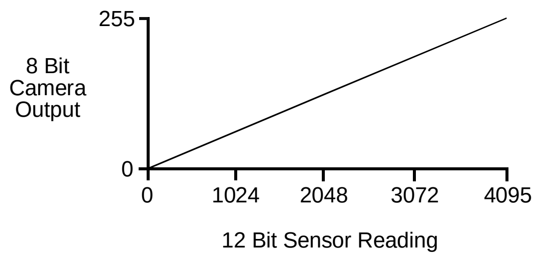


Figure 5-4: LUT with Values Mapped in a Linear Fashion

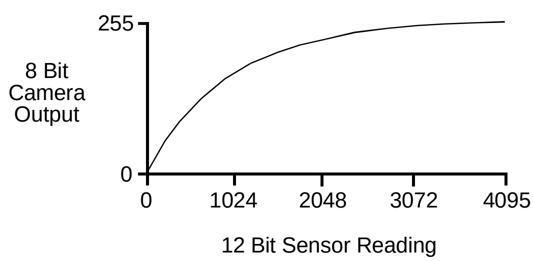


Figure 5-5: LUT with Values Mapped for Higher Camera Output at Low Sensor Readings

The lookup table smart feature does not add information to the image data stream and can be accessed even when the extended data stream feature (see Section 5.7.1) is disabled.

The gain and offset funtions remain active when the lookup table is used. Gain and offset are applied to the pixel values reported from the sensor **before** the pixel values are processed with the lookup table.

The lookup table feature will work when the camera is set for any valid video format.

When you enable the lookup table feature, a default lookup table is automatically loaded into the camera (see Section 5.7.9 for more information about the default table). If you want use your own customized lookup table you must:

1. Use the the look table feature Control and Status Register (CSR) to enable the lookup table feature.
2. Write the values for your customized lookup table to the CSR.

Control and Status Register for the Lookup Table Feature

Name	Lookup_Table		
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.		
CSR GUID	B28C667C - DF9D - 11D7 - 8693 - 000C6E0BD1B0		
Position	**Field**	**Bit**	**Description**
0	Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
	-----	[1 ... 30]	Reserved
	Enable (Read / write)	[31]	Enable / Disable this feature 0: Disable 1: Enable
4	In_Depth_Inq (Read only)	[0 ... 15]	Bit depth of the pixel data reported by the sensor.
	Out_Depth_Inq (Read only)	[16 ... 31]	Bit depth of the pixel data transmitted from the camera.
8	Quadlet_0 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 0. (The 8 LSBs of the quadlet carry the data for the field. The 24 MSBs are all zeros.)
12 ... 36	Quadlet_1 ... 7 (Read / write)	[7 Quadlets]	Not used. The values written in these quadlets will be ignored.
40	Quadlet_8 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 8.
44 ... 68	Quadlet_9 ... 15 (Read / write)	[7 Quadlets]	Not used. The values written in these quadlets will be ignored.
72	Quadlet_16 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 16.
76 ... 100	Quadlet_17 ... 23 (Read / write)	[7 Quadlets]	Not used. The values written in these quadlets will be ignored.
104	Quadlet_24 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 16.
• • •	• • •	• • •	• • •
16296	Quadlet_4072 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 16.

16300 ... 16324	Quadlet_4073 ... 4079 (Read / write)	[7 Quadlets]	Not used. The values written here will be ignored.
16328	Quadlet_4080 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 4080.
16332 ... 16356	Quadlet_4081 ... 4087 (Read / write)	[7 Quadlets	Not used. The values written here will be ignored.
16360	Quadlet_4088 (Read / write)	[1 Quadlet]	Defines the 8 bit value that will be transmitted from the camera when the 12 bit pixel value from the sensor is 4088.
16364 ... 16388	Quadlet_4089 ... 4095 (Read / write)	[7 Quadlets]	Not used. The value written here will be ignored.

Using the SFF Viewer to Upload a Lookup Table

The Configurator window in the Basler SFF Viewer (see Section 5.4) includes a drop down list that can be used to enable the lookup table feature. It also includes an *Upload* button that can be used to easily load a text file containing a customized lookup table into the camera. The file must be plain text and must be formatted correctly. The file must have 4096 lines with each line containing two comma-separated values. The first value on each line represents a 12 bit pixel reading from the sensor and the second value represents the corresponding 8 bit output that will be transmitted from the camera.

Not every value in the file is actually used. Only the values in line 0, line 8, line 16, line 24, etc. are used for the lookup process. However the file **must** contain 4096 lines with two comma separated values on each line. (The values on lines 1 through 7, 9 through 15, 17 through 23, etc. must be included in the text file even though they are ignored.)

The sample below shows part of a typical text file for a lookup table. Assuming that you have enabled the lookup table feature on your camera and used the *Upload* button to load a file similar to the sample into the camera:

- If the sensor reports that a pixel has a value of 0, the camera will output a value of 0.
- If the sensor reports that a pixel has a value of 8, the camera will output a value of 1.
- If the sensor reports that a pixel has a value of 16, the camera will output a value of 3.
- If the sensor reports that a pixel has a value of 4088, the camera will output a value of 255.

```
0,0
1,0
2,0
3,0
4,0
5,0
6,0
7,0
8,1
9,1
10,1
11,1
12,1
13,1
14,1
15,1
16,3
17,3
18,3
19,3
20,3
21,3

4082,254
4083,254
4084,254
4085,254
4086,254
4087,254
4088,255
4089,255
4090,255
4091,255
4092,255
4093,255
4094,255
4095,255
```

Figure 5-6: Sample Text File for Use With *Upload* Button

5.7.9 Lossless Compression

The **A102f** includes a feature that allows lossless compression of the 12 bit output from the camera's sensor to 8 bit output transmitted from the camera. The basis for the lossless compression feature is a mathematical process that compresses the 12 bit output from the sensor to 8 bits by removing information characterized as noise. Because this method removes only the portion of the sensor's output that represents noise, no image information is lost and the resulting images have an extended dynamic range compared to normal 8 bit images.

Lossless compression is implemented on the **A102f** by means of a lookup table. To enable lossless compression, simply enable the lookup table feature as described in Section 5.7.8. When you enable the lookup table feature, a default table is automatically loaded into the camera. The values in the default lookup table were determined using the lossless compression technique. So if you operate the camera with the lookup feature enabled and the default table loaded, the camera will output 8 bit data via lossless compression.

5.7.10 Trigger Flag and Trigger Counter

A102f cameras include a trigger flag and trigger counter feature. The trigger counter increments by one each time an image capture is triggered regardless of whether the trigger is internal (one shot or continuous shot commands) or is external (hardware or software trigger). Triggers that occur when the camera is not ready are discarded and not counted. The trigger counter wraps to zero after 65535 is reached.

If one or more triggers has been detected since the last time the Trigger_Flag field was read, the trigger flag is set to one. The flag self clears with each read access.

Writes to the Trigger_Count or Trigger_Flag fields are ignored.

The counter field or the flag field can be polled by your camera control software to detect the receipt of a trigger signal by the camera. The camera control software can react synchronously to each trigger signal received. By using the results of the polling to know when a trigger signal is received by the camera, you can eliminate the need for a hard wired signal from the hardware device that is issuing the trigger. Keep in mind that your degree of precision depends on your polling frequency and the 1394 bus latency.

ⓘ	The trigger flag and trigger counter smart feature does not add information to the image data stream and can be accessed even when the extended data stream feature (see Section 5.7.1) is disabled. The trigger flag and counter feature is always enabled regardless of the video format.

Control and Status Register for the Trigger Flag and Counter Feature

Name	Trigger_Flag_and_Counter		
Address	See “Determining the Address of Smart Features CSRs” on page 5-4.		
CSR GUID	16C31A78 - 3F75 - 11D8 - 94EC - 00105A5BAE55		
Position	**Field**	**Bit**	**Description**
0	Presence_Inq (Read only)	[0]	Presence of this feature 0: Not Available 1: Available
	-----	[1 ... 15]	Reserved
	Trigger_Count (Read only)	[16 ... 31]	The trigger counter increments by one each time an image capture is triggered. The counter is reset at power on or when an initialize command is issued.
4	-----	[0 ... 30]	Reserved
	Trigger_Flag (Read only)	[31]	The flag is set to 1 by each trigger. It is cleared by a read access to this register.

5.7.11 Output Port Configuration

A102f cameras are equipped with four physical output ports designated as Output Port 0, Output Port 1, Output Port 2, and Output Port 3. The output port configuration feature can be used to change the assignment of camera output signals (such as Integrate Enabled and Trigger Ready) to the physical output ports.

As shown on pages 5-31 and 5-32, there is a control and status register (CSR) for each physical output port. The Source_Select field in each register is used to assign a camera signal to the associated output port. For example, the Source_Select field in the register for Output_Port_0 is used to assign a camera output signal to physical output port 0.

Each physical output port can be unassigned or it can have one and only one camera output signal assigned to it.

You can assign a camera output signal to more than one physical output port. For example, the Trigger Ready signal could be assigned to both physical output port 0 and physical output port 1.

The Source_Select field can also be used to designate an output port as “user set.” If an output port is designated as user set, its state can be set to high or low by using the User_Setting field in the CSR for the port.

The Invert field can be used to invert the signal before it is applied to the output port and the Monitor field can be used to check the current state of the output port.

When using the output port configuration feature, you should follow this sequence:

1. Read the Presence_Inq field and the Source_Select_Inq field for the physical port you want to work with. Determine whether the port configuration feature is available for the port and if the source for the port is selectable.
2. Use the Source_Select field to select a source for the output port.
(If you select “User set” as the source, the state of the physical output port may change when you set the bits in the Source_Select field. This is an artifact of the camera design.)
3. Check the Monitor_Inq, Invert_Inq, and User_Setting_Inq fields. The state of these fields will tell you if the Monitor, Invert, and User_Setting fields are available. (The availability of the Monitor, Invert, and User_Setting fields will vary depending on the selected source.)
4. If the Invert field is available, set the field as desired.
5. If you selected “User set” as the source, use the User_Setting field to set the state of the output as desired.
6. If the Monitor field is available, use the field as desired to check the current state of the output.

The output port configuration smart feature does not add information to the image data stream and can be accessed even when the extended data stream feature (see Section 5.7.1) is disabled.

The output port configuration feature is always enabled regardless of the video format.

By default, the Integrate Enabled signal is assigned to physical output port 0 and the Trigger Ready Signal is assigned to physical output port 1.

Control and Status Registers for the Output Port Configuration Feature

Name	Output_Port_0_Configuration		
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.		
CSR GUID	5A889D7E - 41E5 - 11D8 - 845B - 00105A5BAE55		
Position	**Field**	**Bit**	**Description**
0	Presence_Inq (Read only)	[0]	Presence of the output port 0 configuration feature 0: Not Available 1: Available
	Monitor_Inq (Read only)	[1]	Presence of the monitor field 0: Not Available 1: Available
	Invert_Inq (Read only)	[2]	Presence of the invert field 0: Not Available 1: Available
	User_Setting_Inq (Read only)	[3]	Presence of the user setting field 0: Not Available 1: Available
	-----	[4 ... 26]	Reserved
	Source_Select (Read / write)	[27 ... 31]	Write a value to select a source signal for output port 0: 0: Integrate Enabled signal 1: Trigger Ready signal 3: User set (state can be set with the User_Setting field described below) 4: Strobe
4	Source_Select_Inq (Read only)	[0 ... 31]	If bit *n* is set, then value *n* is valid for use in the Source_Select field. For example, if bit 0 is set, then 0 is a valid value for use in the Source_Select field. If bit 1 is set, then 1 is a valid value for use in the Source_Select field. Etc.
8	-----	[0 ... 30]	Reserved
	Monitor (Read only)	[31]	Shows the current state of the output: 0: Low (non-conducting) 1: High (conducting)
12	-----	[0 ... 30]	Reserved
	Invert (Read / write)	[31]	Enables signal inversion: 0: Do not invert 1: Invert
16	-----	[0 ... 30]	Reserved
	User_Setting (Read / write)	[31]	If the Source_Select field is set to "user set", this field sets the state of the output: 0: Low (non-conducting) 1: High (conducting) (Sets the state of the output <u>before</u> the inverter.)
20	-----	[0 ... 31]	Reserved

Name	Output_Port_1_Configuration		
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.		
CSR GUID	949D820A - 4513 - 11D8 - 9EB1 - 00105A5BAE55		
Position	**Field**	**Bit**	**Description**
0	Same as port 0.		
4	Same as port 0.		
8	Same as port 0.		
12	Same as port 0.		
16	Same as port 0.		
20	Same as port 0.		

Name	Output_Port_2_Configuration		
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.		
CSR GUID	C14E5072 - 4513 - 11D8 - 81F3 - 00105A5BAE55		
Position	**Field**	**Bit**	**Description**
0	Same as port 0.		
4	Same as port 0.		
8	Same as port 0.		
12	Same as port 0.		
16	Same as port 0.		
20	Same as port 0.		

Name	Output_Port_3_Configuration		
Address	See "Determining the Address of Smart Features CSRs" on page 5-4.		
CSR GUID	949D820A - 4513 - 11D8 - 9EB1 - 00105A5BAE55		
Position	**Field**	**Bit**	**Description**
0	Same as port 0.		
4	Same as port 0.		
8	Same as port 0.		
12	Same as port 0.		
16	Same as port 0.		
20	Same as port 0.		

5.8 Customized Smart Features

The Basler **A102f** has significant processing capabilities and Basler can accommodate customer requests for customized smart features. A great advantage of the smart features framework is that it serves as a standardized platform for parameterizing any customized smart feature and for returning the results from the feature.

The Basler camera development team is ready and able to handle requests for customized smart features. The cost to the customer for adding a customized smart feature to the **A102f** will depend on the complexity of algorithm, software, and firmware development, of incorporating the feature within the smart features framework, and of testing to ensure that the feature meets specifications. Please contact your Basler sales representative for more details about customized smart features.

6 Mechanical Considerations

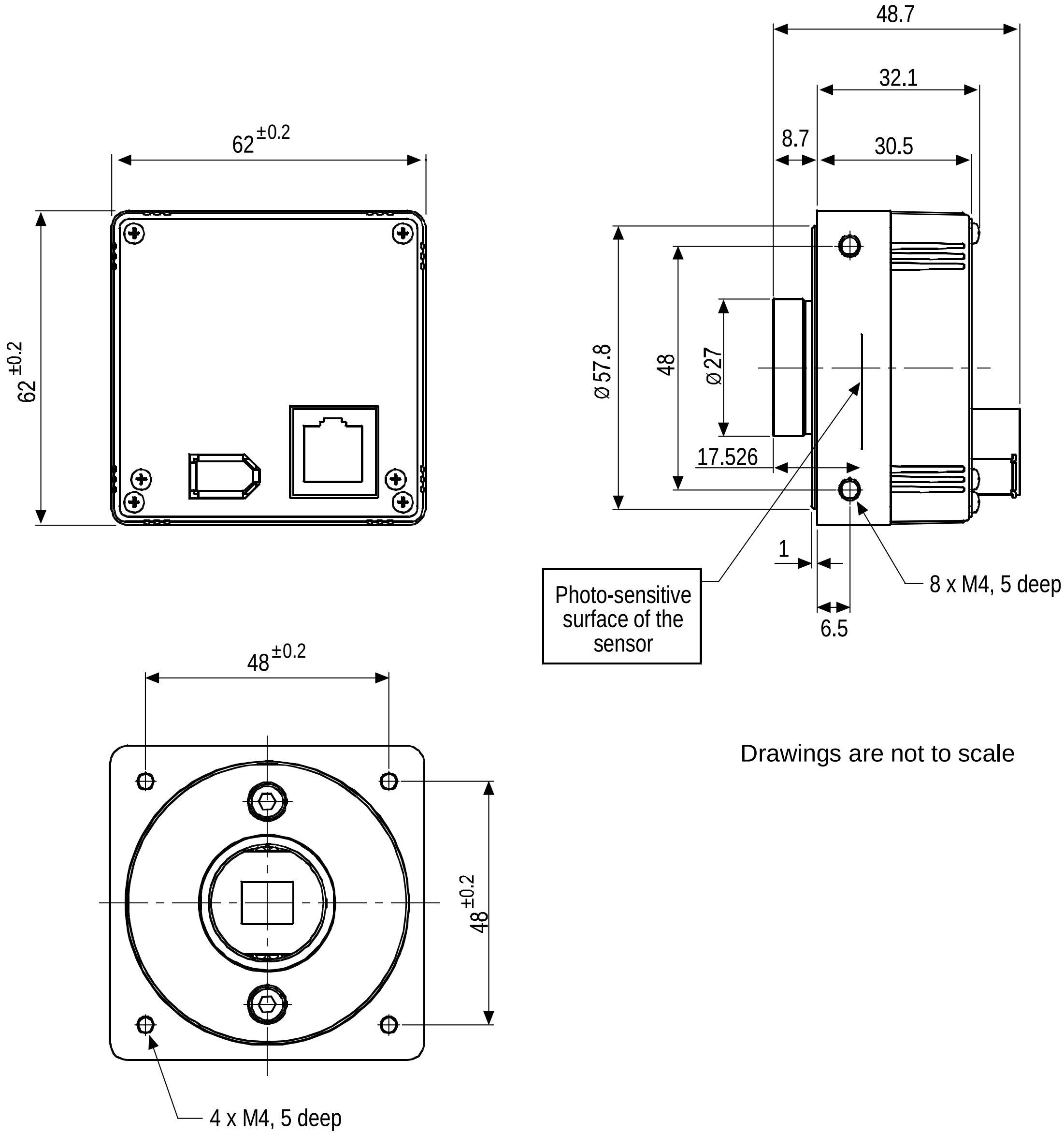
6.1 Camera Dimensions and Mounting Facilities

The **A102f** camera housing is manufactured with high precision. Planar, parallel, and angular sides guarantee precise mounting with high repeatability.

The **A102f** camera is equipped with four M4 mounting holes on the front and two M4 mounting holes on each side as indicated in Figure 6-1.

Caution!

To avoid collecting dust on the sensor, mount a lens on the camera immediately after unpacking it.



Drawings are not to scale

Figure 6-1: **A102f** Mechanical Dimensions (in mm)

6.2 Positioning Accuracy of the Sensor Chip

The positioning accuracy of the sensor chip in the horizontal and vertical directions is as shown in Figure 6-2. Rotational accuracy is also shown in the figure. Reference position is the center of the camera housing.

Since the translatory and the rotational positioning tolerances depend on each other, the worst case of maximum rotational and horizontal/vertical mis-positioning can not occur at the same time.

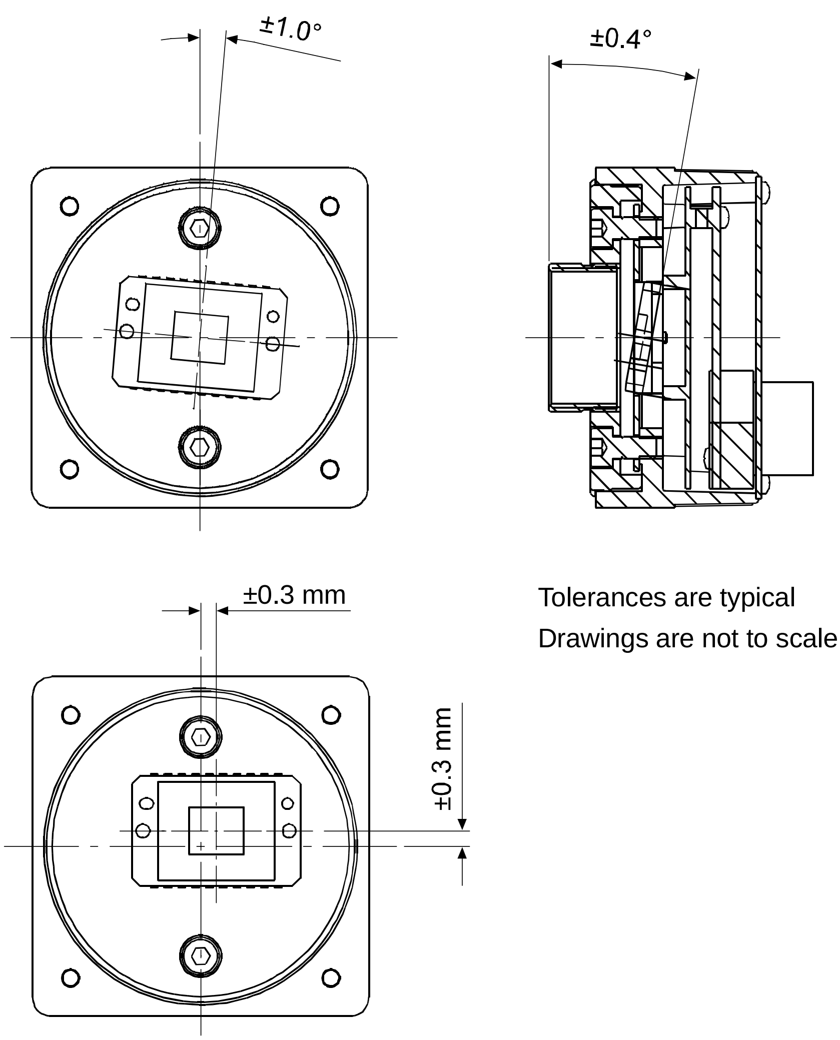


Figure 6-2: Positioning Accuracy of the Sensor Chip

6.3 Maximum Lens Thread Length on the A102fc

A102fc cameras are normally equipped with a C-mount lens adapter that contains an internal IR cut filter. As shown in Figure 6-3, the thread length of the C-mount lens must used on the camera must be less than 7.5 mm. If a lens with a longer thread length is used, the IR cut filter will be damaged or destroyed and the camera will no longer operate.

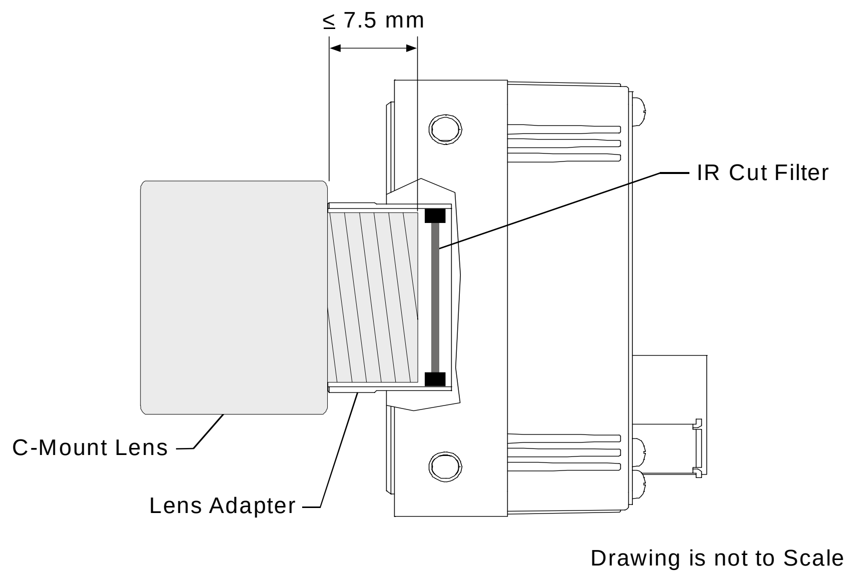


Figure 6-3: Maximum Lens Thread Length on A102fc Cameras

	**Caution!** **A102fc** color cameras are equipped with an IR cut filter mounted in of the lens adapter. The location of the filter limits the thread length of the lens that can be used on the camera. The thread length on your lens must be less than 7.5 mm. If a lens with a longer thread length is used, the camera will be damaged and will no longer operate.

Revision History

Doc. ID Number	Date	Changes
DA00063001	24 October 2003	Initial release of this document.
DA00063002	25 February 2004	Second draft for revision two of the manual. Added Section 1.1 describing document applicability. Updated Sections 2.4, 3.2.5, 3.3, and 3.4 to reflect the new input and output port configuration options. Added Section 3.2.4 describing the new software trigger feature. Updated the numbers in Figure 3-6 to reflect minor changes in the exposure timing. Updated the note box on page 3-19 to show restrictions when setting the AOI on color cameras. Added Section 3.6.1 describing changing the AOI on-the-fly. Updated the description of the color filter ID in Section 3.8.2. Added Section 3.9.2 regarding bit depth selection on color cameras. Added Sections 3.10 and 3.11 describing the new Strobe Control Output Signal and Parallel IO Control features. Added the Raw 16 color ID description on page 3-34. Updated the register descriptions in Section 4.4 to reflect new and revised features. Added Section 5 describing the new smart features capabilities. Revised Figure 6-1 to reflect a minor dimension change.
DA00063003	22 March 2004	Corrected errors in the description of the Control and Status Registers for PIO Control on page 4-14. Corrected errors in the description of the Control and Status Registers for the Output Port Configuration smart feature on page 5-31.

Feedback

Your feedback will help us improve our documentation. Please click the link below to access an online feedback form. Your input will be greatly appreciated.

http://www.baslerweb.com/umfrage/survey.html

Index

R

S

T

V

W

Y