Operating and user manual

Q-12 CoaXPress series

rev2.1

REVISION HISTORY

Revision	Page	Description of revision
2.0	-	Complete revision from the 1.x revision to include the Q-12A180 CXP issue 2.0 camera.
2.1	2	Revision history added.
	28	Added remark about defect pixels and max frame rate in binned mode.
	58/59	Added section 7.17.23 How to transfer a low frequency flat field calibration set from one camera to another one. Sections 7.17.24 to 7.17.27 have been updated to describe the functionality better.
	68	Added APPENDIX B: C++ Example, transfer an LF FF calibration set.

ABOUT ADIMEC

Adimec designs, manufactures, and markets high performance industrial cameras for equipment manufacturers in:

- Machine Vision
- Healthcare
- Global Security

Our high resolution cameras offer a unique combination of excellence in image quality, speed, and reliability. With optimized functionality for the needs of specific applications, Adimec cameras exceed general purpose.

Adimec is a reliable partner with a focus on establishing long term relationships through a worldwide network of highly qualified engineers.

Adimec aligns its roadmap in close cooperation with industry leaders and monitors the market for the latest technology to continuously provide innovative cameras that enhance our customers' competitiveness. With our capabilities, modular designs, process control and commitment to partnership, we can tailor to the exact solution required in a short time to market and with low risk.

Adimec Advanced Image Systems B.V.
PO Box 7909
5605 SH Eindhoven
The Netherlands
Phone: +31 (0)40 235 3900
E-mail: SalesEU@adimec.com

Adimec Japan
2-10-3-103, Narimasu, Itabashi-Ku
175-0094 Tokyo, Japan
Phone: +81 (3) 5968 8377
Fax: +81 (3) 5968 8388
E-mail: SalesJP@adimec.com

Adimec USA
130 New Boston Street, Suite 204
Woburn, MA 01801, USA
Phone: +1 (781) 279 0770
Fax: +1 (781) 279 0571
E-mail: SalesUS@adimec.com

Adimec Asia/Pacific
60 Paya Lebar Road
#08-07 Paya Lebar Square
Singapore 409051
Phone: +65 6334 1236
Fax: +65 6334 1436
E-mail: SalesAP@adimec.com

TABLE OF CONTENTS

1 INTRODUCTION

The Quartz CMOS camera described in this manual provides 12 Mpx resolution through a CoaXPress interface. The sensor has a global shutter architecture, combined with CCD like image quality and reliability. Unlike competitors who are offering general purpose cameras, our products are developed with the specific needs of OEMs and their applications in mind.

1.1 Product highlights

- 12 Megapixel up to 187 fps
- True Global Shutter CMOS
- Column based DSNU and PRNU corrections
- Low frequency flat field correction (issue 2 only)
- Configurable 1,2 or 4 lane CXP3 - 6 interface
- M12 I/O connector
- CoaXPress V1.1.1 compliant
- CoaXPress V1.0 compatible

1.2 About this manual

This manual describes the Q-12 CoaXPress camera series:

Product name	**Issue**	**Part number**
Q-12A180-Fm/CXP-6-1.0	1.0	188160
Q-12A180-Fc/CXP-6-1.0	1.0	188170
Q-12A180-Fm/CXP-6-2.0	2.0	197360
Q-12A180-Fc/CXP-6-2.0	2.0	197370

Camera or issue specific functionality or specifications will be clearly indicated. Especially between issue 1.0 and issue 2.0 there are significant differences in available functionality.

Practical tips or notes are indicated by the “**NOTE:**” sign.

1.3 List of frequently used abbreviations

Abbreviation	Full expression
ASP	Adimec Service Port
CMOS	Complementary Metal Oxide Semiconductor
CRC	Cyclic Redundancy Check
CXP	CoaXPress
DIN	Deutsches Institut für Normung
DPC	Defect Pixel Correction
DSNU	Dark Signal Non-Uniformities
EC	European Commission
EEC	European Economic Community
ESD	Electro-Static Discharge
GenAPI	GenICam Application Programming Interface
GenICam	Generic Interface for Cameras
GUI	Graphical User Interface
HDR	High Dynamic Range
I/O	Input/Output
JIIA	Japan Industrial Imaging Association
LED	Light Emitting Diode
LF FFC	Low Frequency Flat Field Correction
LUT	Look-up Table
PoCXP	Power over CoaXPress
PRNU	Pixel Response Non-Uniformities
RO	Read Only
ROI	Region of Interest
RW	Read and Write
SFNC	Standard Features Naming Convention
USB	Universal Serial Bus
WEEE	Waste Electrical and Electronic Equipment
WO	Write Only
XML	Extensible Markup Language

1.4 Waste Electrical and Electronic Equipment

With regard to waste electrical and electronic equipment (WEEE), Adimec wishes to follow the Directive 2002/96/EC of the European Parliament and of the Council. The purpose of this Directive is, as a first priority, the prevention of waste electrical and electronic equipment (WEEE), and in addition, the reuse, recycling and other forms of recovery of such wastes so as to reduce the disposal of waste. It also seeks to improve the environmental performance of all operators involved in the life cycle of electrical and electronic equipment, e.g. producers, distributors and consumers and in particular those operators directly involved in the treatment of waste electrical and electronic equipment.

Separate collection for electronic equipment in your area is recommended in order to minimize the disposal of WEEE as unsorted municipal waste and to achieve a high level of separate collection of WEEE.

1.5 Liability

Adimec prepares this manual with the greatest care. Please inform Adimec of any inaccuracies or omissions. Adimec Advanced Image Systems B.V. cannot be held responsible for any technical or typographical errors and reserves the right to make changes to the product and manuals without prior notice. Adimec Advanced Image Systems B.V. makes no warranty of any kind with regard to the material contained within this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Adimec Advanced Image Systems B.V. shall not be liable or responsible for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Layout, composing and editorial staff: Adimec Advanced Image Systems B.V.
Illustrations: Adimec Advanced Image Systems B.V.

All correspondence regarding copyrights and translations should be directed to:

Adimec Advanced Image Systems B.V.
PO Box 7909
5605 SH Eindhoven
The Netherlands
Tel: +31 (0)40 235 3900
E-mail: Support@adimec.com
URL: www.adimec.com

2 SAFETY PRECAUTIONS

NOTE: A CMOS sensor camera is a sensitive device. Please read the following precautions carefully before continuing unpacking or operating the camera.

NOTE: It is advised to unpack and handle the camera in a clean ESD protected working area.

NOTE: It is advised to read the whole manual before using the camera.

NOTE: Always keep the sensor cap in place as long as no lens is attached.

NOTE: Remove the lens cap just before the lens is screwed on the camera. It is advised to perform this operation in a clean room or clean bench.

NOTE: Never touch the CMOS sensor surface. The cover glass is easily damaged and the CMOS sensor can be damaged by ESD.

NOTE: In case the camera is used as a subsystem, it is advised to include the text of this chapter in the assembly documents of the main system.

2.1 Cleaning of the CMOS sensor

The cleaning of a CMOS sensor is a difficult task with a high risk on permanent damage to the camera.

NOTE: It is advised to prevent cleaning the CMOS sensor as much as possible.

NOTE: Damage of the CMOS sensor due to scratches on the cover glass or ESD is not covered by warranty!

All cameras are checked for cleanliness in our factory before shipment.

Proper handling instructions during system assembly can prevent the CMOS sensor from getting contaminated.

Should cleaning of the CMOS sensor be necessary, please refer to Appendix A: CMOS Sensor cleaning instructions.

2.2 Camera housing

Thermal interfacing
The actual housing temperature achieved depends on the thermal configuration of the camera and the system in the end-user application. Provisions as to guarantee maximum housing temperature are therefore a responsibility of the end-user.

NOTE: The housing temperature should not exceed +50° Celcius.

NOTE: Mount the camera on a substantial (preferably metal) body that can act as a heat sink.

NOTE: Create airflow over the camera e.g. by using a fan.

Cleaning of the camera housing

The camera housing should NEVER be immersed in water or any other fluid. For cleaning, only use a light moist tissue.

Connectors

Take care of the connectors during handling of the camera. Connectors should not be damaged. Prevent the entry of foreign objects or dirt into the connectors, as this will result in unreliable operation or damage.

Mounting screws

M4 screws should be used with a maximum screw depth of 5 mm. The recommended tightening torque is 188 cNm. Take notice of the maximum length of the screws that may be used for mounting the camera. Using screws too long can cause damage to the camera.

2.3 Camera repair and Warranty

Repair, modification and replacement of parts shall be done only by Adimec to maintain compliance with the directive 89/336/EEC electromagnetic compatibility, directive 72/23/EEC low voltage directive and the international standards.

For repair and warranty claims contact your local dealer or the business offices in your region. The minimum information we need to know for a repair request or warranty claim are the camera serial number and a detailed failure description.

In case the camera needs to be returned to investigate the repair options or grant your warranty claim you will receive a Return Material Authorization (RMA) number. Please use this RMA-number to ship the camera to Adimec. Cameras without RMA number will be rejected.

Once the camera is arrived at Adimec the camera will be investigated to proof possible repair or grant your warranty claim. In case of repair the repair costs will be quoted. After your approval of the repair cost camera will be repaired and returned.

3 QUICK START GUIDE

The procedure to obtain the first images from the camera depends to some extend on the frame grabber brand or type you use.

A general quick start guide is therefore difficult to provide.

NOTE: The Adimec support department has a couple of frame grabbers available. Inform at support@adimec.com which frame grabber specific quick install guides are available or can be created.

The general steps to collect your first images are:

1. Mount a lens on the camera.
2. Connect the CXP cables to the camera.
3. Connect the CXP cables to the frame grabber.
4. Start the PC.
5. Go through the frame grabber specific procedure to configure your frame grabber. Some frame grabbers will automatically identify the camera while for others the right configuration file has to be loaded.
6. Use the capture software supplied by your frame grabber manufacturer to start acquiring images.

For a correct configuration in step 5, the factory default settings might be required. The relevant parameters are listed in Table 3-1.

Table 3-1: Factory default settings for the Q-12 CoaXPress camera series.

Parameter	Value
Revision	1.1.1
ConnectionConfig	CXP3_X1
ConnectionConfigDefault	CXP6_X4
PixelFormat (mono)	Mono10
PixelFormat (bayer)	BayerGB10
AcquisitionMode	Continuous
ExposureMode	Timed

NOTE: Discovery always occurs on CXP3_X1. Most frame grabbers will change the ConnectionConfig feature automatically to the ConnectionConfigDefault value after discovery.

4 INTERFACES

In this chapter the electrical interfaces as well as the optical interface are described.

4.1 Connector overview

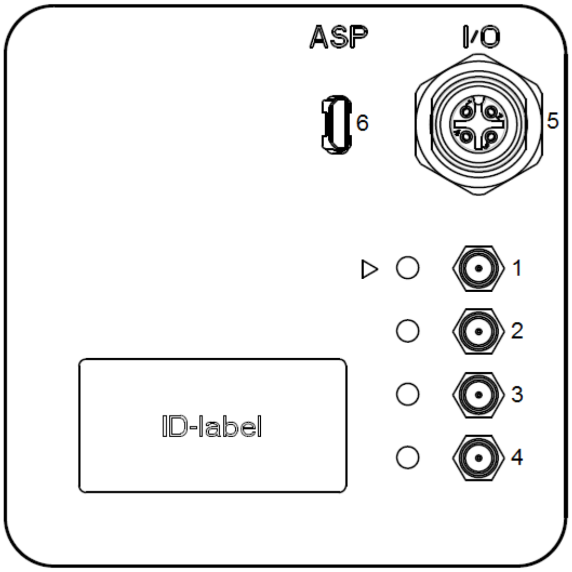


Figure 4-1: Camera back overview.

Table 4-1: Camera connections.

Connector number	*Connector type*	*Description*	*Mating connector*
1	DIN 1.0/2.3	CXP connection 0 (Master connection) Video, power, control, triggering	DIN 1.0/2.3[1]
2	DIN 1.0/2.3	CXP connection 1 (Extension, Dual)[2] Video	DIN 1.0/2.3[1]
3	DIN 1.0/2.3	CXP connection 2 (Extension, Quad)[3] Video	DIN 1.0/2.3[1]
4	DIN 1.0/2.3	CXP connection 3 (Extension, Quad)[3] Video	DIN 1.0/2.3[1]
5	Binder M12 type 09-3432-216-04	I/O connector	Binder M12 type 79-3529-13-04
6	Micro USB, type B (Socket)	Adimec Service Port	Micro USB, type B (Plug)

Note:
[1] Single DIN cables and multi DIN cables supported.
[2] Dual: CXP connection 0, 1 are used.
[3] Quad: CXP connection 0, 1, 2, 3 are used.

4.2 Electrical interfaces

4.2.1 Power and CoaXPress connectors

The CoaXPress interface supports communication in two directions. Power, control data and trigger signals are transferred from the frame grabber to the camera and video data is transferred from the camera to the frame grabber. The function of each connector is listed in Table 4-1. The CXP configurations that are supported by the camera are listed in Table 4-2.

Table 4-2: The supported CXP configurations.

Compliance Labeling	*Max Bit Rate per Coax*	*Nr. connected Cables*	*Maximum cable length (Belden 1694A)*
CXP-3 DIN 1	3.125 Gb/s	1	105 m
CXP-3 DIN 2	3.125 Gb/s	2	105 m
CXP-3 DIN 4	3.125 Gb/s	4	105 m
CXP-6 DIN 1	6.250 Gb/s	1	45 m
CXP-6 DIN 2	6.250 Gb/s	2	45 m
CXP-6 DIN 4	6.250 Gb/s	4	45 m

NOTE: Always connect CXP connector 0 as it supplies the camera with power according to the Power over CoaXPress standard (PoCXP, max 13W).

NOTE: CXP connector 0 is identified by the triangular arrow symbol on the camera housing.

For a complete description of the CoaXPress interface standard please refer to the CoaXPress specification that can be downloaded from http://jiia.org/en.

4.2.2 CoaXPress status LEDs

Next to the CoaXPress connector a multi-color LED status indicator is present. The meaning for each LED indication is shown in Table 4-3, while Table 4-4 lists the frequencies of the fast and slow flashes.

Table 4-3: The meaning of the LED status indicator explained.

LED indication		*Camera / interface status*
	Off	No power
	Solid orange	System booting
	Slow pulse orange	Device / Host connected, waiting for event (e.g. trigger, exposure pulse)
	500 ms red pulse. In case of multiple errors, there shall be at least two green fast flash pulses before the next error is indicated.	Error during data transfer (e.g. CRC error, single bit error detected)
	Fast flash red	System error (e.g. internal error)
	Solid green	Device / Host connected, but no data being transferred
	Fast flash green	Device / Host connected, data being transferred
	Fast flash alternate green / orange. Shown for a minimum of 1 s even if the connection detection is faster	Connection detection in progress, PoCXP active
	Slow flash alternate green / orange	Connection test packets being sent
	Slow flash alternate red / green / orange	Compliance test mode enabled

Table 4-4: LED indicator flash frequency

Flash indication	*Frequency*
Fast flash	12.5 Hz
Slow flash	0.5 Hz
Slow pulse	1 Hz

4.2.3 Adimec Service Port

The Adimec Service Port (ASP) interface is available for firmware uploads to the camera.

USB driver XR21x141x is required to support communication via the ASP port.

The USB connection will be available as a virtual COM port.

4.2.4 I/O Connector

A trigger input and flash strobe output are available at the I/O connector. The input and output are galvanic isolated from the internal camera electronics by means of an optocoupler (Avago ACPL-M50L).

The connector layout is shown in Figure 4-2 with the pin description in Table 4-5.

Table 4-6 lists the recommended resistor values to be used in the termination circuitry to achieve the recommended currents. In Figure 4-3 the recommended termination circuitry is shown.

NOTE: Opto-couplers require a certain response time to turn on or to turn off, i.e. to switch from the non-conductive state to the conductive state or the other way around. This response time depends on the used electronic circuit. For the “trigger in” it can be as short as < 0.5 µs. For the “flash strobe out” it can be as short as < 2 µs. In practice this response time means that when a trigger is applied there will be a delay between the rising edge of the trigger and the reaction of the camera.

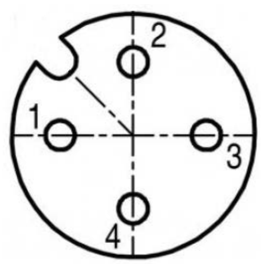


Figure 4-2: Female I/O connector layout viewed from the backside of the camera

Table 4-5: Connector pinning

Pin	*Signal name*	*Type*	*Level*	*Description*
1	Trigger in	Input	10..20 mA (10 mA recommended)	Anode of opto-coupler*
2	Flash strobe out	Output	2.5 mA Recommended	Open collector of opto-coupler photo transistor.
3	Flash strobe return	Output	Isolated gnd	Emitter of opto-coupler photo transistor.
4	Trigger return	Input	Isolated gnd	Cathode of opto-coupler

* serial resistors 2x 220 Ω inside camera.

Table 4-6: Recommended resistor values

Vext [V]	*R1E ext [Ω]*	*R2E ext [Ω]*
3.3	1000	Do not apply
5.0	2000	0
12	4700	470

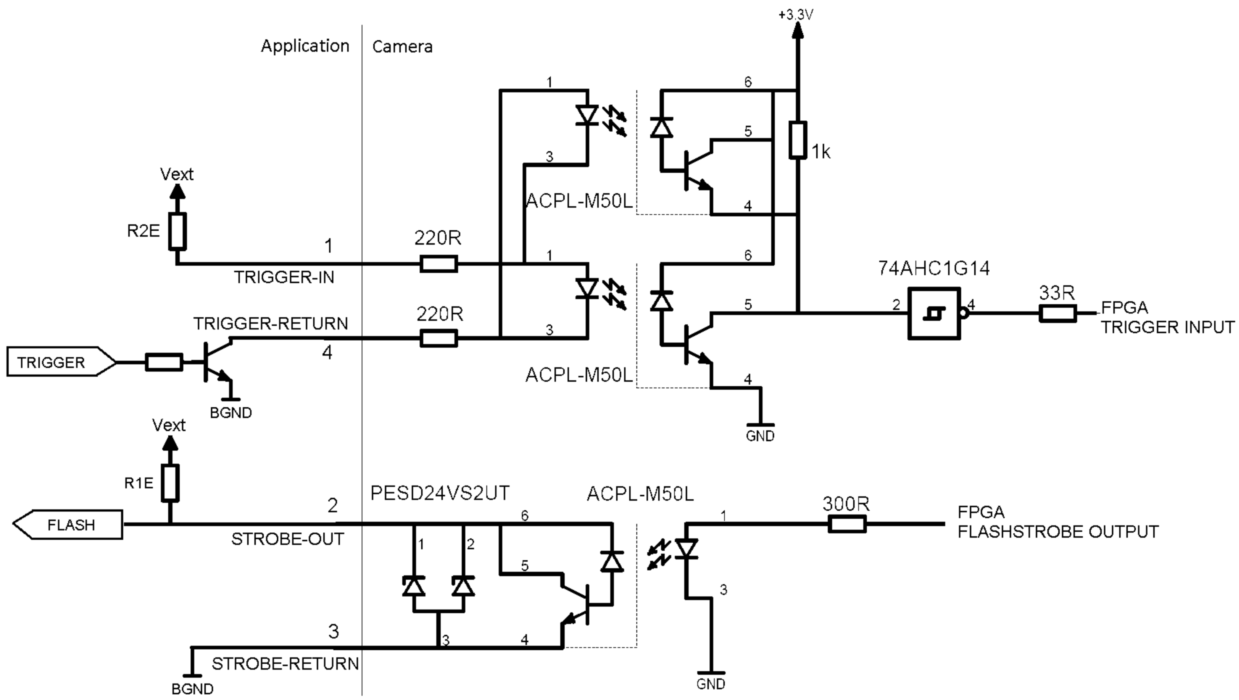


Figure 4-3: Recommended terminations strobe output and trigger input

4.2.5 Lensmount overview

The Q-12 camera series can be supplied with various optical interfaces. The available interfaces are listed in Table 4-7. Other interfaces might be possible as well by using converters. Inform at support@adimec.com for the possibilities if your interface is not listed in Table 4-7.

Table 4-7: Lensmount overview

Interface	Adjustable	Product name	Product code
M42	No	LENSMOUNT-M42-60	191540
T2	Yes	LENSMOUNT-T2-60	189080
Fn	No	LENSMOUNT-Fn-60	187490
TFLII	Yes	LENSMOUNT-TFLII-60	189590
TFLII	No	LENSMOUNT-TFLII-60	188230

5 CONTROL OF THE CAMERA

Access to camera functions and data is provided through the CoaXPress (CXP) protocol. The CoaXPress interface is GenICam compliant.

GenICam compliant means that an XML is stored in the camera that is used to translate the camera internal register addresses to the user friendly feature nomenclature as defined by the Standard Features Naming Convention, SFNC. Basically GenICam is designed to bridge the camera specific register addresses with a camera and manufacturer independent user interface. The SFNC feature names should be used to operate the camera.

How to address the SFNC feature names depends on your frame grabber. With CoaXPress frame grabbers a GenICam Application Programming Interface (GenAPI) is provided. This is a software layer that reads the XML from the camera and builds a graphical user interface (GUI) to control the camera. The GUI is often refered to as the GenICam (feature) Browser.

Next to the GUI often a scripting language will be available in which you can use the SFNC naming to program the camera and frame grabber according to your desired settings.

To illustrate the workflow of CoaXPress we will describe below what will happen if you set the pixel format to 10 bit in a monochrome camera.

When using the GUI:

1. Start the GenICam Browser, The browser will automatically load the XML from the camera and basically builds a user interface.
2. In the GenICam browser search for the feature called PixelFormat.
3. Change this feature to “Mono10”. Often this can be done by selecting “Mono10” from a drop down list.

When using a scripting language

1. Look up the syntax and language used by your frame grabber.
2. By using the frame grabber syntax and language set the feature PixelFormat to Mono10.
3. Execute the script.

In both cases, for the GUI and for the scripting language, on the background the GenAPI uses the XML to link the feature name PixelFormat to the camera register address 0x8144. Furthermore it links the feature value name “Mono10” to a value of 0x1100003. Using the CoaXPress interface, the API will then write a value of 0x1100003 to the camera register 0x8144.

Note: The above mentioned register addresses and values are only for illustrational purposes. The exact addresses in your camera might be different.

The above described communication protocol is schematically shown in Figure 5-1.

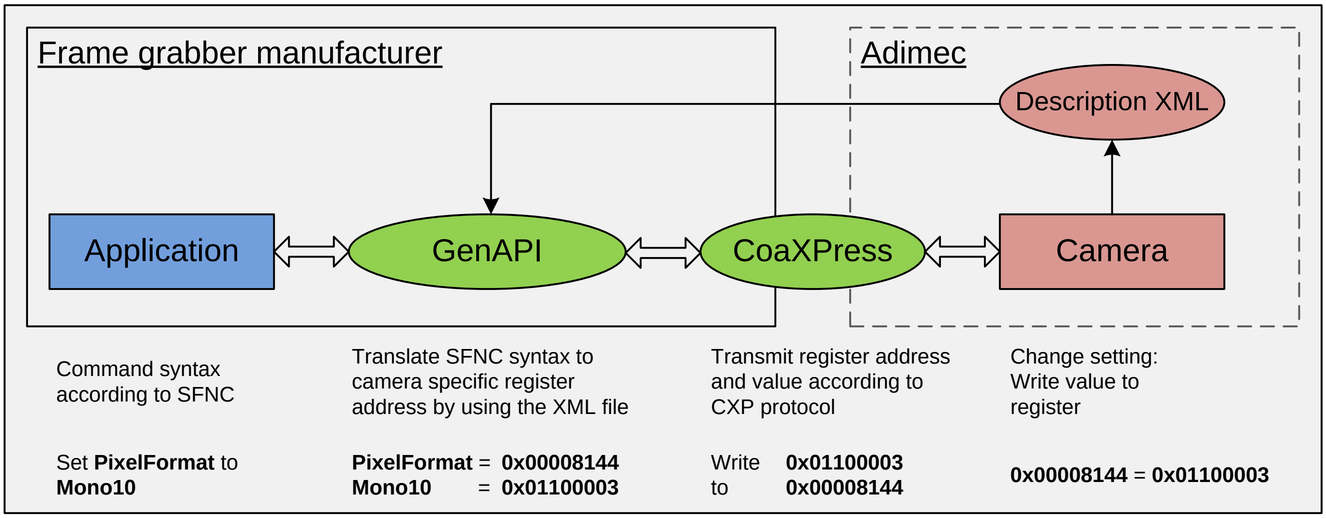


Figure 5-1: Schematic view of the CoaXPress communication protocol.

6 FUNCTIONAL DESCRIPTION

This chapter contains a functional description of the Q-12A180 camera. It briefly describes the main functions and features of the camera using a simplified block diagram. More in-depth explanations on these functions as well as descriptions on how to control them can be found in the next chapters of this manual.

6.1 Block diagram

The diagram below shows the main functional blocks of the Q-12A180 camera.

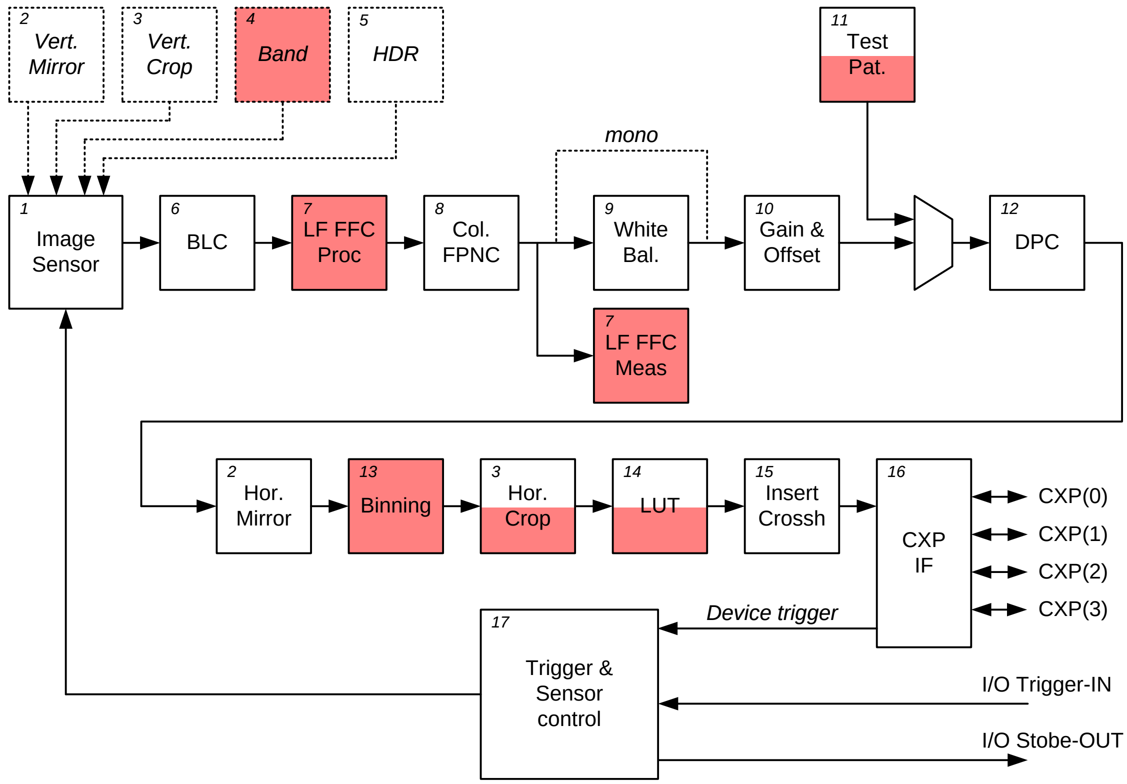


Figure 6-1: Block diagram of the camera. The dashed blocks are sensor functions. The solid blocks are camera functions. The functionality in the red blocks is only available in Issue 2 cameras.

1. An AMS CMV12000 sensor is used in this camera.

2. The output image can be mirrored horizontally and vertically.

3. A Region of Interest (ROI) can be defined by using the vertical and horizontal crop functionality. The camera will only output video information from a programmable rectangular sub frame. This reduces the amount of data and can increase the frame rate.

   Issue 2 only – The increment in width has been reduced from 32 to 16. The increment in OffsetX has been reduced from 16 to 8. This adjustment allows to enable the black clamp and at the same time leave them outside the output image with the largest resolution possible.

4. Issue 2 only - The Band function generates a rectangular image from a set of configurable bands.

5. HDR - High Dynamic Range - With this function an optical high dynamic range can be realized by using the multi slope feature of the sensor.

6. BLC - Black Level Clamp - The Black Level Clamp function clamps the black level per line using the dark reference columns if this function is enabled.

7. Issue 2 only - LF FFC – Low Frequency Flat Field Correction – This functionality can correct for a global (low frequency) non-uniformity. Such a non-uniformity can be present due to a lighting/lens profile or due to the micro lenses on the sensor.

8. FPNC - Fixed Pattern Noise Correction - Dark field and bright field column based corrections allows for correction of column based (high frequency) fixed pattern noise (PRNU and DSNU). Calibration of these functions can be performed in the field.

9. Color only - White balance: a one push and manual white balance is available to set an individual gain for the RGB channels.

10. This processing block contains the basic processing steps like digital fine gain and black level offset.

11. For functional testing of the camera and frame grabber chain, a test pattern generator is available. The test pattern generator can be enabled and disabled on demand.

    Issue 2 only – A running diagonal test pattern has been added.

12. DPC – Defect Pixel Correction - The defect pixel correction can be enabled and disabled on demand. From factory a list is available with the major defect pixels. Defect pixels can be added to or removed from the list by the user.

13. Issue 2 only - Binning – The camera supports up to 2x2 binning. Horizontal and vertical binning can be enabled independent from each other.

14. An output look-up table (LUT) is available; this table allows real-time conversion of the video levels from the processing chain according to a user programmable curve

    Issue 2 only - a Gamma curve is available in the camera. Instead of the LUT this Gamma curve can be used.

15. A crosshair overlay can be enabled. This function can be used for the optical alignment of the optics.

16. CXP IF – CoaXPress Interface - The video data is packed into the CXP format for sending data according to the CXP protocol. The output resolution can be set to 8 bit or 10 bit by user command. The output format can be set by user command. Both CXP 1.0 and 1.1.1 formats are supported and can be configured. Standard the camera is configured for CXP 1.1.1. Contact Adimec if the camera needs to be configured for CXP 1.0.

17. The acquisition rate and exposure time can be controlled externally via a selectable trigger input or the camera can operate in Timed mode where the camera generates the timing.

    The camera is equipped with a flash strobe output signal on the I/O connector. The active state of the flash strobe output can be inverted to adapt to the application requirements. The flash strobe output can be operated in two different modes, which are set through a user command.

    - The automatic mode: The flash strobe will become active after the sensor is reset and a configurable delay time is expired. The strobe will deactivate when the acquisition is completed.
    - The programmed mode: Both delay time after a sensor reset as well as the duration of the active state can be programmed.

    The delay time between the sensor reset operation in the active state of the flash strobe, as well as the duration of the flash strobe if the camera is in programmed strobe timing mode, can be user programmed.

7 FEATURE REFERENCE

7.1 Feature description structure

To clearly explain the camera features the structure as described below is used throughout this chapter.

Section headings indicate the group to which the features belong.
Subsection headings indicate the **feature name** in bold and the **accessibility** and **visibility** in normal font.

For example: **7.5.2 Width** | RW | B |

The possible accessibility and visibility values are given in respectively Table 7-1 and Table 7-2.

Table 7-1: Possible values for the accessibility level of a feature.

Accessibility level	Abbreviation	Description
Read Only	RO	Features that only present values to the user
Write Only	WO	Features that can only be written and do not give any feedback to the user
Read and Write	RW	Features that both, provide information as well as that they can be used to control the camera.

Table 7-2: Possible values for the visibility level of a feature.

Visibility level	Abbreviation	Description
Beginner	B	Features that should be visible for all users via the GUI and API. The number of features with “beginner” visibility is limited to all basic features of the devices so the GUI display is well-arranged and is easy to use.
Expert	E	Features that require a more in-depth knowledge of the camera functionality. This is the visibility level for all advanced features in the cameras.
Guru	G	Advanced features that might bring the cameras into a state where it will not work properly anymore if it is set incorrectly for the cameras current mode of operation. The guru parameters mainly have use in debugging.
Invisible	I	This is a special visibility level. It is applied to features that are required for proper operation of the camera or for specific factory settings. Not all frame grabbers make these features really invisible to the user. However **Adimec advises to not use these features**. For this reason they are not explained in this manual. The features set to invisible are listed at the end of this Chapter.

The features will mostly be described in a two column table in which the left column gives the possible input/output values and the right column a short description of the feature or specific feature value. This general presentation structure is visualized in below table.

Feature input/output value	Feature or feature value description

In some exceptional cases, to improve the readability of the manual a deviation from this lay-out might be used. For example, multiple read only features are sometimes listed in a single table.

7.2 Requirements for changing settings

Most features can be changed when the camera is acquiring images, however some features require acquisition to be stopped before changing them. To easily control also these features while acquisition is running the camera automatically performs a sequence of acquisition stop, setting change and acquisition start, but it could occur that during this sequence frames being dropped. It is thus advised to consciously stop acquisition for these features. The relevant features are listed in Table 7-3.

Table 7-3: Features which require acquisition stop after setting change.

Section	Feature name	Description
Bootstrap CoaXPress	ConnectionConfig	Holds a valid combination of Device connection speed and number of active connections.
	StreamPacketSizeMax	Provide the maximum stream packet data size the Host can accept.
Image Format Control	Width	Width of the Image provided by the device (in pixels).
	Height	Height of the image provided by the device (in pixels).
	OffsetX	Horizontal offset from the origin to the region of interest (in pixels).
	OffsetY	Vertical offset from the origin to the region of interest (in pixels).
	BinningHorizontal	Number of horizontal pixels to combine together. (Issue 2 only)
	BinningVertical	Number of vertical pixels to combine together. (Issue 2 only)
	ReverseX	Flip horizontally the image sent by the device. The Region of interest is applied after the flipping.
	ReverseY	Flip vertically the image sent by the device. The Region of interest is applied after the flipping.
	PixelFormat	Format of the pixel to use for acquisition.
Acquisition Control	TriggerSource	Specifies the internal signal or physical input Line to use as the trigger source.
	ExposureMode	Sets the operation mode of the Exposure (or shutter).
Band (Issue 2 only)	BandEnable	Enable the band functionality.
	BandLoad	(only when BandEnable = true) Restore the band user list to the device and make it active.
	BandAdd	(only when BandEnable = true) Add the band determined by BandWriteOffsetY and BandWriteHeight to non volatile memory.
	BandRemove	(only when BandEnable = true) Remove the band determined by BandWriteOffsetY from non-volatile memory.

Section	Feature name	Description
	BandClearAll	(only when BandEnable = true) Remove all bands from non volatile memory.
UserSet Control	UserSetLoad	Loads the User Set specified by *UserSetSelector* to the device and makes it active.

7.3 Bootstrap CoaXPress

The Bootstrap CoaXPress group contains features that are required for device discovery and basic configuration.

7.3.1 User Read Only Bootstrap features

Name	Visibility	Description
Standard	B	A Unique Identification of the CoaXPress Standard.
Revision	B	Revision of the CoaXPress specification implemented.
XmlManifestSize	G	The number of XML manifests available.
XmlManifestSelector	G	Selects the XML manifest entry.
XmlVersion	G	Indicates the version of the XML file referenced by the XmlManifestSelector.
XmlSchemeVersion	G	Indicates the scheme version of the XML file referenced by the XmlManifestSelector.
XmlUrlAddress	G	Indicates the start of the URL string referenced by the XmlManifestSelector.
Iidc2Address	G	If the Device supports the IIDC2 protocol, then this feature shall provide the address of the start of the IIDC2 register space.
DeviceConnectionID	G	Provides the ID of the Device connection via which this register is read.
ControlPacketSizeMax	G	Provides the maximum control packet data size. The size is defined in bytes, and shall be a multiple of 4 bytes.
ConnectionConfigDefault	B	Holds a valid default mode combination of Device connection speed and number of active connections.
TestErrorCount	G	Current connection error count selected by TestErrorCountSelector.
TestPacketCountTx	G	Current connection test transmit packet count selected by TestErrorCountSelector.
TestPacketCountRx	G	Current connection test receive packet count selected by TestErrorCountSelector.

Name	Visibility	Description
WidthAddress	G	This feature provides the address in the manufacturer-specific register space of the feature with the corresponding name.
HeightAddress	G	
AcquisitionModeAddress	G	
AcquisitionStartAddress	G	
AcquisitionStopAddress	G	
PixelFormatAddress	G	
DeviceTapGeometryAddress	G	
Image1StreamIDAddress	G	
HsUpconnection	G	Indicates the Device support of the optional high speed upconnection.

7.3.2 Beginner writable Bootstrap features

7.3.2.1 ConnectionConfig | RW | B |

With ConnectionConfig the connection speed and number of active connections is configured.

NOTE: Acquisition must be stopped before changing the ConnectionConfig feature.

CXP3_X1	Image acquisition with 1 connection and data transfer at 3.125 Gb/s
CXP6_X1	Image acquisition with 1 connection and data transfer at 6.250 Gb/s
CXP3_X2	Image acquisition with 2 connections and data transfer at 3.125 Gb/s
CXP6_X2	Image acquisition with 2 connections and data transfer at 6.250 Gb/s
CXP3_X4	Image acquisition with 4 connections and data transfer at 3.125 Gb/s
CXP6_X4	Image acquisition with 4 connections and data transfer at 6.250 Gb/s

The CXP link configuration is programmable. The number of links can be programmed to be 1, 2 or 4. Independently the bit rate that applies to all active links can be selected to be 3.125 Gbps or 6.25 Gbps.

Depending on the CXP link configuration the camera is optimized for minimum power dissipation.

7.3.3 Guru writable Bootstrap features

Features with a “Guru” visibility level control advanced camera settings. If these features are used incorrectly the camera might not work properly anymore. Most of the time guru parameters mainly have use in debugging.

Name	Access	Description
ConnectionReset	RW	Write “1” to reset all connections of the Device.
MasterHostConnectionID	RW	Holds the Host Connection ID of the Host connection connected to the Device Master connection.
StreamPacketSizeMax	RW	Provide the maximum stream packet data size the Host can accept. The size is defined in bytes, and shall be a multiple of 4 bytes. The default value is always “0”. This value is set by the Host and not the Device.
TestMode	RW	Enables test packet transmission from Device to Host.
TestErrorCountSelector	RW	Selects the TestErrorCount register. Selection shall be a valid Device Connection ID.
ElectricalCompliancyTest	RW	Supports the formal electrical compliancy testing of the Device.

7.4 Device control

The device control features give basic information about the device. It contains features related with the identification and status of the device.

7.4.1 User Read Only Device Control features

Name	Visibility	Description
DeviceVendorName	B	Name of the manufacturer of the device.
DeviceModelName	B	Model of the device.
DeviceManufacturerInfo	B	Manufacturer information about the device.
DeviceVersion	B	Version of the device.
DeviceFirmwareVersion	B	Version of the firmware in the device.
DeviceSerialNumber	E	Device serial number.
SensorTemperature	E	Returns the temperature of the sensor in °C.

7.4.2 BuiltInTest | RO | E |

BuiltInTest can give multiple error messages at the same time. The error values are then added together. To decode which error has occurred start with the largest value that fits into the returned BuiltInTest value and then subtract it. After subtracting again search for the largest number that fits into the remaining value. Continue until after subtracting the value equals zero.

Example:

BuiltInTest 2084
2048 – Camera configuration corrupt

Remainder 0036
0032 – User defect pixel data corrupt

Remainder 0004
0004 – Factory settings corrupt

Value	Issue	Failure condition	When tested
1	1 & 2	FPGA not booted	At start-up
2	1 & 2	Flash not recognized	At start-up
4	1 & 2	Factory settings corrupt	When data is read from flash
8	1 & 2	User settings corrupt	When data is read from flash
16	1 & 2	Factory defect pixel data corrupt	When data is read from flash
32	1 & 2	User defect pixel data corrupt	When data is read from flash
64	1 & 2	Calibration corrupt	When data is read from flash
128	1 & 2	5V power supply error	At start-up
256	1 & 2	1.8V power supply error	At start-up
512	1 & 2	1.2V power supply error	At start-up
1024	1 & 2	1V Power supply error	At start-up
2048	1 & 2	Camera configuration corrupt (m/c, Bayer phase)	At start-up
4096	1 & 2	Look-Up Table corrupt	When data is read from flash
8192	1 & 2	Device Names corrupt	At start-up
16384	1 & 2	Sensor data alignment failed	Continuous; actual status is updated on a 1 sec. interval
32768	2	Band table corrupt	When data is read from flash
65536	2	Sequence table corrupt	When data is read from flash
131072	2	LF FF Set table corrupt	When data is read from flash

7.4.3 **DeviceUserID** | RW | B |

String	User-programmable device identifier. The string is directly written to non-volatile memory. Up to 16 Characters can be used.

7.4.4 **DeviceIndicatorMode** | RW | E |

Inactive	Turn off the status indicator LEDs.
Active	Turn on the status indicator LEDs.
ErrorStatus	Let the LEDs show the error status of the camera.

7.5 Image Format Control

The features in the image format control group influence the image format. You can set here a region of interest (ROI) and the pixel format for example.

7.5.1 User Read Only Image Format Control features

Name	Visibility	Description
SensorWidth	E	Effective width of the sensor in pixels.
SensorHeight	E	Effective height of the sensor in pixels.
WidthMax	E	Maximum width (in pixels) of the image.
HeightMax	E	Maximum height (in pixels) of the image.
DeviceTapGeometry	E	Tap geometry to be used by streams of the Device.
Image1StreamID	G	Identification of stream 1.

7.5.2 **Width** | RW | B |

ISSUE 1	32 to 4096, Increment: 32	Set the width of the image in pixels.
ISSUE 2	32 to 4096, Increment: 16	Set the width of the image in pixels.

ISSUE 2 NOTE: When binning, the maximum range has to be divided by two.

ISSUE 2 NOTE: The width has to be a multiple of 16 in both, binned and non binned mode. If a width is selected that is dividable by 16 in non-binned mode but only by 8 in binned mode then when binning is enabled the width will be clipped to the largest width that is dividable by 16.

Example:
When not binning, the width is set to 4080. This is dividable by 16. When 2x horizontal binning is enabled, this width should be divided by two, resulting in 2040. However 2040 is not dividable by 16. Therefore the width in binned mode is round down to 2032 which is dividable by 16. When returning to no binning, the rounded width will be multiplied by two and become 4064.

ISSUE 2 NOTE: This feature is automatically updated if any of the following features is changed: BinningHorizontal

7.5.3 **Height** | RW | B |

32 to 3072 Increment: 4	Set the height of the image in pixels.

ISSUE 2 NOTE: When binning is enabled, the maximum range and increment have to be divided by two.

ISSUE 2 NOTE: This feature is automatically updated if any of the following features is changed: BandAdd, BandClearAll, BandEnable, BandRemove, BandLoad, BandSelector, BinningVertical

7.5.4 **OffsetX** | RW | B |

ISSUE 1	0 to max 4064, Increment: 16	Set the horizontal offset from the origin to the region of interest in pixels. OffsetX is limited by the output width programmed.
ISSUE 2	0 to max 4064, Increment: 8	Set the horizontal offset from the origin to the region of interest in pixels. OffsetX is limited by the output width programmed.

ISSUE 2 NOTE: When binning is enabled, the maximum range and increment have to be divided by two.

ISSUE 2 NOTE: This feature is automatically updated if any of the following features is changed: BinningHorizontal

7.5.5 **OffsetY** | RW | B |

0 to max 3068 Increment: 4	Set the horizontal offset from the origin to the region of interest in pixels. OffsetY is limited by the output height programmed.

ISSUE 2 NOTE: When binning is enabled, the maximum range and increment have to be divided by two.

ISSUE 2 NOTE: This feature is automatically updated if any of the following features is changed: BinningVertical

7.5.6 Binning

ISSUE 2 ONLY

In section 7.5.7 to 7.5.9 the camera features related to binning are described. This section gives some general remarks about the use of the binning function.

The binning functionality cannot be used in combination with the low frequency flat field correction.

The binning functionality is not available in color cameras.

There are three binning situations possible. Two are performed by the digital processing in the FPGA while one is done on sensor level. The situations are summarized in the below table.

BinningHorizontal	BinningVertical	Type
2	1	FPGA
1	2	FPGA
2	2	Sensor

For 2x2 binning, the ProgrammableGainAmplifier and SensorBitDepth features are not available. In binned mode the PGA gain is set to 1 and the sensor bit depth to 10 bit.

Separate DF_ and BF_ column calibrations are available for sensor binning for both average and summed binning modes.

Binning and frame rate
For 2x2 sensor binning the maximum frame rate in full ROI will increase to 267.4 fps. This maximum frame rate does not depend on the sensor bit depth or the interface bit depth. (StreamPacketSizeMax=4096, ConnectionConfig=CXP6_X4, InterfaceUtilization=100)

Binning and defect pixel behavior
From factory the defect pixel list is created in non-binned mode. It might occur that in 2x2 sensor binning mode defect binned pixels show up while none of the physical pixels inside the bin is defect in non-binned mode. These binned defect pixels need to be added manually to the defect pixel list.

A binned pixel can be marked defect by marking one of the pixels inside the bin as defect. So for 2x2 binning the following coordinate transformation can be used to mark a binned pixel defect:

$Y_{non\text{-}binned} = 2Y_{binned}$, $X_{non\text{-}binned} = 2X_{binned}$, (top left pixel has coordinates 0,0)

7.5.7 BinningHorizontal | RW | E |
ISSUE 2 ONLY

1 or 2	Set the number of horizontal pixels to combine together. This reduces the horizontal width of the image. A value of 1 indicates that no horizontal binning is performed by the camera.

7.5.8 BinningVertical | RW | E |
ISSUE 2 ONLY

1 or 2	Set the number of vertical pixels to combine together. This reduces the vertical height of the image. A value of 1 indicates that no vertical binning is performed by the camera.

7.5.9 BinningMode | RW | B |
ISSUE 2 ONLY

Sum	Set the binned pixel signal level to the sum of the signal levels of the individual pixels of which it is composed.
Average	Set the binned pixel signal level to the average of the signal levels of the individual pixels of which it is composed.

7.5.10 ReverseX | RW | E |

True	The output image is flipped horizontally. For a color camera, the pixel format is updated automatically to reflect the correct Bayer phase. The Region of interest is applied after the flipping.
False	The output image is not flipped.

7.5.11 ReverseY | RW | E |

True	The output image is flipped horizontally. For a color camera, the pixel format is updated automatically to reflect the correct Bayer phase. The Region of interest is applied after the flipping.
False	The output image is not flipped.

7.5.12 PixelFormat | RW | B |

Mono8 (Mono Only)	Set the pixel format for acquisition to 8 bit mono.
Mono10 (Mono Only)	Set the pixel format for acquisition to 10 bit mono.
BayerBG8 (Color Only)	Set the pixel format for acquisition to 8 bit BayerBG.
BayerBG10 (Color Only)	Set the pixel format for acquisition to 10 bit BayerBG.
BayerGB8 (Color Only)	Set the pixel format for acquisition to 8 bit BayerGB.
BayerGB10 (Color Only)	Set the pixel format for acquisition to 10 bit BayerGB.
BayerGR8 (Color Only)	Set the pixel format for acquisition to 8 bit BayerGR.
BayerGR10 (Color Only)	Set the pixel format for acquisition to 10 bit BayerGR.
BayerRG8 (Color Only)	Set the pixel format for acquisition to 8 bit BayerRG.
BayerRG10 (Color Only)	Set the pixel format for acquisition to 10 bit BayerRG.

NOTE: PixelFormat can only be changed if there is no acquisition active.

This feature is automatically updated if any of the following features is changed:
ReverseX, ReverseY

NOTE: Selection of one of the 4 Bayer phases is arbitrary. The PixelFormat feature in the CXP image header stream however will report the code for correct phase to be used by the host for color decoding. For example, the format changes when *ReverseX* and/or *ReverseY are enabled*.

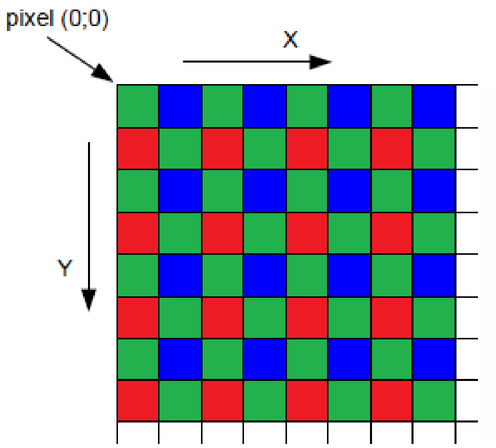


Figure 7-1: (Color Only) Bayer pattern for a color camera with ReverseX and ReverseY set to false.

The Bayer phase of the color camera with deactivated functions is defined in Figure 7-1. Pixel(0;0) is the first pixel from the camera (left-top pixel in image), the corresponding Bayer phase is GBRG. If the *ReverseX* and *ReverseY* functions are activated, the Bayer phase changes (*PixelFormat*) according to the table below.

ReverseX	*ReverseY*	*Bayer phase*	*PixelFormat*
Off	Off	GBRG	BayerGB8 / BayerGB10
Off	On	RGGB	BayerRG8 / BayerRG10
On	Off	BGGR	BayerBG8 / BayerBG10
On	On	GRBG	BayerGR8 / BayerGR10

Table 7-4: The influence of the ReverseX and ReverseY feature on PixelFormat.

7.5.13 **TestImageSelector** | RW | B |

Off	No test pattern is shown.
AdimecTestPattern	Specific Adimec test pattern with grey bars, color bars (only active in a color camera) and contour lines, see Figure 7-2.
UniformVideoLevel	Uniform test pattern to verify corrections. The video level can be set with the TestImageVideoLevel feature.
DiagonalPattern	The diagonal test pattern is defined by **$V_{x,y}$ = (x + y) modulo 2^N** where N represents the bit depth and $V_{x,y}$ the digital pixel value at pixel (x,y). See Figure 7-3 for the resulting pattern.
DiagonalPatternRunning (ISSUE 2 ONLY)	This running test pattern is different from the diagonal test pattern by the addition of the frame counter in the defining formula: **$V_{x,y}$ = (x + y + frame counter) modulo 2^N**. See Figure 7-3 for the resulting pattern.

The camera can generate test patterns in the mode the camera is currently working. The camera will continue to work in the selected mode, but instead of the usual image an artificial image is displayed.

NOTE: The test patterns are amplified with the set gain.

NOTE: The test image is resized when a different width and/or height is set.

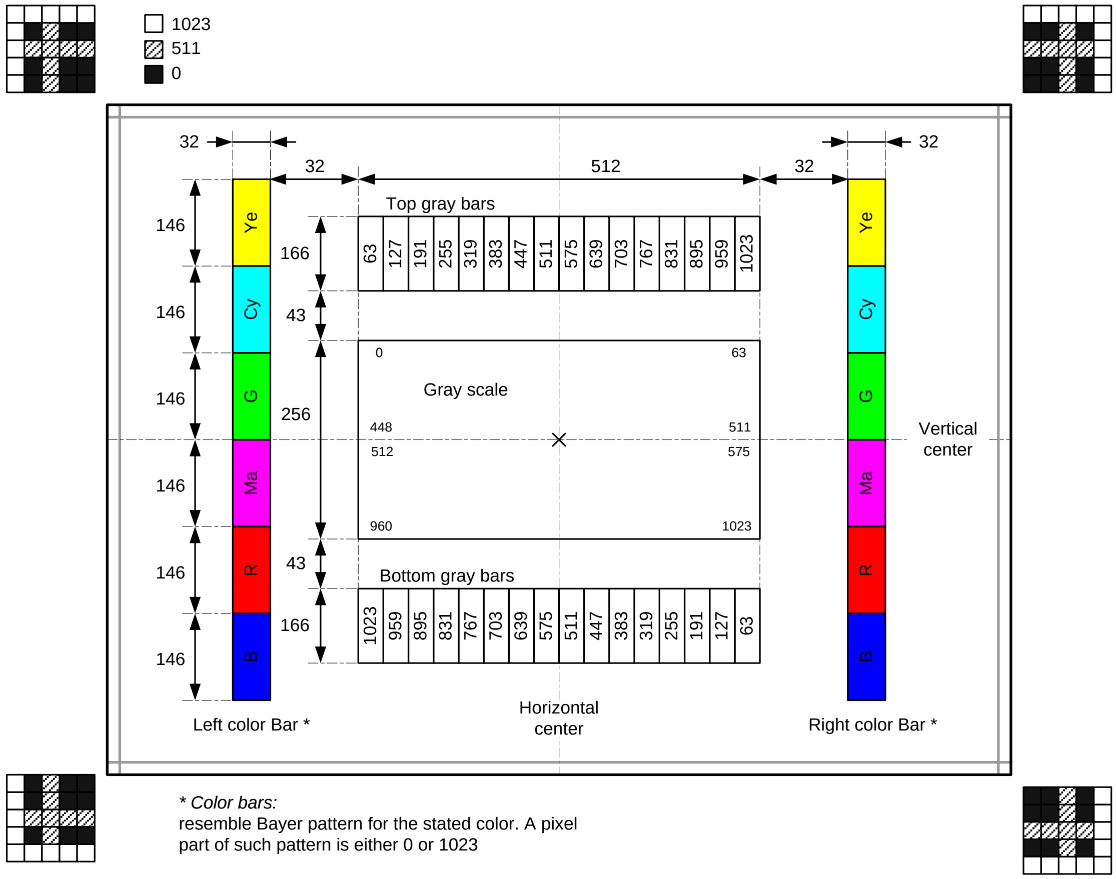


Figure 7-2: The Adimec Test Pattern in a 10 bit pixel resolution, black is 0 and white is 1023. The test image is defined in 10 bit. When an 8 bit pixel format is selected the lowest 2 bits are discarded. When the region of interest gets too small parts of the test pattern will disappear. The border of the test pattern will always be visible. Monochrome camera models do not have the side color bars.

(a) (b)

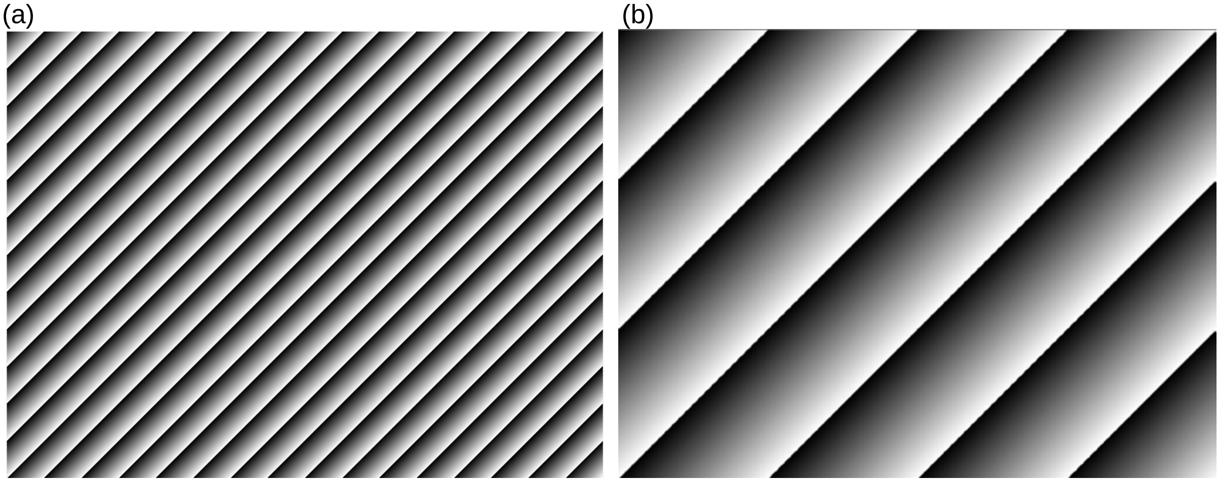


Figure 7-3: Visualization of the diagonal test pattern in 8 bit (a) and 10 bit (b) pixel resolution.

7.5.14 **TestImageVideoLevel** | RW | B |

0 to 1023, Increment: 1	Set the video level for the UniformVideoLevel test image. The video level is always in 10 bit.

7.5.15 CrosshairOverlay | RW | E |

True	Add a crosshair overlay to the image sent by the device. The crosshair is applied to the center of the image.
False	No crosshair will be added to the camera image.

7.6 Acquisition control

7.6.1 AcquisitionMode | RW | B |

Continuous	Set the acquisition mode of the device.

NOTE: AcquisitionMode is related to how data is transferred over the interface. ExposureMode is related to the sensor operation.

7.6.2 AcquisitionStart | RW | B |

Command	Start the Acquisition of the device.

7.6.3 AcquisitionStop | RW | B |

Command	Stop the Acquisition of the device at the end of the current frame.

7.6.4 AcquisitionFrameRate | RW | B |

Min 10 Hz	Control the acquisition rate (in Hertz) at which the frames are captured. The maximum depends on the camera configuration.

This feature is automatically updated if any of the following features is changed: AcquisitionFrameRate, AcquisitionFramePeriod, AcquisitionMaxFrameRate, ConnectionConfig, InterfaceUtilization, PixelFormat, Width, Height, StreamPacketSize

7.6.5 AcquisitionFramePeriod | RW | B |

Max 100000 µs	Control the acquisition rate (in 1 µs steps) at which the frames are captured. The minimum depends on the camera configuration.

This feature is automatically updated if any of the following features is changed: AcquisitionFrameRate, AcquisitionFramePeriod, AcquisitionMaxFrameRate, ConnectionConfig, InterfaceUtilization, PixelFormat, Width, Height

7.6.6 AcquisitionFramePeriodRaw | RW | B |

Max 100000 µs	Control the acquisition rate (in 1 µs steps) at which the frames are captured. The minimum depends on the camera configuration.

This feature is automatically updated if any of the following features is changed: AcquisitionFrameRate, AcquisitionFramePeriod, AcquisitionMaxFrameRate, ConnectionConfig, InterfaceUtilization, PixelFormat, Width, Height

7.6.7 **AcquisitionMaxFrameRate** | WO | B |

Command	Set the camera to the maximum frame rate as is possible with the current settings. This function could be also used to calculate the maximum achievable frame rate in case the camera is set in a slave exposure mode.

NOTE: To achieve the maximum frame rate of 187 fps in non binned and full sensor ROI, SensorBitDepth and PixelFormat have to be set to 8 bit.

7.6.8 **TriggerSource** | RW | B |

Trigger	Use an internal signal for triggering.
IO_Connector	Use the physical input line to provide the trigger signal.

Not applicable in ExposureMode Timed.

7.6.9 **TriggerActivation** | RW | B |

FallingEdge	Use the falling edge as the trigger activation event.
RisingEdge	Use the rising edge as the trigger activation event.

Not applicable in ExposureMode Timed.

NOTE: External control signals are conditioned by the camera: pulses shorter than 133.33 ns are rejected (glitch rejection).

7.6.10 ExposureMode | RW | B |

The camera can be operated in 5 exposure modes. Refer to Figure 7-4 for a description of these modes.

Timed	Free run mode. The camera is master: frame period and integration time are both fixed and controllable via the AcquisitionFramePeriod feature and ExposureTime feature respectively.
TriggerWidth	Camera is slave; In this mode an external trigger starts integration. The integration time is determined by the duration of the trigger pulse.
SyncControlMode	Camera is slave: In sync control mode the acquisition is controlled by the active edge of the input signal. The inactive edge does not trigger any event. The active edge is programmable; rising or falling. The integration time is targeted to be equal to the frame period, however due to the time required for initiating frame output (Frame Overhead Time/FOT) the integration time will always be slightly shorter than the frame period. **ISSUE 2 NOTE:** If acquisition is started, the first trigger pulse will output an image (the first integration time is directly initiated). Note that in general this results in an image from which the integration time is unknown.
TimedTriggerControl	Camera is slave: Start of integration time is determined by the start of the trigger, the integration time is fixed and can be controlled via the ExposureTime feature.
TimedSync	Camera is slave but timing should be synchronized with internal camera timing. In this exposure mode the frame sequence is synchronized to the edge of the external trigger signal. Characteristics: - The *AcquisitionFramePeriod* must be set slightly smaller than the external sync period time. (Take into account that the absolute camera timing has a deviation of ±120 ppm). - When locked the camera frame rate equals the applied external sync signal frequency.

Acquisition must be stopped when changing the ExposureMode.

7.6.10.1 Exposure time and Frame Overhead Time

CMOS sensors require a certain time after the requested integration time is finished before they can start with the readout process. This additional time is called the Frame Overhead Time (FOT). The FOT consists of a period during which the sensor remains light sensitive and a period in which the sensor is not light sensitive anymore. Especially the light sensitive FOT time is often important to take into account in applications.

The light sensitive FOT time is a fixed value. It only changes with the SensorBitDepth setting. The duration of this light sensitive FOT time for the available SensorBitDepth settings is listed in Table 7-5. The non light sensitive FOT time depends on the camera configuration in a more complex way. Therefore in issue 2 cameras it is possible to read-out the actual FOT time with the FrameOverheadTime feature (section 7.10.3). This feature is especially useful for the SyncControlMode as it influences the maximum integration time as indicated in Table 7-6.

During the FOT it is not allowed to sent an acquisition start trigger.

Figure 7-5 and Figure 7-6 show an example of how the FOT time influences the timing in the TriggerWidth and SyncControlMode. While for the TriggerWidth only the light sensitive FOT (t_sens) influences the integration time, for the SyncControlMode the whole FOT time influences the integration time.

Table 7-5: Minimum exposure time. Please note that the minimum value in the ExposureTime feature is 1 µs. Therefore the minimum exposure time equals the FOT + 1 µs.

SensorBitDepth	t_sens (µs)	Min. exposure time (µs)
8	15.41	15.41 + 1 = 16.41
10	15.3	15.3 + 1 = 16.3

Table 7-6: Integration time for the SyncControlMode.

SensorBitDepth	Actual integration time [µs]
8-bit	AcquisitionFramePeriod – FOT + 15.41
10-bit	AcquisitionFramePeriod – FOT + 15.3

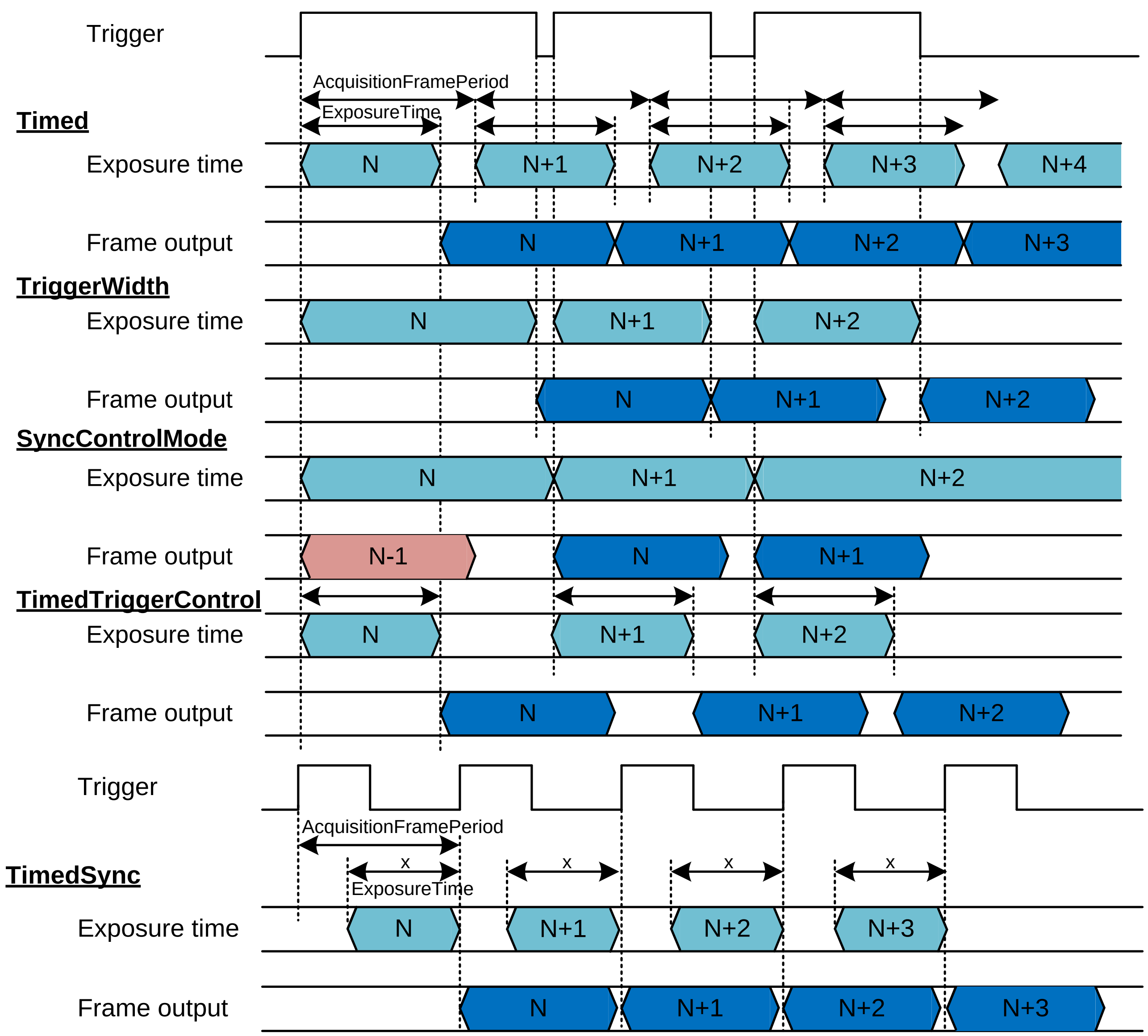


Figure 7-4: Exposure modes timing diagram. For the SyncControlMode, note that the N-1 image is only present in the issue 2 camera. The integration time for the N-1 image in general is unknown.

TriggerWidth

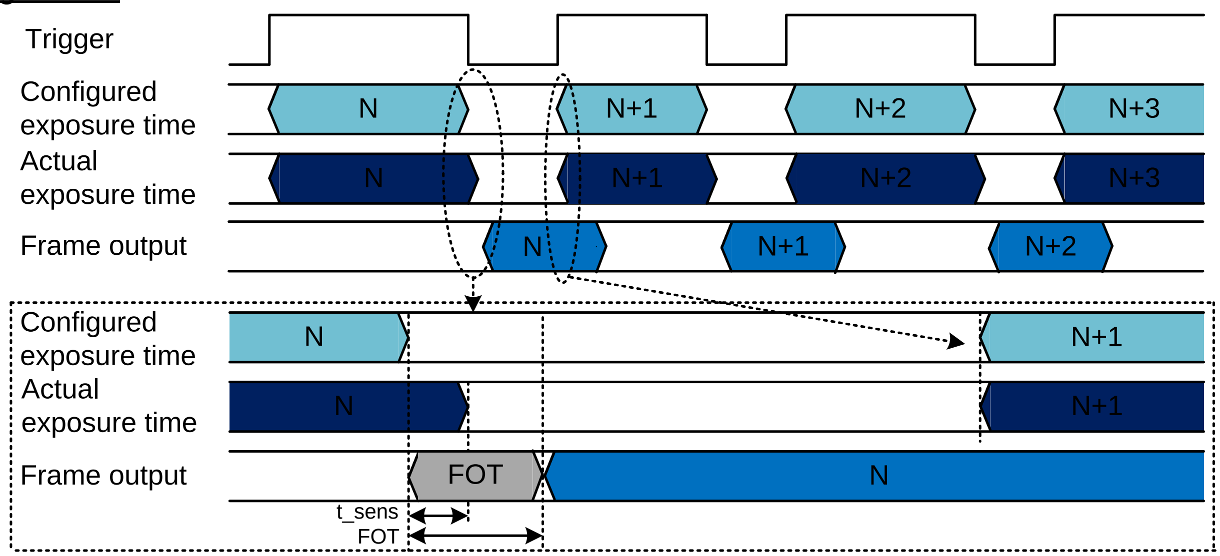


Figure 7-5: Exposure timing and latency during FOT in TriggerWidth mode.

SyncControlMode

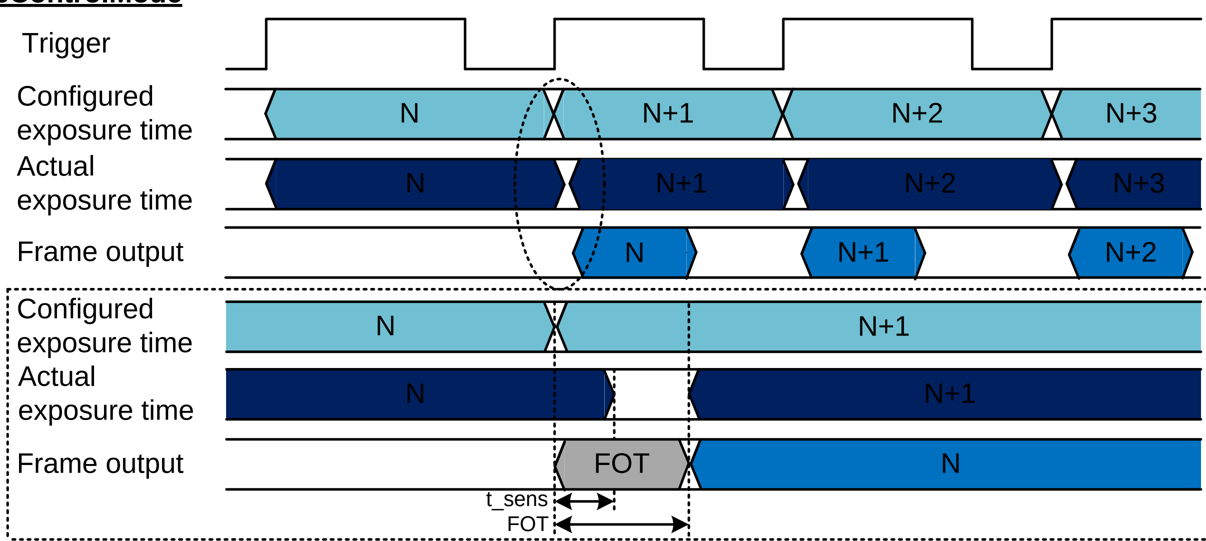


Figure 7-6: Exposure timing and latency during FOT in SyncControlMode.

7.6.10.2 Frame skipping due to over triggering

In the triggered mode, the sensor can be over triggered; in this case a new image is requested from the sensor, while it is still busy outputting an image frame. In this case the requested frame is being delayed according to Figure 7-7.

TriggerWidth

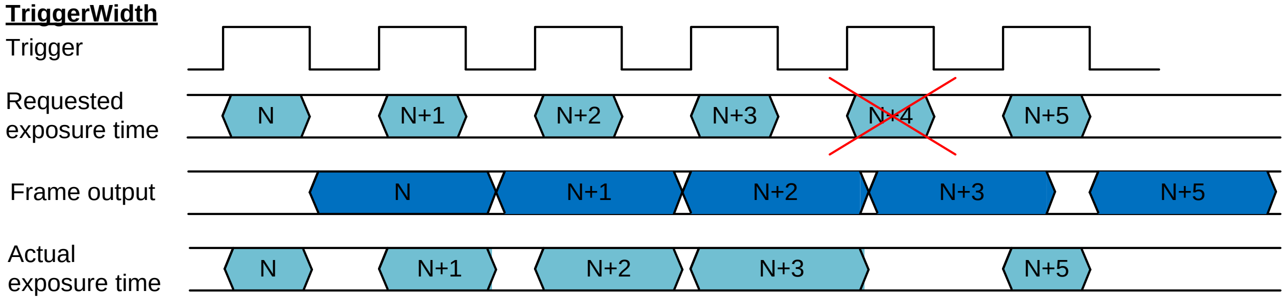


Figure 7-7: Over triggering behavior.

7.6.11 ExposureTime | RW | B |

Min: 1µs Max: AquisitionFramePeriod	Set the exposure time (in 1µs steps). The exposure time is not corrected for the light sensitive FOT of the sensor. To obtain the actual integration time the light sensitive FOT has to be added to the exposure time set with this feature. See section 7.6.10.1 for details about the FOT.

This feature is automatically updated if any of the following features is changed:
ExposureTimeRaw

7.6.12 ExposureTimeRaw | RW | B |

Min: 1µs Max: AquisitionFramePeriod	Set the exposure time (in 1µs steps). The exposure time is not corrected for the light sensitive FOT of the sensor. To obtain the actual integration time the light sensitive FOT has to be added to the exposure time set with this feature. See section 7.6.10.1 for details about the FOT.

This feature is automatically updated if any of the following features is changed:
ExposureTime, AcquisitionFramePeriodRaw

7.6.13 InterfaceUtilization | RW | B |

50% to 100%, Increment: 1%	Decrease the data rate of the interface in order to prevent the frame grabber from being overrun. Example: a utilization factor of 50 halves the available interface bandwidth.

The utilization factor does not change the interface configuration (*ConnectionConfig*) but creates an empty period in between the readout of lines. An example of the utilization factor is shown in Figure 7-8. When lowering the utilization factor the line readout period will increase which directly influences the duration of the frame output. The maximum achievable frame rate after changing the utilization factor can be requested with the *AcquisitionMaxFramerate* feature.

Example: 50% utilization factor is a doubling of the readout time of one line with respect to the fastest line readout period (100%). The maximum frame rate will thus reduce by 50% if the frame grabber is the limiting data transfer factor.

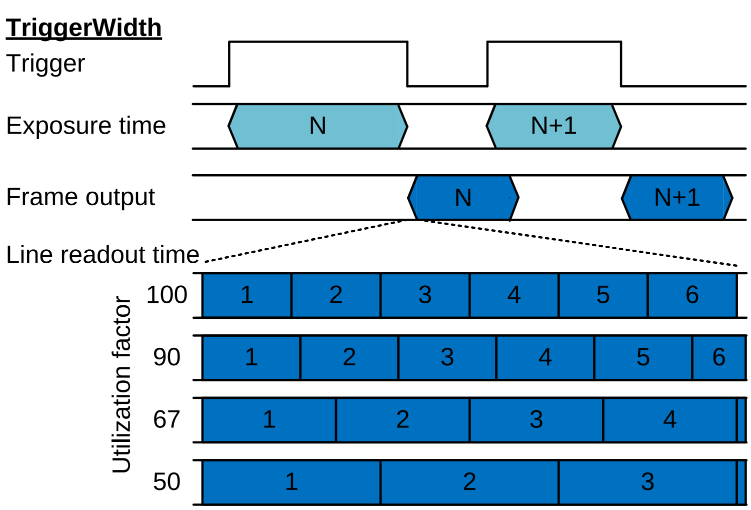


Figure 7-8: Impact of utilization factor on line readout timing.

7.7 Counter and timer control

This group contains the features related with externally providing the camera with a trigger signal.

7.7.1 **FlashStrobeMode** | RW | B |

Disabled	Flash strobe inactive.
Automatic	The flash strobe output will become active during the integration time of the sensor (regardless of the exposure mode).
Programmed	The flash strobe output will become active during a programmable time (FlashStrobeDuration) and after a programmable delay (FlashStrobeDelay) starting at the beginning of sensor integration.

7.7.2 **FlashStrobeDuration** | RW | B |

1 to 65535 µs Increment: 1 µs	Control the duration of the flash strobe in µs. The duration cannot be longer then the frame period.

This feature is automatically updated if any of the following features is changed:
FlashStrobeDurationRaw

7.7.3 **FlashStrobeDurationRaw** | RW | B |

1 to 65535 µs, Increment: 1 µs	Control the duration of the flash strobe in 1 µs steps. The duration cannot be longer then the frame period.

This feature is automatically updated if any of the following features is changed:
FlashStrobeDuration

7.7.4 **FlashStrobeDelay** | RW | B |

0 to 65535 µs Increment: 1 µs	Control the delay of the flash strobe in µs. The delay cannot be longer then the frame period.

This feature is automatically updated if any of the following features is changed:
FlashStrobeDelayRaw

7.7.5 **FlashStrobeDelayRaw** | RW | B |

0 to 65535 µs, Increment: 1 µs	Control the delay of the flash strobe in 1 µs steps. The delay cannot be longer then the frame period.

This feature is automatically updated if any of the following features is changed:
FlashStrobeDelay

7.7.6 **FlashStrobeActiveState** | RW | B |

NonConducting	The photo transistor at the camera output is non conductive during the strobe active time.
Conducting	The photo transistor at the camera output is conductive during the strobe active time.

7.8 High Dynamic Range Mode

In the camera, a high optical dynamic range can be realized by setting the camera to a piece-wise linear response with up to 3 slopes. This section describes briefly the HDR features. An application note is available for a more detailed explanation. Contact support@adimec.com to request the application note.

NOTE: The multi slope can currently only be used in *Timed* and in *TimedTriggerControl* exposure mode.

7.8.1 HDR_Mode | RW | E |

Off	Disable the HDR mode.
MultiSlope2Slopes	Activate the 2 slopes HDR mode with one node.
MultiSlope3Slopes	Activate the 3 slopes HDR mode with two nodes.

7.8.2 SecondExposureTimeRaw | R | Invisible* |

Read only	This is for a factory test only it is not accessible for users.

* Some frame grabbers might show features that are marked as invisible

7.8.3 SecondExposureTime | R | Invisible* |

Read only	This is for a factory test only it is not accessible for users.

* Some frame grabbers might show features that are marked as invisible

7.8.4 MultiSlopeNodeSelector | RW | E |

1 or 2	Select to edit the MultiSlopeLevel and/or MultiSlopeTime for node 1 or node 2. See Figure 7-10 for a graphical explanation.

7.8.5 MultiSlopeLevel | RW | E |

1 to 100% Increment: 1%	Set the level at which the selected node saturates. The level is set in percentage of the pixel video level. See Figure 7-10 for a graphical explanation. **NOTE:** the setting of the second node can't be smaller than the setting of the first node.

7.8.6 MultiSlopeTime | RW | E |

1 to 99% Increment: 1%	Set the time duration for which the selected node should hold its saturation level. The time duration is set in a percentage of the integration time. See Figure 7-10 for a graphical explanation. **NOTE:** the setting of the second node can't be smaller than the setting of the first node.

7.8.7 The advantages of using the HDR mode

Basically the multi slope response serves as a white compression, still enabling contrast in parts that else would be saturated. For high exposures, this ensures contrast in bright part of the scene that would otherwise saturate. An example of the HDR mode disabled and enabled is shown in Figure 7-11. This example shows clearly that the multi slope allows for contrast in the lamp which is with a normal linear response saturated.

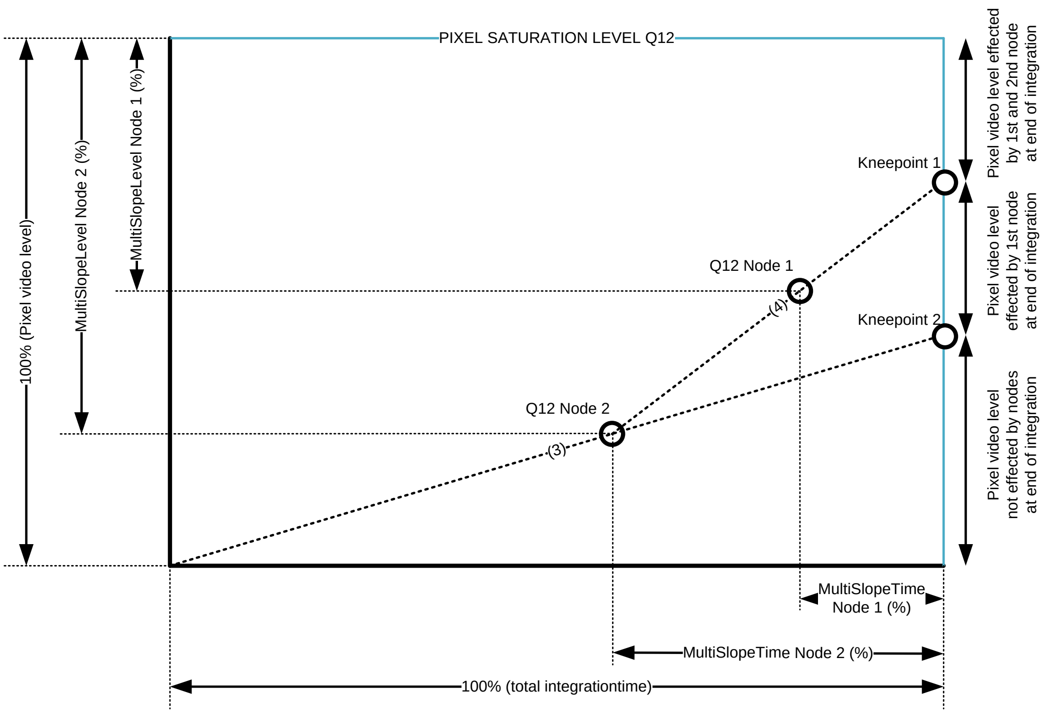


Figure 7-9: The graph above illustrates how the nodes are defined with the HDR features. It also shows how the nodes define the kneepoints as shown in Figure 7-10.

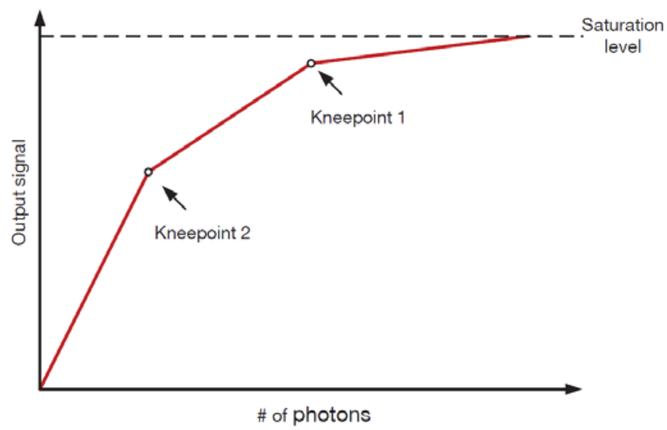


Figure 7-10: The graph above illustrates the output signal that can be obtained by defining the nodes with the HDR functionality.

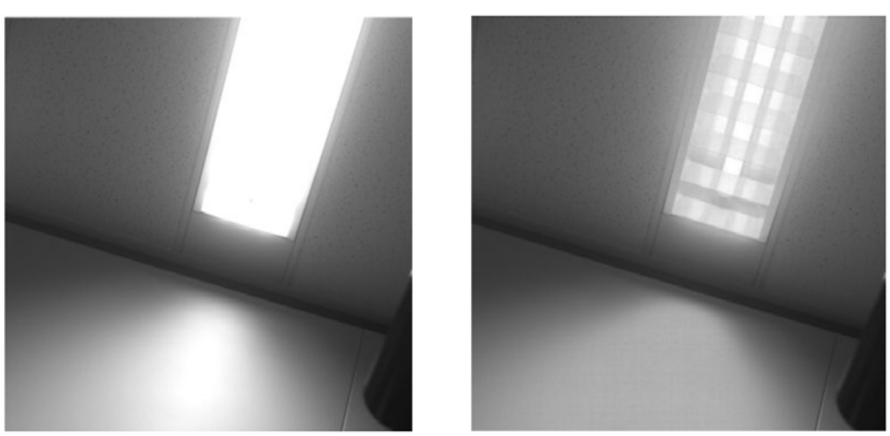


Figure 7-11: Image of fluorescent light with normal operation (left) and multislope function enabled (right).

7.9 Analog control

Analog control functions like gain and white balance can be found in this group.

7.9.1 **GainSelector** | RW | B |

All	Gain features will influence all pixels.
Red (Color Only)	Gain features will only influence the red pixels.
Green (Color Only)	Gain features will only influence the green pixels.
Blue (Color Only)	Gain features will only influence the blue pixels.

NOTE: Gain is applied before BlackLevel.

7.9.2 **Gain** | RW | B |

GainSelector value	Gain range
All	1 to 8 (Color Only) 1 to 32 (Monochrome Only) Increment: 0.001.
Red	1 to 4 (Color Only) Increment: 0.001.
Green	1 to 4 (Color Only) Increment: 0.001.
Blue	1 to 4 (Color Only) Increment: 0.001.

NOTE: This gain is a digital gain. For analog gain use the ProgrammableGainAmplifier.

This feature is automatically updated if any of the following features is changed:
GainRaw, GainSelector, WhiteBalanceCalibrate

7.9.3 **GainRaw** | RW | B |

GainSelector value	Gain range
All	1000 to 8000 (Color Only) 1000 to 32000 (Monochrome Only) Increment: 1.
Red	1000 to 4000 (Color Only) Increment: 1.
Green	1000 to 4000 (Color Only) Increment: 1.
Blue	1000 to 4000 (Color Only) Increment: 1.

NOTE: This gain is a digital gain. For analog gain use the ProgrammableGainAmplifier.

This feature is automatically updated if any of the following features is changed:
Gain, GainSelector, WhiteBalanceCalibrate

7.9.4 **BlackLevel** | RW | B |

0 to 511 Increment: 1	Control the analog black level as an absolute physical value.

This feature is automatically updated if any of the following features is changed:
BlackLevelRaw

For the 8-bit pixel formats, the configured value is presented at the video output as BlackLevel/4.

NOTE: BlackLevel is applied after gain.

7.9.5 **BlackLevelRaw** | RW | B |

0 to 511 Increment: 1	Control the analog black level as an absolute physical value.

This feature is automatically updated if any of the following features is changed:
BlackLevel

For the 8-bit pixel formats, the configured value is presented at the video output as BlackLevelRaw/4.

NOTE: BlackLevelRaw is applied after gain.

7.9.6 White balance window

The calculation of the white balance is performed on the defined white balance window. This white balance window is clipped to fall within the set video region of interest. The window is defined with the features as described in 7.9.7 to 7.9.10.

7.9.7 **WhiteBalanceWidth** | RW | E |

Color Only

32 to SensorWidth Increment: 1	Width of the “white balance region of interest” that is used for white balance calibration.

This feature is automatically updated if any of the following features is changed:
Width

7.9.8 **WhiteBalanceHeight** | RW | E |

Color Only

2 to SensorHeight Increment: 1	Height of the “white balance region of interest” that is used for white balance calibration.

This feature is automatically updated if any of the following features is changed:
Height

7.9.9 **WhiteBalanceOffsetX** | RW | E |

Color Only

0 to SensorWidth-16 Increment: 1	Horizontal offset, relative to OffsetX, for the “white balance region of interest”, the region that is used for white balance calibration. (in pixels).

This feature is automatically updated if any of the following features is changed:
OffsetX

7.9.10 WhiteBalanceOffsetY | RW | E |

Color Only

0 to SensorHeight-2 Increment: 1	Vertical offset, relative to OffsetY, for the “white balance region of interest”, the region that is used for white balance calibration. (in pixels).

This feature is automatically updated if any of the following features is changed:
OffsetY

7.9.11 WhiteBalanceCalibrate | WO | E |

Color Only

Command	Start the white balance calibration in the “white balance region of interest”. A gain value for red, green and blue are calculated to obtain the right white balance. These gains are automatically set.

If the acquisistion is active, an image from the stream is used for the calculation. If the acquisition is inactive, the camera automatically starts an acquisition of one image.

7.9.12 WhiteBalanceStatus | RO | E |

Color Only

WhiteBalanceIdle	No white balance related information is available. Waiting for white balance functions to be executed.
WhiteBalanceStarted	The calibration of the white balance has been started.
WhiteBalanceCalibrateOk	The white balance calibration has successfully finished.
WhiteBalance_SensorTooDark	The sensor is too dark to perform a reliable white balance calibration.
WhiteBalance_ColorClipped	One of the color gains has clipped during calibration.

This feature is automatically updated if any of the following features is changed:
WhiteBalanceCalibrate

7.10 Sensor

Settings in the sensor chapter directly control sensor related settings.

NOTE: For each SensorBitDepth and ProgrammableGainAmplifier configuration option a different dark field and bright field set is selected due to a change in sensor characteristics. Each set can be recalibrated to optimize for the environmental conditions.

7.10.1 SensorBitDepth | RW | E |

Resolution_8_Bit	Set the sensor resolution in 8 bit. 8 Bit is required to achieve 187 fps.
Resolution_10_Bit	Set the sensor resolution in 10 bit.

This function controls the sensor ADC bit depth which is selectable between 8 and 10 bits per pixel. It is necessary to select 8 bit per pixel if a frame rate higher than 150 fps is required at full resolution. In the 8 bit resolution the maximum achievable frame rate is 187 fps. When 10 bit per pixel is selected the image quality is slightly better as compared to 8 bit ADC mode. For the 187 fps at full resolution also the PixelFormat needs to be set to 8 bit.

ISSUE 2 NOTE: This feature is automatically updated if any of the following features is changed: BinningHorizontal, BinningVertical

7.10.2 ProgrammableGainAmplifier | RW | G |

The programmable gain amplifier controls the analog gain in the sensor. Changing the sensors analog gain amplifier will affect the maximum full well capacity and read noise of the image sensor. Typical measured values with different analog gain settings are shown below.

Value	Definition	Max full well (e^-)	Readnoise (e^-)	Dynamic range (dB)
1	Unity gain	10000	15,9	56,0
2	2 x analog gain	5000	11,8	52,6
3	3 x analog gain	3400	10,8	49,9
4	4 x analog gain	2500	10,1	47,9

ISSUE 2 NOTE: This feature is automatically updated if any of the following features is changed: BinningHorizontal, BinningVertical

7.10.3 FrameOverheadTime | RO | E |

ISSUE 2 ONLY

Integer (ns)	This is the actual Frame Overhead Time (FOT) for the current camera configuration. For more details about the FOT and where to use this value for see section 7.6.10.1.

7.11 Factory

The factory settings are not user accessible, this mode is only required to adjust factory settings.

7.12 LUT Control

This group describes the Look Up Table features. In the camera an output look-up table (LUT) can be applied. When the table is programmed in the camera, it can be enabled with the *LUTEnable* feature. A Gamma curve can be enabled (issue 2 only) as a LUT or the LUT can be programmed by the following sequence of features:

- LUTStart
- $\text{LUTValue}_{\text{value\_index0}}$
- $\text{LUTValue}_{\text{value\_index1}}$
- ………………………………
- $\text{LUTValue}_{\text{value\_index1022}}$
- $\text{LUTValue}_{\text{value\_index1023}}$
- LUTEnd

Were value_indexn is a value between 0 and 1023 representing the output value of the LUT.
The exact amount of 1024 entries should be written, this can be verified with the *LUTStatus* feature.

7.12.1 LUTEnable | RW | E |

True	Activate the Look Up Table (LUT). The LUT transforms the video signal from the image processing to the output.
False	Deactivate the Look Up Table.

7.12.2 **LUTSelection** | RW | E |

ISSUE 2 ONLY

ProgrammableLUT	In this mode the LUT can be fully programmed.
Gamma	In this mode a gamma curve will be used as the LUT. The gamma will be applied by filling the LUT according to following formula: $$Output = 1023 \cdot \left(\frac{Input}{1023}\right)^{GammaFactor}$$ The Gamma factor can be set with the GammaFactor feature.

7.12.3 **LUTStart** | WO | E |

Command	Start the creation of a LUT. The exact amount of 1024 entries should be written using the LUTValue feature to successfully create a LUT.

7.12.4 **LUTValue** | RW | E |

0 to 1023, Increment: 1	After executing LUTStart, use this feature to consecutively write each index of the LUT. This feature also returns the value that is written at the LUT index selected with the LUTIndex feature.

This feature is automatically updated if any of the following features is changed:
LUTIndex

7.12.5 **LUTEnd**| WO | E |

Command	Finish the creation of a LUT. Make sure you wrote a value to all 1024 LUT entries.

7.12.6 **LUTIndex** | RW | E |

0 to 1023, Increment: 1	Select the LUT index for which you want to know the assigned value. The assigned value will be displayed in the LUTValue feature.

7.12.7 **LUTStatus** | RO | E |

LUT_Idle	LUT programming sequence in idle situation.
LUT_Started	LUT programming sequence started.
LUT_Restarted	LUT programming sequence restarted.
LUT_TooMuchEntries	Too much LUT entries (LUTValue) written before LUTEnd command is written.
LUT_NotEnoughEntries	Not enough LUT entries (LUTValue) written before LUTEnd command is written.
LUT_Stored	LUT programming sequence finished and stored in memory.
LUT_NotStarted	LUT programming sequence has not been started yet.

This feature is automatically updated if any of the following features is changed:
LUTStart, LUTEnd

7.12.8 **GammaFactor** | RW | E |

ISSUE 2 ONLY

0.01 to 10 Increment: 0.01	Set the Gamma factor.

7.12.9 **GammaFactorRaw** | RW | E |

ISSUE 2 ONLY

1 to 1000 Increment: 1	Set the Gamma factor.

7.13 Transport Layer Control

7.13.1 **PayloadSize** | RO | E |

Integer	Provides the number of bytes transferred for each image or chunk on the stream channel.

This feature is automatically updated if any of the following features is changed:
Width, Height, PixelFormat

7.14 Defect pixel

A defect pixel correction is available that replaces a defect pixel by a horizontally interpolated value or a vertically interpolated value or a horizontal nearest neighbor value depending on the local defect distribution. The correction method is illustrated in Figure 7-12. For a color camera the same method applies but with the difference that in this situation only pixels of the same color plane are regarded.

A factory list with defect pixels is stored in each camera. Additional defect pixels can be added to the list by the user. Up to 1000 unique defect pixel coordinates can be stored in non volatile memory.

	X	X	X	X	0	X	X	0	X	X	1	X	X	X	X
MASK	0	1	0	1	1	X	X	1	1	0	1	1	0	1	1
	X	X	X	X	0	X	X	0	X	X	X	X	X	1	X
	0	0	0	0	1/2	0	0	1/2	0	0	0	0	0	0	0
Correction	1/2	0	1/2	0	0	0	0	0	0	1	0	0	1	0	0
	0	0	0	0	1/2	0	0	1/2	0	0	0	0	0	0	0
	X	1	X	X	X	X	X	0	X	X	1	X	X	1	X
MASK	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
	X	X	X	X	1	X	X	1	X	X	0	X	X	1	X
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Correction	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Figure 7-12: Defect pixel correction method. Per tile, the center pixel correction method is indicated. In Mask, pixels marked by 1 are defect, pixels marked by 0 are not defect and irrelevant pixels are indicated by x. In the correction tile the weight factor per pixel is indicated.

7.14.1 How to use the defect pixel correction

This section describes some frequently occurring tasks with respect to the defect pixel correction functionality.

Task	Description
Add defect	The following steps are required to add a defect pixel to the defect pixel list: - Write the defect pixel location (origin of the most left top pixel has coordinate 0;0): *DefectPixelWriteX* and *DefectPixelWriteY*. - Execute *DefectPixelAdd* to add the defect pixel. If the marked defect is not existing and the defect list does contain less than 1000 defects, the defect is added to the non volatile defect list. Defects can be added in any order.
Remove defect	The following steps are required to remove a defect pixel from the defect pixel list: - Write the location of the defect pixel that should be removed: *DefectPixelWriteX* and *DefectPixelWriteY*. - Execute *DefectPixelRemove* for the removal of defect pixel. If existing, the defect is removed from the list in volatile memory. The defect pixel correction uses the updated list showing the removed defect location unmodified. Defects can be removed from the list in any order.
Obtain defect list	User can read the total number of defects from *DefectPixelTotal*. A defect location is obtained by writing the index to *DefectPixelSelect* and then read the coordinates from *DefectPixelReadX* and *DefectPixelReadY*. The index of the first defect in the table is 0.
Store user defect list	Execute *DefectPixelSave* to store the current defect list in non-volatile memory.
Restore user defect list	Execute *DefectPixelRestore* to restore the defect list to the stored user defect list. Defects that were added by the user and have not been saved are lost.
Restore factory default list	Execute *DefectPixelRestoreFactoryDefault* to restore the defect list to the original factory default defects. Defects that were added by the user and have not been saved are lost.
Clear defect list	Execute *DefectPixelClearAll* to clear the defect pixel list.

7.14.2 DefectPixelCorrectionEnable | RW | G |

True	Enable the defect pixel correction, the pixels as listed in the volatile memory will be corrected.
False	Disable the defect pixel correction.

7.14.3 **DefectPixelTestMode** | RW | E |

Off	Turn off the defect pixel test mode.
MarkDefectsWhiteOnVideo	Mark defects white on video, for use in a dark environment.
MarkDefectsBlackOnVideo	Mark defects black on video, for use in a light environment.
ShowDefectsAsWhiteOnBlackBackground	Generate a non-video test pattern that indicates the defect pixels.

7.14.4 **DefectPixelTotal** | RO | E |

Integer	Returns the total amount of pixels that will be corrected.

This feature is automatically updated if any of the following features is changed:
DefectPixelAdd, DefectPixelClearAll, DefectPixelRemove, DefectPixelRestore, DefectPixelRestoreFactory

7.14.5 **DefectPixelSelect** | RW | E |

0 to 999, Increment: 1	Select the defect pixel index from which you want to know the coordinates. The coordinates are displayed in the DefectPixelReadX and DefectPixelReadY feature.

This feature is automatically updated if any of the following features is changed:
DefectPixelAdd, DefectPixelClearAll, DefectPixelRemove, DefectPixelRestoreFactory

7.14.6 **DefectPixelReadX** | RO | E |

Integer	Returns the horizontal coordinate of the pixel selected by DefectPixelSelect.

This feature is automatically updated if any of the following features is changed:
DefectPixelAdd, DefectPixelClearAll, DefectPixelRemove, DefectPixelRestoreFactory, DefectPixelSelect

7.14.7 **DefectPixelReadY** | RO | E |

Integer	Returns the vertical coordinate of the pixel selected by DefectPixelSelect.

This feature is automatically updated if any of the following features is changed:
DefectPixelAdd, DefectPixelClearAll, DefectPixelRemove, DefectPixelRestoreFactory, DefectPixelSelect

7.14.8 **DefectPixelWriteX** | RW | E |

0 to 4096 Increment: 1	Select the horizontal coordinate of a defect pixel that needs to be corrected. Defect pixel coordinates x, y are referenced to the full sensor image of 4096 x 3072 pixels, where (0, 0) is the top-left most image pixel.

7.14.9 DefectPixelWriteY | RW | E |

0 to 3072 Increment: 1	Select the vertical coordinate of a defect pixel that needs to be corrected. Defect pixel coordinates x, y are referenced to the full sensor image of 4096 x 3072 pixels, where (0, 0) is the top-left most image pixel.

7.14.10 DefectPixelAdd | WO | E |

Command	Add the defect pixel determined by DefectPixelWriteX and DefectPixelWriteY to the defect pixel list in volatile memory.

7.14.11 DefectPixelRemove | WO | E |

Command	Remove the defect pixel determined by DefectPixelWriteX and DefectPixelWriteY from the defect pixel list in volatile memory.

7.14.12 DefectPixelClearAll | WO | E |

Command	Clear all defect pixels from the defect pixel list in volatile memory.

7.14.13 DefectPixelSave | WO | E |

Command	Save the defect pixel list from volatile memory as a user list to the non-volatile memory of the camera.

7.14.14 DefectPixelRestore | WO | E |

Command	Load the defect pixel user list from the non-volatile memory to the volatile memory. All existing defect pixel coordinates in the volatile memory will be lost.

7.14.15 DefectPixelRestoreFactory | WO | E |

Command	Load the factory default defect pixel list from the non-volatile memory to the volatile memory. All existing defect pixel coordinates in the volatile memory will be lost.

7.14.16 DefectPixelSaveAsFactoryDefault | WO | G |

Command	Not accessible by the user.

7.15 Dark field

Two independent dark field correction mechanisms are present in the camera, i.e. *DF_BlackClamp* and *DF_ColumnOffsetCorrection*.

7.15.1 DF_BlackClamp | RW | E |

True	Enable the dark field black clamp. There will be 8 pixels on each side of the image over the full sensor height that will appear dark. They are set to an electrical dark reference. These pixels are used for a row wise correction; they basically define for each row the black level. It will reduce row wise noise. The usable image resolution reduces to 4080x3072.
False	Disable the dark field black clamp. The full sensor resolution 4096x3072 can be used for imaging.

NOTE: By default DF_BlackClamp is true. When the black clamp is set to false please note that it is advised to perform a new dark field column correction with the DF_Calibrate feature.

7.15.2 DF_ColumnOffsetCorrection | RW | E |

True	Enable the dark field column offset correction. This correction compensates for dark signal non-uniformities (DSNU) in between columns. This correction is factory calibrated but can be re-calibrated in the field as well with *DF_Calibrate*.
False	Disable the dark field column offset correction.

A column offset correction calibration has to be performed in dark. During calibration the camera uses the internal timing generator to acquire images with an exposure time of 1 µs, the flash strobe output is inactive during calibration. A user calibration is automatically saved to non-volatile memory. It is always possible to restore to the factory calibrated correction.

Eight separate correction sets have to be stored for each *ProgrammableGainAmplifier* step (4 in total) in both *SensorBitDepth* modes. The calibration is only performed for the active configuration.

7.15.3 DF_Calibrate | WO | E |

Command	Start the dark field calibration of the camera. Perform this operation with no illumination on the camera. The calibrated correction will automatically be saved in the non-volatile memory.

NOTE: During calibration the camera uses the internal timing generator to acquire images with an integration time of 1 µs.

NOTE: The flash strobe output is inactive during calibration.

7.15.4 DF_Status | RO | E |

DF_CalibrateOK	The calibration is successfully finished and stored in non volatile memory.
DF_CalibrateError	An error occurred during calibration. A new calibration is required.
DF_SensorNotDark	The calibration could not be executed as the sensor is not dark.

This feature is automatically updated if any of the following features is changed:
DF_Calibrate

7.15.5 DF_RestoreFactory | WO | E |

Command	Restore the correction to the factory calibration status. Only the factory set in the selected operating condition will be loaded (*SensorBitDepth* in combination with the selected *ProgrammableGainAmplifier*).

7.15.6 DF_SaveAsFactoryDefault | WO | G |

Command	Not accessible by the user.

7.15.7 DF_IsUserCalibration | RO | E |

True	The user calibrated correction is active.
False	The factory calibrated correction is active.

This feature is automatically updated if any of the following features is changed:
BF_Calibrate, BF_RestoreFactory, BF_SaveAsFactoryDefault

7.16 Bright field

The bright field calibration measures and compensates for the pixel response non-uniformities (PRNU) between columns. This correction is factory calibrated but can be re-calibrated in the field.

It is important that the full image sensor resolution (4096x3072 pixels) is illuminated with a uniform scene because the calibration is always performed on the full resolution.

If the *ExposureMode* is configured to *Timed* the camera uses the internal timing generator to acquire images. The exposure time is adjusted to accomplish the configured video level in the *BF_CalibrationVideoLevel* feature. If the *BF_AutoLevelAdjust* is disabled the user should adjust the *ExposureTime* or illumination to achieve a usable calibration scene.

It is recommended to adjust the video level to 50-70% of the full scale in case of a bright field calibration. In case of a color camera, the color plane with the highest video level is adjusted to the configured set point. It is also important to use a white light source such that all color planes are illuminated sufficient.

During calibration the scene should be continuously illuminated or the light source should be synchronized with the flash strobe output.

If the *ExposureMode* is configured to *TriggerWidth*, *SyncControlMode* or *TimedTriggerControl* the video level could not be automatically adjusted, the user is in this case responsible for the adjustment of the illumination or exposure time.
After execution of the bright field calibration the calibration set is stored automatically in non-volatile memory. Eight separate correction sets have to be stored for each *ProgrammableGainAmplifier* step (4 in total) in both *SensorBitDepth* modes. The calibration is only performed for the active configuration.

7.16.1 BF_ColumnGainCorrection | RW | E |

True	Enable the bright field column gain correction.
False	Disable the bright field column gain correction.

7.16.2 BF_AutoLevelAdjust | RW | E |

True	Before performing the calibration the camera will adjust the integration time such that the video level will equal the level that is set in the BF_calibrationVideoLevel feature.
False	The integration time will not be adjusted.

This feature is automatically updated if any of the following features is changed:
ExposureMode

NOTE: This feature is only available when ExposureMode is set to Timed.

NOTE: For a color camera, the color plane with highest amplitude is adjusted to the configured set point.

7.16.3 BF_CalibrationVideoLevel | RW | E |

10 to 90, Increment: 1	Set the target video level in a percentage of the full scale at which the bright field calibration will take place.

7.16.4 BF_OutputImagesDuringCalibration | RW | E |

True	The camera will output the images that are acquired during calibration. 1 for the Bright field calibration and up to 10 for the auto level adjustment when BF_AutoLevelAdjust is enabled.
False	The camera does not output the images that are acquired during calibration.

7.16.5 BF_Calibrate | WO | E |

Command	Start the bright field calibration of the camera. Perform this operation with uniform constant illumination on the camera. Flash strobe control can be used as well. The calibrated correction will automatically be saved in the non-volatile memory.

NOTE: If the camera is running in a triggered mode, the illumination shall be adjusted by the user. The light source may be either constant or flashed synchronously with the applied trigger.

NOTE: Before performing a bright field calibration, first perform a dark field calibration.

7.16.6 BF_Status | RO | E |

BF_CalibrateOK	The calibration is successfully finished and stored in non volatile memory.
BF_UnderExposed	Calibration failed, configured video level cannot be reached with adjusted integration time. Increase exposure level and recalibrate.
BF_OverExposed	Calibration failed, configured video level cannot be reached with adjusted integration time. Decrease exposure level and recalibrate.
BF_UnstableExposure	Calibration failed, adjustment in illumination is required.
BF_CalibrateError	An error occurred during calibration. A new calibration is required.
BF_WrongExposureMode	Calibration not executed because wrong exposure mode is selected.

This feature is automatically updated if any of the following features is changed:
BF_Calibrate

7.16.7 BF_RestoreFactory | WO | E |

Command	Restore the correction to the factory calibration status. Only the factory set in the selected operating condition will be loaded (*SensorBitDepth* in combination with the selected *ProgrammableGainAmplifier*).

7.16.8 BF_SaveAsFactoryDefault | WO | G |

Command	Not accessible by the user.

7.16.9 BF_IsUserCalibration | RO | E |

True	The user calibrated correction is active.
False	The factory calibrated correction is active.

This feature is automatically updated if any of the following features is changed:
BF_Calibrate, BF_RestoreFactory, BF_SaveAsFactoryDefault

7.17 LF FF Calibration

ISSUE 2 MONOCHROME ONLY
This function corrects for a spatial low frequency non-uniformity caused by the illumination conditions. Figure 7-13 illustrates the effect before and after enabling the Low Frequency Flat Field Correction (LF-FFC) function. Multiple correction sets can be stored in the camera and different sets can be applied to subsequent frames.

NOTE: LF FFC cannot be used when in band mode or in 2x2 binning mode.

7.17.1 LF_FF_Correction | RW | B |

True	Enables the spatially low frequency flat field correction.
False	Disables the spatially low frequency flat field correction.

This feature is automatically updated if any of the following features is changed:
BandEnable

7.17.2 LF_FF_ReferencePoint | RW | B |

Peak	Position of the reference video level after FFC calibration will equal the maximum video level present during FFC calibration. See Figure 7-13a.
Average	Position of the reference video level after FFC calibration will equal the average video level present during FFC calibration. See Figure 7-13b.

7.17.3 LF_FF_SetSelectionMode | RW | B |

Manual	In this mode the active FFC set is selected manually with the LF_FF_SetSelector feature.
Sequencer	In this mode the active FFC set is selected by the sequencer. The sequencer advances one entry per acquired image.

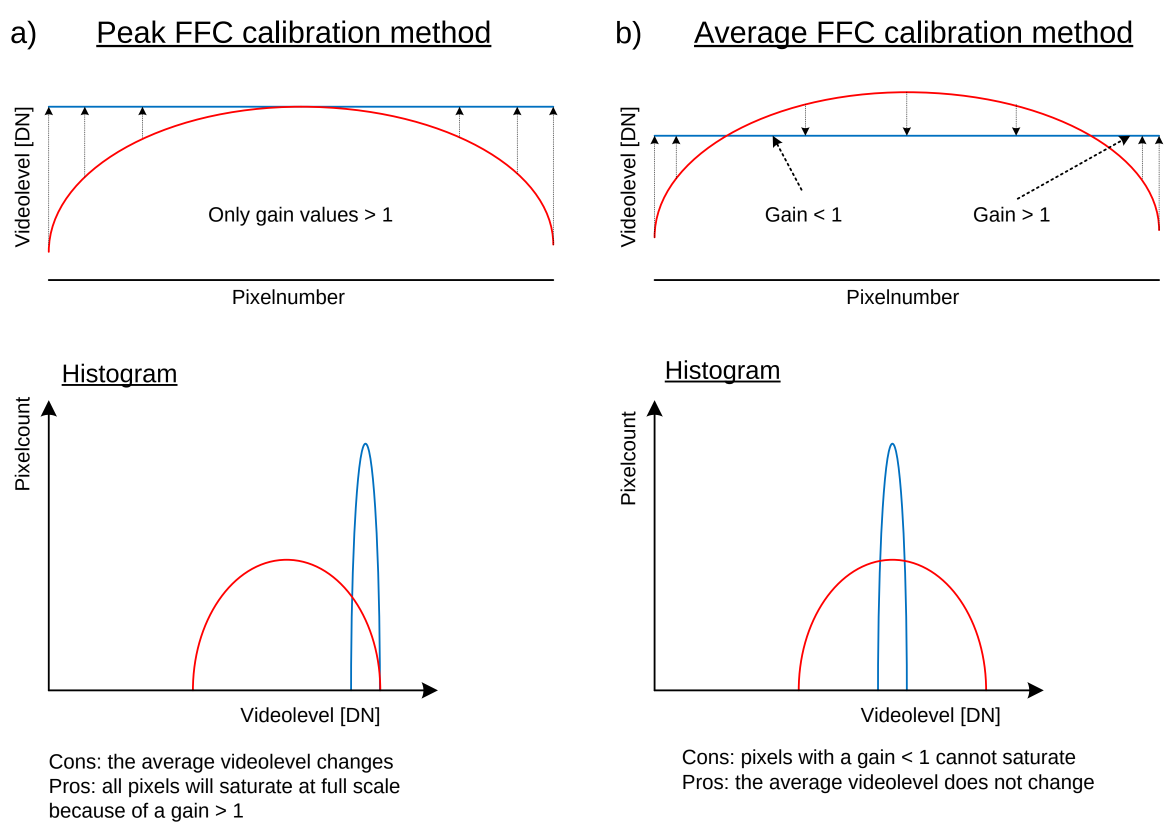
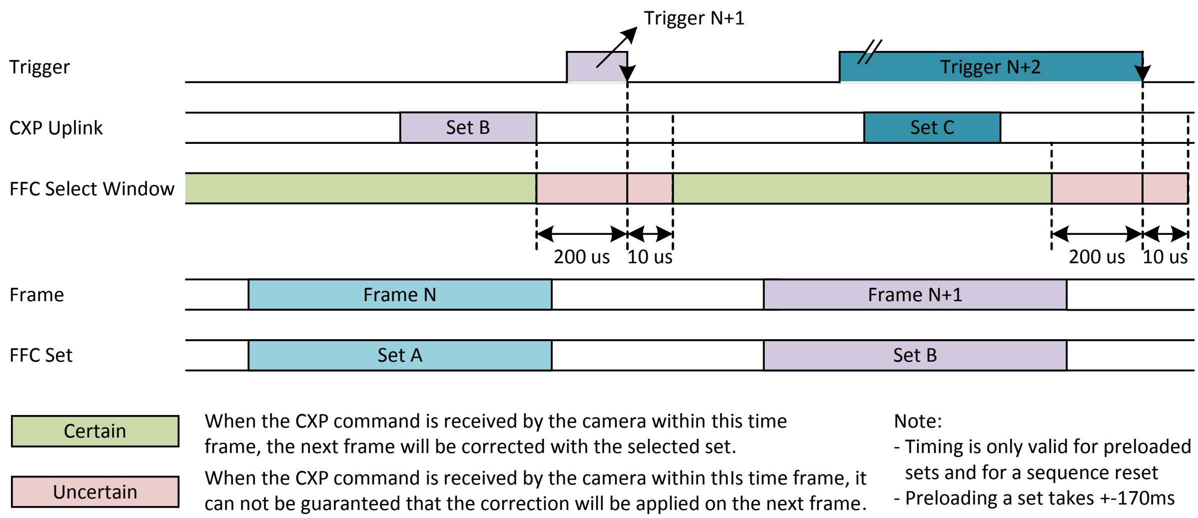


Figure 7-13: A visual representation of the low frequency non uniformity (red line) and how the LF-FFC function can correct for this (blue line). The LF-FFC function offers the possibility to have the flat field equal the maximum video level (a) or the average video level (b).



7.17.4 LF_FF_ManualSetSelector | RW | B |

0 to 49, Increment: 1	Select the active Flat Field correction set. The last 18 sets that where loaded will be stored in a cache memory. Sets that are present in cache memory can be loaded as active set in between two subsequent frames. For timing details see Figure 7-14. This feature is only available for LF_FFSetSelectionMode = Manual

NOTE: Rapid switching of sets without additional delay requires the selected sets to be present in cache. This can be accomplished *before* image acquisition by sequentially selecting the sets to be used, or by programming a sequence containing the sets to be used.

7.17.5 LF_FF_CalibrationROI | RW | B |

UserROI	The FF calibration is performed on the user ROI.
SensorROI	The FF calibration is performed on the full sensor.

Figure 7-14: Timing diagram for the FFC function. If the command to change the correction set arrives in the green timeframes it will be applied to the next frame.

7.17.6 LF_FF_Calibrate | WO | B |

Command	Execute the FF calibration. A successful result is stored in the set indicated by LF_FF_ManualSetSelector. This feature is only available for LF_FFSetSelectionMode = Manual.

7.17.7 LF_FF_ResetCurrentSet | WO | B |

Command	The current selected set is being reset. The gain is set to 1 for the whole image. The set can thus be used as if the FFC function is off. This feature is only available for LF_FFSetSelectionMode = Manual.

7.17.8 LF_FF_CalibrationStatus | RO | B |

LF_FF_CalibrateOK	The FF calibration went successful.
LF_FF_CalibrationIdle	The FF calibration is not running and waiting for a new command.

LF_FF_ImageTooDark	The average video level is below 12.5% of the full scale.
LF_FF_ImageTooBright	The average video level is above 90% of the full scale.
LF_FF_ImagePeakTooLow	The peak video level is below 12.5% of the full scale.
LF_FF_ImagePeakTooHigh	The peak video level is above 90% of the full scale.
LF_FF_CalibrationError	The camera was unable to grab the images required for the calibration.
LF_FF_GainClipped_Warning	The gain required to obtain a flat field in a part of the image is above the gain limit of 2.75. A calibration set is still stored, however part of the image might have a lower reference level compared to the rest of the image.

This feature is automatically updated if any of the following features is changed:
LF_FF_Calibrate

7.17.9 **LF_FF_SequenceStartProgram** | WO | B |

Command	Initialize sequencer programming.

7.17.10 **LF_FF_SequenceSet** | RW | B |

0 to 49, Increment: 1	After activating LF_FF_SequenceStartProgram, write here the numbers of the calibration sets that are required in the sequence. Write the numbers to the camera one by one and in the right order. The same number and corresponding set can be used multiple times. Due to the limited cache size, only 18 unique sets can be loaded into the sequencer.

This feature is automatically updated if any of the following features is changed:
LF_FF_SequenceLoad, LF_FF_SequenceReadCurrentSet, LF_FF_SequenceReadSelectedIndex

NOTE: Although only up to 18 unique sets can be loaded into the sequence, the sequence itself can contain up to 2000 sets.

7.17.11 **LF_FF_SequenceStopProgram** | WO | B |

Command	Finalize sequencer programming.

7.17.12 **LF_FF_SequenceStatus** | RO | B |

LF_FF_SequenceProgramIdle	The FF sequence program is not running and waiting for a new command.
LF_FF_SequenceProgramming	The sequence is being programmed.
LF_FF_SequenceProgramFinished	The sequence is successfully programmed.
LF_FF_SequenceProgramFailed	There are more unique sets loaded into the sequence than allowed. For mono this limit is 18.

LF_FF_SequenceProgrammingNotStarted	The sequence programming has not yet started.
LF_FF_SequenceProgramLimitReached	The maximum sequence length of 2000 entries has been reached.
LF_FF_SequenceSaved	The current sequence has been saved to non-volatile memory.
LF_FF_SequenceLoaded	The sequence that was stored in the non-volatile memory has been loaded as the active sequence.
LF_FF_SequenceLoadFailed	The sequence that was stored in the non-volatile memory could not be loaded.

This feature is automatically updated if any of the following features is changed:
LF_FF_SequenceLoad, LF_FF_SequenceSave, LF_FF_SequenceSet, LF_FF_SequenceStartProgram, LF_FF_SequenceStopProgram

7.17.13 **LF_FF_SequenceTotal** | RO | B |

Integer	Returns the total amount of sequence entries that have been written.

This feature is automatically updated if any of the following features is changed:
LF_FF_SequenceLoad, LF_FF_SequenceSet, LF_FF_SequenceStartProgram

7.17.14 **LF_FF_SequenceSave** | WO | B |

Command	Save the programmed sequence to non-volatile memory.

7.17.15 **LF_FF_SequenceLoad** | WO | B |

Command	Load the programmed sequence from non-volatile memory.

7.17.16 **LF_FF_SequenceReset** | WO | B |

Command	Reset the sequencer pointer to the first step during sequenced correction.

7.17.17 **LF_FF_SequenceRepeat |** RW | B |

True	Every time the sequencer pointer reaches the last sequence entry, the sequence will be reset and started again from the first sequence entry.
False	After reaching the last sequence entry, the sequence stops and the last sequence entry remains the active FF correction set.

7.17.18 **LF_FF_ SequenceReadCurrentSet** | WO | B |

Command	Reads the set number which is currently used for correction. The value that is read is returned in the “LF_FF_SequenceSet” feature.

7.17.19 **LF_FF_ SequenceIndexSelector** | RW | B |

0 to 1999, Increment: 1	Select the sequence index that will be read with LF_FF_SequenceReadSelectedIndex command.

This feature is automatically updated if any of the following features is changed:
LF_FF_SequenceLoad

7.17.20 **LF_FF_SequenceReadSelectedIndex** | WO | B |

Command	Returns the set number of the sequence index position indicated by LF_FF_SequenceIndexSelector. The value that is read is returned in the “LF_FF_SequenceSet” feature.

7.17.21 **LF_FF_ViewCorrection** | RW | B |

True	The active spatially low frequency flat field correction set will be shown. In 8 bit, a correction factor of 1 equals a grey level of 64. In 10 bit, a correction factor of 1 equals a grey level of 256.
False	The normal camera output will be shown.

7.17.22 **LF_FF_ViewMetaData** | RW | B |

True	Enable the inclusion of meta data in the first 7 pixels in the top left corner. See Figure 7-15 for the data encoding format.
False	Disable the inclusion of meta data in the first pixel row.

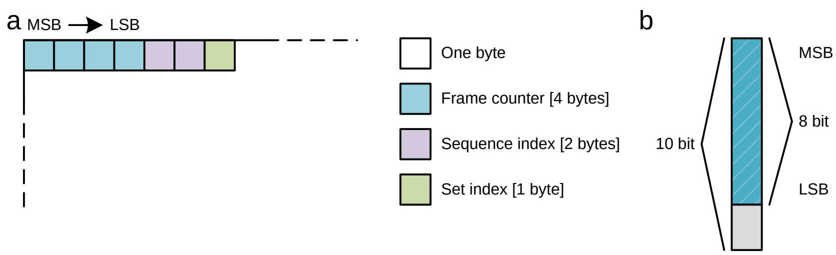


Figure 7-15: (a) MetaData encoding format. The first 4 pixels are used to encode the frame number. The next 2 pixels are used to encode the sequence index. The 7th pixel in the first row encodes the set index. (b) In each pixel 8 bits are used for the encoding. In the 10 bit pixel format the 8 most significant bits will be used.

7.17.23 How to transfer a low frequency flat field calibration set from one camera to another one.

The camera has functionality to read out a low frequency flat field calibration set to a computer. A write function to write a low frequency flat field calibration to a camera is available as well. Together these two functions can be used to transfer a low frequency flat field calibration from one camera to another one or from one set index position to another set index.

NOTE: This functionality is not suitable to write a custom correction set to the camera.

NOTE: The set that is written to the camera has to originate from another Adimec camera with the same sensor type.

The following steps have to be followed to complete this task.

1. Connect the camera from which you want to transfer the calibration set.
2. Set **LF_FF_SetSelectionMode** to **Manual**.
3. Select the set you want to transfer with **LF_FF_ManualSetSelector**.
4. Start the reading process by executing **LF_FF_RawSetDataReadStart**.

5. Read the raw set data from the camera by requesting 887x **LF_FF_RawSetData** for data. The data format is 64 bit.

NOTE: Sending too many or too few data requests to **LF_FF_RawSetData** will cause an error.

6. Finish the reading process by executing **LF_FF_RawSetDataEnd**.
7. Check if no errors occurred with **LF_FF_RawSetDataStatus**.
8. Write the data to a file.

NOTE: If you only want to write the calibration set to a different set index in the same camera it is already sufficient to store the data in a variable. For an example script that uses a variable please refer to APPENDIX B: C++ Example, transfer an LF FF calibration set.

9. Connect the camera to which you want to write the file to.
10. Set **LF_FF_SetSelectionMode** to **Manual**.
11. Select the set index to which you want to write the calibration set with **LF_FF_ManualSetSelector**.
12. Start the writing process by executing **LF_FF_RawSetDataWriteStart**.
13. Write the raw set data from the file created in step 8 to the camera by sending 887x data to **LF_FF_RawSetData**. The data format should be 64 bit.

NOTE: Sending too many or too few data to **LF_FF_RawSetData** will cause an error.

14. Finish the writing process by executing **LF_FF_RawSetDataEnd**.
15. Check if no errors occurred with **LF_FF_RawSetDataStatus**.

This finishes the process of transferring a calibration set to a different camera.

7.17.24 **LF_FF_RawSetDataWriteStart** | WO | B |

Command	Start the raw data transfer of an LF_FFC set to the camera. The data will be stored in the set as selected with LF_FF_ManualSetSelector.

7.17.25 **LF_FF_RawSetDataReadStart** | WO | B |

Command	Start the raw data transfer of a LF_FFC set from the camera. The data will be read from the set as selected with LF_FF_ManualSetSelector.

7.17.26 **LF_FF_RawSetDataEnd** | WO | B |

Command	End the raw data transfer of a LF_FFC set.

7.17.27 **LF_FF_RawSetData** | RW | B |

Integer	Raw data feature for reading or writing an LF_FFC set. In the writing or reading process it is required to send data to or read data from this feature 887x. The data format is 64 bit.

7.17.28 **LF_FF_RawSetDataStatus** | RO | B |

LF_FF_RawSetDataOK	The raw set data is read/written correctly.
LF_FF_RawSetDataReading	The raw set data is being read.
LF_FF_RawSetDataWriting	The raw set data is being written.

LF_FF_RawSetDataSizeNotCorrect	The size of the written set data is not correct.
LF_FF_RawSetDataCRCerror	A CRC error occurred while transferring the raw set data.

This feature is automatically updated if any of the following features is changed:
LF_FF_RawSetDataEnd, LF_FF_RawSetDataReadStart, LF_FF_RawSetDataWriteStart

7.18 User set control

At delivery, the factory settings are stored as the power-up default settings. The user may however change the power-up settings without losing the factory default settings. Therefore a copy of the factory default settings is stored in the camera.

7.18.1 UserSetSelector | RW | B |

Default	This is the factory default set. The user cannot change this set.
UserSet1	This is the set the user can save. This set is automatically loaded at power up.

This feature is automatically updated if any of the following features is changed:
UserSetLoad, UserSetSave

Table 7-7 lists the features and their default values that are stored in the factory default set.

7.18.2 UserSetLoad | WO | B |

command	Activate the set that is selected with UserSetSelector.

NOTE: Loading a set requires acquisition to be stopped.

7.18.3 UserSetSave | WO | B |

Command	If UserSet1 is selected in the UserSetSelector, this feature can be used to save the current camera settings in UserSet1.

Table 7-7: User savable features and default values

Group	Feature	Default	Remark
DeviceControl	DeviceUserID	<empty>	
	DeviceIndicatorMode	Active	
ImageFormatControl	Width	4096	
	Height	3072	
	OffsetX	0	
	OffsetY	0	
	ReverseX	0	
	ReverseY	0	
	PixelFormat	Mono10/ BayerGB10	
	TestImageSelector	0	
	TestImageVideoLevel	0	
	CrossHairOverlay	0	
	BinningHorizontal	1	
	BinningVertical	1	
	BinningMode	Average	
AcquisitionControl	AcquisitionFramePeriodRaw	100000	10 fps
	TriggerSource	Trigger	
	TriggerActivation	RisingEdge	
	ExposureMode	Timed	
	ExposureTimeRaw	5000	
	InterfaceUtilization	100	
CounterAndTimerControl	FlashStrobeMode	Disabled	
	FlashStrobeDurationRaw	250	
	FlashStrobeDelayRaw	0	
	FlashStrobeActiveState	Conducting	
HDR	HDR_Mode	Off	
	MultiSlopeLevel	33	%
	MultiSlopeTime	33	%
AnalogControl	All, Red, Green, Blue	1000	1x

	BlackLevelRaw	20	
	WhiteBalanceWidth	4096	
	WhiteBalanceHeight	3072	
	WhiteBalanceOffsetX	0	
	WhiteBalanceOffsetY	0	
Sensor	SensorBitDepth	Resolution_10_Bit	
	ProgrammableGainAmplifier	1	
LUTControl	LUTEnable	0	
	LUTSelection	ProgrammableLUT	
	GammaFactorRaw	45	
DefectPixel	DefectPixelCorrectionEnable	1	
	DefectPixelTestMode	Off	
DarkField	DF_BlackClamp	1	
	DF_ColumnOffsetCorrection	1	
BrightField	BF_ColumnGainCorrection	1	
	BF_AutoLevelAdjust	0	
	BF_CalibrationVideoLevel	65	%
	BF_OutputImagesDuring Calibration	0	
LF_FF_Calibration	LF_FF_ReferencePoint	Peak	
	LF_FF_SetSelectionMode	Manual	
	LF_FF_ManualSetSelector	0	
	LF_FF_CalibrationROI	SensorROI	
	LF_FF_SequenceRepeat	1	
	LF_FF_ViewMetaData	0	
Band	BandEnable	0	

7.19 Band

With the band function rectangular areas for sensor read out can be selected. By using the band function not every sensor line has to be read out which might increase the frame rate. All created bands will be combined into a single image at the interface output.

7.19.1 How to use the band function

This section describes some frequently occurring tasks with respect to the band functionality.

Task	Description
Add band	User writes *BandWriteOffsetY* and *BandWriteHeight*. Addition is executed after writing to *BandAdd*. If the band is not overlapping another and within the sensor dimensions, the band is added to the list. The list is sorted ascending on vertical start position. The camera uses the updated list for the resulting image output. Bands can be added in any order. The number of bands is limited to 32.
Remove band	User writes *BandWriteOffsetY*. Removal is executed after writing to *BandRemove*. If a band with the indicted *BandWriteOffsetY* exists, it is removed from the list in volatile memory. The camera uses the updated list. Bands can be removed from the list in any order.
Obtain Band list	User can read number of bands from *BandTotal*. A band is obtained by writing the index to *BandSelector* and then read the position and height from *BandReadOffsetY and BandReadHeight.* The index of the first band in the table is 1.
Store band list	The current band list is stored to non-volatile memory by writing to Band*Save*.
Restore band list	The band list is restored from non-volatile memory by writing to *BandLoad*. Bands that were added and have not been saved are lost.
Clear list	The band list is cleared by writing to *BandClearAll*. The camera will operate in ROI mode.

7.19.2 BandEnable | RW | E |

True	Enable the band functionality.
False	Disable the band functionality.

7.19.3 BandTotal | RO | E |

Integer	Returns the number of bands created.

This feature is automatically updated if any of the following features is changed:
BandAdd, BandClearAll, BandRemove, BandLoad

7.19.4 BandSelector | RW | E |

1 to 32, Increment: 1	Select a band from which you want to read back the offset and height.

7.19.5 BandReadOffsetY | RO | E |

Integer	Returns the y offset of the band selected by BandSelector.

This feature is automatically updated if any of the following features is changed:
BandAdd, BandClearAll, BandRemove, BandLoad, BandSelector

7.19.6 BandReadHeight | RO | B |

Integer	Returns the height of the band selected by BandSelector.

This feature is automatically updated if any of the following features is changed:
BandAdd, BandClearAll, BandRemove, BandLoad, BandSelector

7.19.7 BandWriteOffsetY | RW | B |

Integer Increment: 4	Set the y offset of the band to be added or to be removed. When binning the increment has to be divided by 2.

7.19.8 BandWriteHeight | RW | B |

Integer Increment: 8	Set the height of the band to be added or to be removed. When binning the increment has to be divided by 2.

7.19.9 BandAdd | WO | E |

Command	Add the band determined by BandWriteOffsetY and BandWriteHeight to non volatile memory.

7.19.10 BandRemove | WO | B |

Command	Remove the band determined by BandWriteOffsetY from non-volatile memory.

7.19.11 BandClearAll | WO | B |

command	Remove all bands from non volatile memory.

7.19.12 BandLoad | WO | B |

command	Restore the band user list to the device and make it active.

7.19.13 BandSave | WO | B |

command	Save the current band list as a user list to the non-volatile memory of the device.

7.19.14 BandAvailableOffsetPlusHeight | RO | E |

Integer	Returns the actual height available for bands. BandOffsetY + BandHeight should not exceed BandAvailableHeight.

This feature is automatically updated if any of the following features is changed:
OffsetY

7.19.15 **BandStatus |** RO **|** B **|**

BandStatusNoError	No error occurred in programming the bands.
BandStatusOutsideAvailableHeight	The band cannot be created as it is outside the available height.
BandStatusOverlap	The band overlaps with an already existing band.
BandStatusCouldNotAdd	The band could not be added to the band list.
BandStatusCouldNotRemove	The band could not be removed from the band list.
BandStatusRestoreError	The user band list could not be loaded from non-volatile memory.

This feature is automatically updated if any of the following features is changed:
BandAdd, BandClearAll, BandRemove, BandLoad, BandSave, BandSelector

APPENDIX A: CMOS SENSOR CLEANING INSTRUCTIONS

When you would like to clean the CMOS sensor because the sensor got contaminated with dust particles that influence your image quality, this appendix describes the right procedure with the lowest chance on damage. However, due to the high risk of sensor damage it is strongly advised to only perform cleaning when it is really necessary and cannot be avoided.

NOTE: Damage of the CMOS sensor due to scratches on the cover glass or ESD is not covered by warranty!

The correct working environment for cleaning is essential in order to ease cleaning and to prevent damage of the CMOS sensor. Especially take care of the following precautions.

Precautions:

- Take precautions to prevent ESD that can damage the CMOS sensor.
- Never try to clean the CMOS sensor at a relative humidity lower than 30%. A relative humidity of 40% or higher is preferred in order to minimize the chance of damage due to ESD.
- It is advisable to use an ionizer, in order to minimize the built-up of ESD.
- Cleaning of the CMOS sensor and lens assembly is preferably performed in a clean room or clean bench.
- Use non-fluffing Q-tips and Alcohol (or Hexane) for cleaning. De-ionized water may be necessary to remove ionic contaminants like salts.
- Any Q-tip should be used only once - you will otherwise move dirt from one place to another.
- Be sure to clean the lens mount of the lens before assembly.
- Never dry rub the window. This may cause static charges or scratches that can damage the CMOS sensor.

Cleaning instructions:

1. First try to remove the contamination by using clean, dry air. (Use an ultra-filtered, non-residue dust remover spray). Avoid blowing air into the screw thread of the lens mount, because this may cause contamination on the CMOS sensor due to loose particles and traces of oil or grease.

   If this step does not result in an acceptable result, continue with step 2.

2. Remove the lens mount by unscrewing the 4 screws (m3x6 screws; torx10) that hold the lens mount (see Figure A-1).
3. Clean the CMOS sensor cover glass using alcohol or hexane and a Q-tip. Gently and carefully rub the window always in the same direction, e.g. top - down.
4. Install the lens mount back on the camera.

   Make sure that the maximum tightening force may not exceed 108 cNm.

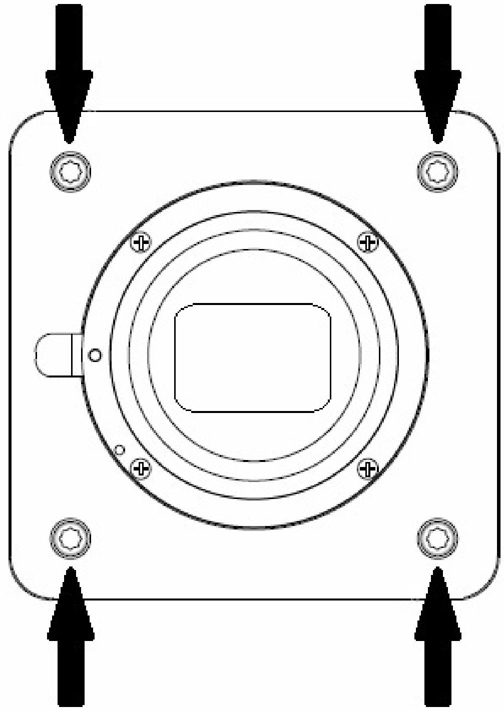


Figure A-0-1: Front view of the camera. The lens mount fixation points are indicated with the black arrows.

5. Install a lens, power up the camera, set the lens at a small aperture (F16) and point the lens at a bright source. Adjust gain and integration time if necessary.

6. Check the image on the monitor for dark spots and stripes caused by contamination on the CMOS sensor cover glass. (Note that the image on the monitor should not saturate due to overexposure - if necessary close the iris even further).

If the CMOS sensor is not clean, repeat steps 2 - 6 using a new Q-tip. After three unsuccessful tries, it is advised to wait a few minutes before a new attempt is made. The waiting time allows the electric charge that has been built up during cleaning to neutralize.

APPENDIX B: C++ EXAMPLE, TRANSFER AN LF FF CALIBRATION SET

This appendix shows an example script that moves a low frequency flat field calibration set to a set with a different index in the same camera. With small adjustments it can be used to transfer a calibration set between different Adimec cameras with the same sensor.

In this example, set 10 is copied to set 7.

This script only works with a MATROX frame grabber. It uses a MATROX library to control the frame grabber. Other frame grabbers use different libraries. This example, together with your frame grabber documentation should provide sufficient information to achieve a similar result.

```
/*******************************************************************************/
/*
 * File name: moveLFFFSet.cpp
 *
 * Synopsis:  This program demonstrates how to move one LF FF set to another one.
 *
 */

#include <mil.h>
#include <stdint.h> /* Required for uint64_t support */

/* Determine the number of required transfers for the LF FFC set,
 * S-50A30:     TRANSFERS = 3083
 * S-25A70/80:  TRANSFERS = 1767
 * Q-12A180:    TRANSFERS = 887
 */

#define TRANSFERS 887

uint64_t lf_ff_data[TRANSFERS];
char     Cmd[256];
char     CmdParam[65];
char   myString[256];
int         i;

int MosMain(void)
{
      /* Start camera discovery */
      /* Allocate defaults. */

      MIL_ID MilApplication,  /* Application identifier.  */
           MilSystem,       /* System identifier.       */
           MilDigitizer,    /* Digitizer identifier.    */
           MilImage;        /* Image buffer identifier. */

   MappAllocDefault(M_DEFAULT, &MilApplication, &MilSystem,
                          M_NULL, &MilDigitizer, &MilImage);

   /* End camera discovery */
   /* Start LF FFC operation */

   /* Set manual set selector to manual */
   MosPrintf(MIL_TEXT("LF_SetSelectionMode is set to Manual"));
   sprintf_s(Cmd, "LF_FF_SetSelectionMode");
   sprintf_s(CmdParam, "Manual");
   MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

   /* Select the set you want to read. In this example set 10. */
   MosPrintf(MIL_TEXT("Set 10 is selected"));
   sprintf_s(Cmd, "LF_FF_ManualSetSelector");
   sprintf_s(CmdParam, "10");
   MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

   /* Start the reading process */
   MosPrintf(MIL_TEXT("Reading data..."));
   sprintf_s(Cmd, "LF_FF_RawSetDataReadStart");
   sprintf_s(CmdParam, "1");
   MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

   /* Read the data */
```

```
    sprintf_s(Cmd, "LF_FF_RawSetData");
    for (i = 0; i < TRANSFERS; i++) {
          MdigInquireFeature(MilDigitizer, M_FEATURE_VALUE, Cmd, M_TYPE_INT64, &lf_ff_data[i]);
          printf("Record: %i - %lli\n", i, lf_ff_data[i]);
    }

    /* End the reading process */
    sprintf_s(Cmd, "LF_FF_RawSetDataEnd");
    sprintf_s(CmdParam, "1");
    MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

    /* Check if everything went fine and no errors occured */
    sprintf_s(Cmd, "LF_FF_RawSetDataStatus");
    MdigInquireFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_DEFAULT, myString);
    printf("Status: %s\n", myString);
    MosPrintf(MIL_TEXT("Reading data finished"));

    /* Select the set you want to write to. In this example set 7. */
    MosPrintf(MIL_TEXT("Set 7 is selected. \n\n"));
    sprintf_s(Cmd, "LF_FF_ManualSetSelector");
    sprintf_s(CmdParam, "7");
    MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

    /* Start the writing process */
    MosPrintf(MIL_TEXT("Writing data..."));
    sprintf_s(Cmd, "LF_FF_RawSetDataWriteStart");
    sprintf_s(CmdParam, "1");
    MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

    /* Write the data */
    sprintf_s(Cmd, "LF_FF_RawSetData");
    for (i = 0; i < TRANSFERS; i++) {
          MdigControlFeature(MilDigitizer, M_FEATURE_VALUE, Cmd, M_TYPE_INT64, &lf_ff_data[i]);
          printf("Record: %i\n", i);
    }

    /* End the writing process */
    sprintf_s(Cmd, "LF_FF_RawSetDataEnd");
    sprintf_s(CmdParam, "1");
    MdigControlFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_TYPE_STRING, CmdParam);

    /* Check if everything went fine and no errors occured */
    sprintf_s(Cmd, "LF_FF_RawSetDataStatus");
    MdigInquireFeature(MilDigitizer, M_FEATURE_VALUE_AS_STRING, Cmd, M_DEFAULT, myString);
    printf("Status: %s\n", myString);
    MosPrintf(MIL_TEXT("Writing data finished"));

    /* End LF FFC operation */
    /* Free defaults. */

    MappFreeDefault(MilApplication, MilSystem, M_NULL, MilDigitizer, MilImage);

    return 0;
}
```