

EoSens Cube6&7 and mini1&2

HighSpeed Memory Camera With GigEVision Interface



GigEVision GenICam Reference Manual

EoSens Cube6&7 and mini1&2 Camera Manual

Rev. 1.3

Copyright © 2009 Mikrotron GmbH

Mikrotron GmbH
Landshuter Str. 20-22
D-85716 Unterschleissheim
Germany

Tel.: +49 89 726342 00
Fax: +49 89 726342 99
info@mikrotron.de
www.mikrotron.de



Table of contents

| | | |
|------|---|----|
| 1 | Introduction..... | 3 |
| 1.1 | General Target..... | 3 |
| 1.2 | GEN<i>CAM Specification..... | 4 |
| 1.3 | Interface Type..... | 4 |
| 2 | Feature description..... | 5 |
| 2.1. | Device Information..... | 5 |
| 2.1 | Image Size Control..... | 8 |
| 2.2 | Acquisition and Trigger Controls..... | 14 |
| 2.3 | Recording (Custom Features)..... | 18 |
| 2.4 | Streaming (Custom Features)..... | 24 |
| 2.5 | Analog Controls..... | 28 |
| 2.6 | Multiple Slope (Custom Features)..... | 29 |
| 2.7 | ExtTrigger (Custom Features)..... | 30 |
| 2.8 | ImageBlitz (Custom Features)..... | 31 |
| 2.9 | Synchronisation (Custom Features)..... | 35 |
| 2.10 | Custom Features..... | 37 |
| 2.11 | CustomData (Custom Features)..... | 39 |
| 2.12 | UserSets..... | 39 |
| 3 | General Programming Guide..... | 41 |
| 3.1 | Camera States..... | 41 |
| 3.2 | State Transformations..... | 42 |
| 4 | Record Programming Guide..... | 44 |
| | Continuous Recording, one sequence..... | 44 |
| 4.1 | Continuous Recording with more than one sequence..... | 46 |
| 4.2 | Single Recording..... | 47 |
| 4.3 | Triggered Recording..... | 49 |
| 4.4 | RecordStandby (UserSets)..... | 49 |
| 5 | Streaming Programming Guide..... | 50 |
| 6 | ImageBLITZ® Programming Guide..... | 52 |
| 7 | DecimationXY mode (Subsampling)..... | 53 |
| 8 | Feature overview..... | 54 |

1 Introduction

Today's digital cameras are packed with much more functionality than just delivering an image. Processing the image and appending the results to the image data stream, controlling external hardware, and doing the real-time part of the application have become common tasks for machine vision cameras. As a result, the programming interface for cameras has become more and more complex.

The goal of GenICam is to provide a generic programming interface for all kinds of cameras. No matter what interface technology (GigE Vision, Camera Link, 1394 DCAM, USB, etc.) they are using or what features they are implementing, the application programming interface (API) should be always the same.

GEN<i>CAM as a standard for the definition of software and hardware (cameras) is becoming increasingly important.

Our cameras of the production line EoSens Cube6&7 and mini1&2 are GigE Vision compliant cameras. The GigE Vision standard uses Ethernet as the hardware interface to connect a camera to its host. GigE Vision is based on GEN<i>CAM which ensures compatibility to a wide range of programming interfaces and applications.

The description of this manual covers the Mikrotron cameras from the series

EoSens Cube6, EoSens Cube7, EoSens mini1, EoSens mini2

1.1 General Target

The configuration, status and control capabilities of the cameras are accessible by a number of camera 'features'. This manual gives you an overview of the features defined in the MC136x and MC137x. Special functions of the cameras will be described in detail to give you all informations you may need to realize your software projects with our cameras.

There are a number of standard features, which are common to all GigE Vision compliant cameras. For example, there are features to configure the communication over the Ethernet interface. Not all of these features are described in this manual. They are described in detail in the document 'GEN<i>CAM Standard Features Naming Convention' which can be downloaded from the homepage of the GEN<i>CAM organisation (<http://www.genicam.org/>).

Please use a GEV-Viewer to show all possible features of the camera (e.g. GEVPlayer from Pleora).

This specification provides the definitions of all camera related features and show you how to use them for your programming.

A deeper going and more detailed description of the hard- and software features of the camera can be found in the **EoSens Cube&mini Hardware Reference Manual** available from Mikrotron GmbH.

1.2 GEN<i>CAM Specification

This manual is based on the

GEN<i>CAM Standard Features Naming Convention, Version 1.2.1, 2008-08-19

from the GEN<i>CAM Organisation at www.genicam.org.

1.3 Interface Type

Each feature is accessible by a specific interface. Each interface has a number of attributes describing the current value/state and capabilities (e.g. value range) of a feature. The following interfaces are currently defined in the GenICam standard (each one is given with the typical widget used to map it on a graphical user interface):

| Interface Type | Widget |
|----------------|--|
| IIInteger | maps to a slider with value, min, max, and increment |
| IFloat | maps to a slider with value, min, and max plus a physical unit |
| IString | maps to an edit box showing a string |
| IEnumeration | maps to a drop down box |
| ICommand | maps to a command button |
| IBoolean | maps to a check box |
| IRegister | maps to an edit box showing a hex string |
| ICategory | maps to an entry in a tree structuring the camera's features |
| IPort | maps to the camera port and is typically not shown graphically |

2 Feature description

This chapter describes the camera features list in chapter 2 in detail.

The methods you may use to read from and write to the features depend on the functions provided by the API of the GigE Vision SDK of your choice. Because the GigE Vision interface is standardized there are a number of different SDK's from different vendors you can use to communicate with the camera.

The format of the values you will get from the functions of the API of your choice may be differ from this description. Please read the documentation of your SDK for details, how they handle read/write commands to the camera.

The most of the camera feature settings gets preserved if the camera gets switched off and on.

2.1. Device Information

DeviceVendorName

This feature provides the name of the manufacturer of the device.

| Feature name | DeviceVendorName |
|--------------|--|
| Access | Read only |
| Interface | IString |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | the string is limited by a trailing „\0“ |

DeviceModelName

This feature provides the model of the device.

| Feature name | DeviceModelName |
|--------------|--|
| Access | Read only |
| Interface | IString |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | the string is limited by a trailing „\0“ |

DeviceManufacturerInfo

This feature provides extended manufacturer information. Here it is the position of Mikrotron GmbH.

| Feature name | DeviceManufacturerInfo |
|--------------|--|
| Access | Read only |
| Interface | IString |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | the string is limited by a trailing „\0“ |

DeviceVersion

This feature provides the version of the device.

| Feature name | DeviceVersion |
|--------------|--|
| Access | Read only |
| Interface | IString |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | the string is limited by a trailing „\0“ |

DeviceID

This feature stores a camera identifier.

| Feature name | DeviceID |
|--------------|--|
| Access | Read only |
| Interface | IString |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | the string is limited by a trailing „\0“ |

DeviceUserID

This feature stores a user-programmable identifier.

| | |
|--------------|--|
| Feature name | DeviceUserID |
| Access | Read/Write |
| Interface | IString |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | the string is limited by a trailing „\0“ |

DeviceScanType

This feature specifies the scan type of the sensor.

| | |
|--------------|----------------|
| Feature name | DeviceScanType |
| Access | Read only |
| Interface | IEnumeration |
| Unit | -- |
| Range | Areascan |
| Invalidation | -- |
| Precondition | -- |
| Note | |

DeviceReset

When supported by the camera, this command resets the camera's circuitry.
The IP Engine reset.

| | |
|--------------|--|
| Feature name | DeviceReset |
| Access | Write only |
| Interface | ICommand |
| Unit | -- |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | After a DeviceReset, you have to do a reconnect to the camera to work with it again. |

2.1 Image Size Control

SensorWidth

This feature indicates the effective width of the sensor in pixels.

| Feature name | SensorWidth |
|--------------|--|
| Access | Read only |
| Interface | Integer |
| Unit | pixel |
| Range | EoSens Cube6 & mini1 : 1280 EoSens Cube7 & mini2 : 1696 |
| Invalidation | -- |
| Precondition | -- |
| Note | |

SensorHeight

This feature indicates the effective height of the sensor in pixels.

| Feature name | SensorHeight |
|--------------|--|
| Access | Read only |
| Interface | Integer |
| Unit | pixel |
| Range | EoSens Cube6 & mini1 : 1024 EoSens Cube7 & mini2 : 1710 |
| Invalidation | -- |
| Precondition | -- |
| Note | |

SensorDigitizationTaps

This feature represents the number of digitized samples outputted simultaneously by the camera A/D conversion stage.

| Feature name | SensorDigitizationTaps |
|--------------|------------------------|
| Access | Read/Write |
| Interface | Enum |
| Unit | -- |
| Range | Two |
| Invalidation | -- |
| Precondition | -- |
| Note | |

WidthMax

This feature represents the maximum width (in pixels) of the image after horizontal binning, decimation or any other function changing the horizontal dimensions of the image.

| | |
|--------------|--|
| Feature name | WidthMax |
| Access | Read only |
| Interface | Integer |
| Unit | pixel |
| Range | EoSens Cube6 & mini1 : 1280 EoSens Cube7 & mini2 : 1696 |
| Invalidation | |
| Precondition | -- |
| Note | |

HeightMax

This feature represents the maximum height (in pixels) of the image after vertical binning, decimation or any other function changing the vertical dimensions of the image.

| | |
|--------------|--|
| Feature name | HeightMax |
| Access | Read only |
| Interface | Integer |
| Unit | pixel |
| Range | EoSens Cube6 & mini1 : 1024 EoSens Cube7 & mini2 : 1710 |
| Invalidation | -- |
| Precondition | -- |
| Note | |

Width

This feature represents the actual image width expelled by the camera (in pixels).

| | |
|--------------|--|
| Feature name | Width |
| Access | Read/Write |
| Interface | Integer |
| Unit | pixel |
| Range | 128... WidthMax - OffsetX , Inc. 16 |
| Invalidates | AcquisitionFrameRateMax ExposureTimeMax RecordMaxFrames RecordMaxFramesPerSequence AcquisitionFrameRateRaw |
| Precondition | CameraState = Idle |
| Note | |

Height

This feature represents the actual image height expelled by the camera (in pixels).

| | |
|--------------|--|
| Feature name | Height |
| Access | Read/Write |
| Interface | Integer |
| Unit | pixel |
| Range | 2...HeightMax – OffsetY, Inc. 2 |
| Invalidation | AcquisitionFrameRateMax ExposureTimeMax RecordMaxFrames RecordMaxFramesPerSequence AcquisitionFrameRateRaw |
| Precondition | CameraState = Idle |
| Note | |

OffsetX

This feature represents the horizontal offset from the origin to the AOI – Area Of Interest (in pixels).

| | |
|--------------|--|
| Feature name | OffsetX |
| Access | Read/Write |
| Interface | Integer |
| Unit | pixel |
| Range | 0...WidthMax - Width, EoSens Cube6 & mini1 : Inc. 96 EoSens Cube7 & mini2 : Inc. 64 |
| Invalidation | AcquisitionFrameRateMax ExposureTimeMax RecordMaxFrames RecordMaxFramesPerSequence AcquisitionFrameRateRaw |
| Precondition | CameraState = Idle |
| Note | |

OffsetY

This Feature represents the vertical offset from the origin to the AOI – Area Of Interest (in pixels).

| | |
|--------------|--|
| Feature name | OffsetY |
| Access | Read/Write |
| Interface | Integer |
| Unit | pixel |
| Range | 0... HeightMax – Height , Inc. 2 |
| Invalidation | AcquisitionFrameRateMax ExposureTimeMax RecordMaxFrames RecordMaxFramesPerSequence AcquisitionFrameRateRaw |
| Precondition | CameraState = Idle |
| Note | |

PixelFormat

This feature indicates the format of the pixel to use during the acquisition.

| | |
|--------------|---|
| Feature name | PixelFormat |
| Access | Read only |
| Interface | IEnumeration |
| Unit | -- |
| Range | Mono8, BayerGR8 |
| Invalidation | -- |
| Precondition | -- |
| Note | Mono8 ← monocrome BayerGR8 ← color – Bayer Pattern Coded |

PixelCoding

This feature indicates the coding of the pixels in the image.

The camera returns frame data in raw format, that means in the native format of the sensor. The color version of the camera uses a 'Bayer Filter' in front of the sensor. The data returned by this camera type includes the color information in coded form, so the user has to convert the image to RGB by himself,

| Feature name | PixelCoding |
|--------------|------------------------------------|
| Access | Read only |
| Interface | IEnumeration |
| Unit | -- |
| Range | Mono, Raw |
| Invalidation | -- |
| Precondition | -- |
| Note | Mono <- monochrome Raw <- color |

PixelSize

This feature indicates the total size in bits of a pixel of the image.

| Feature name | PixelSize |
|--------------|--------------|
| Access | Read only |
| Interface | IEnumeration |
| Unit | -- |
| Range | Bpp8 |
| Invalidation | -- |
| Precondition | -- |
| Note | |

PixelColorFilter

This feature indicates the type of color filter that is applied to the image.

| Feature name | PixelColorFilter |
|--------------|--|
| Access | Read only |
| Interface | IEnumeration |
| Unit | -- |
| Range | None, BayerXX |
| Invalidation | -- |
| Precondition | -- |
| Note | None ← monochrome BayerXX ← Bayer Pattern Filter, XX order of the filters |

TestImageSelector

This feature selects the type of test image that is expelled by the camera.

| | |
|--------------|---|
| Feature name | TestImageSelector |
| Access | Read/Write |
| Interface | IEnumeration |
| Unit | -- |
| Range | Off, IPEngineTestPattern |
| Invalidation | -- |
| Precondition | CameraState = Idle |
| Note | Off□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Testpattern off IPEngineTestPattern ← Grayscale test pattern from the IP engine |

2.2 Acquisition and Trigger Controls

The features in this section is for to start streaming data from the camera to the host

AcquisitionMode

This feature controls the acquisition mode of the device.

It defines the number of frames to capture during an acquisition and the way the acquisition stops. Actually only the acquisition mode 'Continuous Mode' is supported.

In Continuous Mode frames are captured continuously without stopping. Acquisition can be stopped by an AcquisitionStop command.

| Feature name | AcquisitionMode |
|--------------|--------------------|
| Access | Read/Write |
| Interface | IEnumeration |
| Unit | -- |
| Range | Continuous |
| Invalidation | -- |
| Precondition | CameraState = Idle |
| Note | |

AcquisitionSource

This feature defines the source of image acquisition (live image or image from camera image buffer).

| Feature name | AcquisitionSource |
|--------------|--|
| Access | Read/Write |
| Interface | IEnumeration |
| Unit | -- |
| Range | Live, Memory |
| Invalidation | -- |
| Precondition | CameraState = Idle |
| Note | <p>Live Frame data is streamed directly from the camera to host without buffering.</p> <p>Memory Frame data is streamed from the camera internal frame buffer to the host.</p> |

AcquisitionFrameRateRaw

This feature sets the frame rate in dependency of the given frame size (in Hertz).

The EoSens Cube7 and mini2 cameras are high speed cameras with up to 523 fps and a resolution of 1696x1710 pixel. This results in a data rate from more than 1500 Mbytes/second. It is no problem to write to the camera internal memory with this high data rate, but it is far behind the data rate we can reach on a Gigabit Ethernet interface.

So the effective frame rate we can use depends on the selected data path.

If we stream live images directly from the camera to the host or if we read frames from the camera internal memory to the host, the effective frame rate is automatically reduced to a data rate that can be transferred over the Gigabit Ethernet.

But if we do recording to the camera internal memory, the frame rate is exactly the rate we set with this feature.

That's up to 523 fps at 1696*1710 (or any fps which fits the selected frame size) as described above.

| | |
|--------------|--|
| Feature name | AcquisitionFrameRateRaw |
| Access | Read only |
| Interface | Integer |
| Unit | Hz (fps – frame per second) |
| Range | 1...max, Inc. 1 |
| Invalidation | ExposureTimeMax |
| Precondition | CameraState = Idle or Live |
| Note | The max. acquisition rate depends of the current frame size. |

ExposureMode

This feature is used to set the exposure mode of the camera.

Currently there is only Timed exposure available. The exposure duration time is defined by the feature ExposureTimeAbs,

| | |
|--------------|--------------------|
| Feature name | ExposureMode |
| Access | Read/Write |
| Interface | Enumeration |
| Unit | -- |
| Range | Timed |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

ExposureTimeRaw

This feature is used to set the Exposure time in multiples of microseconds when ExposureMode is Timed.

This controls the duration where the photosensitive cells are exposed to light.

| | |
|--------------|--|
| Feature name | ExposureTimeRaw |
| Access | Read/Write |
| Interface | Integer |
| Unit | microseconds - us |
| Range | 2...ExposureTimeMax, Inc. 1 |
| Invalidation | |
| Precondition | CameraState = Idle or Live |
| Note | The max. exposure time depends of the current frame rate |

AcquisitionStart

This feature starts the Acquisition of the device.

Sending this command prepares the camera for image transmission to the host.
The source of the images depends on the setting of the feature **AcquisitonSource**.

If **AcquisitionSource** is set to **Live**, streaming goes directly from the camera head to the hosts. The transmission of the frames starts immediately after the command is given. It is possible to do live streaming to the host and recording to the internal buffer, at the same time. If live streaming and recording runs parallel, the frames get stored in the camera buffer at the full frame rate, while the number of frames send to the host gets adapted to the ability of the Gigabit Ethernet interface as described by the feature **AcquisitionFrameRateRaw**.

If **AcquisitionSource** is set to **Memory**, the requested frames comes from the camera internal buffer. The start command just prepares the camera for streaming but streaming itself is not started by the command. First set the start and end frame you want to read from the buffer and perform the command **StreamSequenceStart**. This finally will start the transmission of the requested frames from the camera to the host.

| | |
|--------------|--|
| Feature name | AcquisitionStart |
| Access | Write only |
| Interface | ICommand |
| Unit | -- |
| Range | -- |
| Invalidation | CameraState |
| Precondition | CameraState = Idle, AcquisitionSource |
| Note | <p>If AcquisitionSource is Live, then streaming comes after AcquisitionStart directly from the camera. The CameraState becomes 'Live'.</p> <p>If AcquisitionSource is Memory, then streaming comes after AcquisitionStart from the camera internal buffers. The CameraState becomes 'Streaming'.</p> |

AcquisitionStop

This feature stops the Acquisition of the device at the end of the current Frame.
If no Acquisition is in progress, the command is ignored.

| | |
|--------------|---|
| Feature name | AcquisitionStop |
| Access | Write only |
| Interface | ICommand |
| Unit | -- |
| Range | -- |
| Invalidation | CameraState |
| Precondition | CameraState = Live or Recording or Streaming |
| Note | CameraState → Idle |

2.3 Recording (Custom Features)

The EoSens cameras have an internal buffer of 2, 4 or 8 Gigabytes, which can be divided in 1, 2, 4, 8 or 16 subbuffers. This buffer is meant for storing sequences of images in the camera for later use.

Recording to the camera internal memory can be done at full camera speed, which is 625 Mbytes/second.

The camera can be run in live mode, to get live images for visualisation, while recording gets started / is running.

Recording can be stopped by the command **RecordStop** or an external or internal trigger signal. The line for an external trigger signal must be connected to the trigger in pin of the camera. A internal trigger signal can be generated by the ImageBlitz feature of the camera (see sections **ExtTrigger** and **ImageBLITZ**).

Recording to the camera memory can be done in single or continuous mode:

Single

In single mode, the camera buffer is filled with images untill the end of the buffer is reached. Afterwards, the camera stops recording. Single buffer mode just works with one buffer.

Continuous

In continuous mode, the camera buffer gets filled in a circular way. If the end of the buffer is reached, recording continuous at the start of the buffer, overriding previously grabbed images. If an internal or an external trigger signal is detected, the camera stops recording to the buffer. If more than one buffer is defined, the camera switches to next buffer and continues recording as described above. If there is no next buffer if a trigger impuls arises, recording stops finally.

RecordMode

This feature switches recording mode between continuous and single recording.

| | |
|--------------|---|
| Feature name | RecordMode |
| Access | Read/Write |
| Interface | IEnumeration |
| Unit | -- |
| Range | Single, Continuous |
| Invalidation | RecordMaxFrames RecordMaxFramesPerSequence RecordNumberOfSequences RecordTrailer |
| Precondition | CameraState = Idle |
| Note | If single recording is selected, RecordNumberOfSequences is set to 1. Recording stops automatically at the end of the record buffer. |

RecordNumberOfSequences

This feature defines the number of different recording sequences

The default value is one sequence. Possible values are 1, 2, 4, 8 or 16 sequences.

| | |
|--------------|--|
| Feature name | RecordNumberOfSequences |
| Access | Read/Write |
| Interface | IEnumeration |
| Unit | |
| Range | One, Two, Four, Eight, Sixteen |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | If RecordMode = Single , only 1 sequence can be used for recording. |

RecordTrailer

This feature defines the number of trailing frames to capture after recording stop.

This feature defines the number of frames to record after a record stop trigger is detected. The default value is 0.

| | |
|--------------|--|
| Feature name | RecordTrailer |
| Access | Read/Write |
| Interface | IInteger |
| Unit | number of picture |
| Range | 0... (RecordMaxFramesPerSequence-1) |
| Invalidation | |
| Precondition | CameraState = Idle, RecordMode = Continuous |
| Note | only for the continuous recording. If RecordMode = Single , the feature will be locked and the value will be invalid. |

RecordMaxFrames

This feature returns the max. possible frames of all sequences.

This is the number of frames that can be recorded for all sequences in sum.

| | |
|--------------|---|
| Feature name | RecordMaxFrames |
| Access | Read only |
| Interface | IInteger |
| Unit | number of picture |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | The maximal number of frames depends on the frame size. |

RecordNumberOfFrames

This feature returns the number of recorded frames of all sequences.

If recording is finished, this feature is set to the sum of all recorded images of all sequences.

| | |
|--------------|---------------------|
| Feature name | RecordNumerOfFrames |
| Access | Read only |
| Interface | Integer |
| Unit | number of picture |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | |

RecordMaxFramesPerSequence

This feature returns the number of max. possible frames per sequence.

This is the maximal number of frames, that can be recorded in one sequence.

| | |
|--------------|---|
| Feature name | RecordMaxFramesPerSequence |
| Access | Read only |
| Interface | Integer |
| Unit | number of picture |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | The maximal number of frames per sequence depends on the frame size and the quantity of sequence. |

RecordNumberOfFramesPerSequence

This feature returns the number of recorded frames per sequence.

The sequence for which the number of recorded frames shall be returned must be selected by the feature '**StreamSequenceSelector**' before reading (see feature category 'Streaming').

| | |
|--------------|--|
| Feature name | RecordNumberOfFramesPerSequence |
| Access | Read only |
| Interface | Integer |
| Unit | number of picture |
| Range | -- |
| Invalidation | StreamEndNumber |
| Precondition | -- |
| Note | Selector StreamSequenceSelector must be set before for reading the number of recorded frames in the sequence. |

RecordStart

This command starts the recording to the camera image buffer.

Recording will start at the first sequence buffer of the defined number of sequences.

| | |
|--------------|------------------------------------|
| Feature name | RecordStart |
| Access | Write only |
| Interface | ICommand |
| Unit | -- |
| Range | -- |
| Invalidation | CameraState |
| Precondition | CameraState = Idle or Live, |
| Note | |

RecordStop

This command stops the recording to the camera image buffer.

| | |
|--------------|--|
| Feature name | RecordStop |
| Access | Write only |
| Interface | ICommand |
| Unit | -- |
| Range | -- |
| Invalidation | CameraState |
| Precondition | CameraState = Recording |
| Note | This command stops recording even if not all defined buffers are filled. |

RecordWhileTriggerActive

If this feature is true the camera records frames as long as the external trigger or the image trigger is active. Only available in single recording mode.

| | |
|--------------|---|
| Feature name | RecordWhileTriggerActive |
| Access | Read/Write |
| Interface | IBoolean |
| Unit | -- |
| Range | True, False |
| Invalidation | RecordBurstCnt |
| Precondition | CameraState = Idle, RecordMode = Single |
| Note | only for the single recording. If RecordMode = Continuous , the feature will be locked and the value will be invalid. |

RecordStandby

If this feature is True, the camera enters standby mode after next/current recording ends.

In stand by mode the camera is set to a state where it consumes as little current as possible. Just the part of the camera which is responsibly for keeping the camera's memory stable is hold active. This mode is used to keep records in the camera's memory as long as possible for later reading.

| | |
|--------------|---|
| Feature name | RecordStandby |
| Access | Read/Write |
| Interface | IBoolean |
| Unit | -- |
| Range | True, False |
| Invalidation | -- |
| Precondition | CameraState = Idle |
| Note | It is not possible to read from the camera memory as long as the camera is in standby mode. To get the camera out of the standby mode you have to press the camera's power switch for a short moment (less than 1 second). Please pay attention: If you press the switch longer than 1 second, you will shut down the camera completely and the camera's recording will be lost! |

RecordBurstCnt

This feature defines the number of consecutive frames to record after an external trigger or an image trigger occurs. Only available in single recording mode.

| | |
|--------------|---|
| Feature name | RecordBurstCnt |
| Access | Read/Write |
| Interface | IInteger |
| Unit | -- |
| Range | 0...1022, 0 means to record until the buffer is filled |
| Invalidation | -- |
| Precondition | CameraState = Idle RecordMode = Single RecordWhileTriggerActive = False |
| Note | Each external trigger causes the camera to record RecordBurstCount frames. The frames gets attached to previously sampled frames already stored in the camera buffer. New frames/frame sequences can be added, until the buffer is full. only for the single recording. If RecordMode = Continuous , the feature will be locked and the value will be invalid. |

RecordCameraBufferSize

This feature returns the size of the camera internal image buffer in Giga Bytes.

| | |
|--------------|------------------|
| Feature name | RecordBurstCount |
| Access | Read |
| Interface | Integer |
| Unit | Giga Bytes |
| Range | -- |
| Invalidation | -- |
| Precondition | -- |
| Note | |

RecordCurrentSequenceNumber

If recording is running, this feature returns the number of the currently active sequence.

| | |
|--------------|-----------------------------|
| Feature name | RecordCurrentSequenceNumber |
| Access | Read |
| Interface | Integer |
| Unit | -- |
| Range | 0...16 |
| Invalidation | -- |
| Precondition | -- |
| Note | |

2.4 Streaming (Custom Features)

The streaming features are used to transfer previously recorded frames from the camera internal memory to the host computer.

The camera can be configured to use 1, 2, 4, 8 or 16 different buffers (also called **Sequences**) for recording. The camera memory is divided by the number of sequences to record. If just one buffer is defined, the complete camera memory is reserved for this one buffer. If two sequences are defined, the memory is divided by 2. So each buffer gets assigned the half of the camera memory, and so on.

One buffer is always the current buffer, that gets circularly filled with images if the camera is in continuous recording mode. An external or an internal trigger puls stops recording to the current buffer and switches to the next buffer to continue with recording. This buffer gets the new current buffer. If the last buffers gets a stop signal by the trigger, recording stops finally and the camera enters the idle state.

Now the images in the different buffers can be read out for further processing.

Before reading frames from a buffer, it must be selected by its sequence number (0...16). All streaming commands and setting will refer to this selected buffer from now on.

The frames can be read from a buffer as single frames or as a sequence of frames.

Single frames can be read out randomly by setting its frame number, starting with 0. The transmission of the frames starts at the moment the frame number is written to the according feature.

Before starting the transfer from the camera to the host, the host and the camera must be set to a state where they accept respectively transfer image data. This is done on the camera side by sending the **AcquisitionSource**(Memory) command and the **AcquisitionStart** command to the camera.

Reading a sequence of frames requires two steps. At first the start and the end number of the frames must be defined by setting the start and the end number feature. Now, streaming of the frames can be started by executin the '**StreamSequenceStart**' command.

The frame with the number 0 of a sequence is always the oldest frame in the buffer, even if recording is done in continuous mode (this is organized automatically by the camera). If there was an trailer defined for the current recording (see feature **RecordTrailer**), the frame at which the trigger arises ('trigger frame'), is the last frame in the sequence minus the number of trailer frames.

Please note: Starting a new recording will overwrite all previously recorded frames !

StreamSequenceSelector

This feature selects the current sequence for streaming.

If streaming from the camera to the host is selected, this feature defines the index of the sequence the frames are read from.

| | |
|--------------|---|
| Feature name | StreamSequenceSelector |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 0 .. 15, Increment 1 |
| Invalidation | RecordNumberOfFramesPerSequence StreamStartNumber StreamEndNumber |
| Precondition | CameraState = Idle or Live |
| Note | <p>This features are indexed by the value of StreamSequenceSelector:</p> <p>RecordNumberOfFramesPerSequence StreamStartNumber StreamEndNumber StreamSequenceStart StreamSingleFrame</p> <p>For example, to get the number of recorded frames in a sequence, the StreamSequenceSelector must be set to its index. After setting the sequence number, the number of frames can be read from feature RecordNumberOfFramesPerSequence.</p> |

StreamStartNumber

This feature defines the number of the start frame for reading frames from an image buffer.

The index of the referenced streaming buffer must be selected by the feature **StreamSequenceSelector** before setting this parameter. The frame number is relative to each buffer and starts with 0.

| | |
|--------------|------------------------------------|
| Feature name | StreamStartNumber |
| Access | Read/Write |
| Interface | Integer |
| Unit | number of picture |
| Range | 0... StreamEndNumber |
| Invalidation | |
| Precondition | CameraState = Idle or Live, |
| Note | |

StreamEndNumber

This feature defines the number of the end frame for reading frames from an image buffer.

The index of the referenced streaming buffer must be selected by the feature **StreamSequenceSelector** before setting this parameter. The frame number is relative to each buffer.

| | |
|--------------|--|
| Feature name | StreamEndNumber |
| Access | Read/Write |
| Interface | Integer |
| Unit | number of picture |
| Range | StreamStartNumber... RecordNumberOfFramesPerSequence-1, Inkrement1 |
| Invalidation | |
| Precondition | CameraState = Idle or Live |
| Note | |

StreamSequenceStart

This feature starts streaming of a sequence of images.

To prepare for streaming the feature **AcquisitionSource** must be set to **Memory**. The sequence from which we want to read the images must be selected by the feature **StreamSequenceSelector**. **StreamStartNumber** and **StreamEndNumber** must be set to define the range of images to stream.

AcquisitionStart must be set before a call to the feature **StreamSequenceStart** starts streaming.

| | |
|--------------|---|
| Feature name | StreamSequenceStart |
| Access | Write only |
| Interface | ICommand |
| Unit | -- |
| Range | -- |
| Invalidation | CameraState |
| Precondition | CameraState = Idle and AcquisitionSource = Memory StreamStartNumber StreamEndNumber AcquisitionStart |
| Note | |

StreamSingleFrame

This feature reads a single frame from an image buffer.

The frame to read is absolutely addressed by its index in the sequence buffer.

Before the image number is set, the feature **AcquisitionSource** must be set to **Memory**. The sequence the frame has to be read from must be selected by the feature **StreamSequenceSelector**. To stream the images to the host, **AcquisitionStart** must be active. Streaming the frame to the host starts as soon as the image number is set by this feature.

| | |
|--------------|---|
| Feature name | StreamSingleFrame |
| Access | Read/Write |
| Interface | Integer |
| Unit | number of picture |
| Range | 0... RecordNumberOfFramesPerSequence-1 |
| Invalidation | |
| Precondition | CameraState = Idle and AcquisitionSource = Memory AcquisitionStart |
| Note | |

2.5 Analog Controls

GainRaw

This feature sets the digital gain of the camera.

| | |
|--------------|--|
| Feature name | GainRaw |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | EoSens Cube6 & mini1 : 100,150,200,250,300,350,400 EoSens Cube7 & mini2 : 100,150,200 |
| Invalidation | |
| Precondition | |
| Note | The value correspond with x1, x1.5, x2, x2.5, x3, x3.5, x4 |

BlackLevelRaw

This feature sets the black level of the camera.

| | |
|--------------|---------------|
| Feature name | BlackLevelRaw |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 50...200 |
| Invalidation | |
| Precondition | |
| Note | |

2.6 Multiple Slope (Custom Features)

The Multi Slope feature of the camera can be used to increase the dynamic range of images taken by the camera.

At a user defined time within the exposure time, all pixel that exceeds a predefined value of luminosity gets its luminosity reset to this value. Pixels below this value gets untouched. The rest of the exposure time all pixels gets exposed as usual.

The effect of this algorithm is, that normally over exposed pixels gets reduced in its luminosity while darker pixels keeps their value. The results is an exposure with an increased dynamic range.

The EoSens Cube6 & mini1 cameras can set one or two different points within one frame time to reset over exposed pixels to a predefined value. The EoSens Cube7 & mini2 can set only one point.

MultipleSlopeMode

This feature defines the Multiple Slope Mode.

The multiple slope mode is used to extend the dynamic range of the camera.

| | |
|--------------|--|
| Feature name | MultipleSlopeMode |
| Access | Read/Write |
| Interface | IEnumeration |
| Unit | |
| Range | NormalShutter, DualSlope, TripleSlope |
| Invalidation | MultipleSlopeDualRaw MultipleSlopeTripleRaw |
| Precondition | CameraState = Idle or Live |
| Note | NormalShutter – disable slops DualSlope – dual slope enable TripleSlope – triple slope enable (EoSens Cube6 & mini1) |

MultipleSlopeDualRaw

This feature sets the value for Dual Multiple Slope in percent of the exposure time.

| | |
|--------------|--|
| Feature name | MultipleSlopeDualRaw |
| Access | Read/Write |
| Interface | IInteger |
| Unit | |
| Range | 1...MultipleSlopsTripleRaw-1(TripleSlope) or 1...99(DualSlope) |
| Invalidation | |
| Precondition | CameraState = Idle or Live and MultipleSlopsMode = DualSlope or TripleSlope |
| Note | |

MultipleSlopeTripleRaw

This feature sets the value for Triple Multiple Slope in percent of the exposure time.

| | |
|--------------|--|
| Feature name | MultipleSlopeTripleRaw |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | MultipleSlopsDualRaw+1...99 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and MultipleSlopsMode = TripleSlope |
| Note | only for EoSens Cube6 & mini1 |

2.7 ExtTrigger (Custom Features)

If the camera is in circular recording mode, capturing frames can be stopped by a signal on the external trigger input pin of the camera. The external stop trigger feature must be enabled in the camera before it can be used.

Alternatively the trigger button on the back side of the camera can be used to send trigger signals to the camera (EoSens Cube6&7 only).

Instead of an external trigger signal, the ImageBlitz feature of the camera can be used to generate trigger pulses internally.

If more than one capture buffers (sequences) are defined for recording, the camera stops recording to the current buffer and starts recording to the next buffer on each trigger puls. If the camera is writing to the last buffer (or the first, if just one buffer is defined), the next trigger puls will stop recording to the buffer and set the camera to idle state.

If the camera has entered the idle state, recorded frames can be read out of the camera for further processing.

ExtTriggerEnable

This Feature enables/disables the input port for an external trigger signal.

| | |
|--------------|--------------------|
| Feature name | ExtTriggerEnable |
| Access | Read/Write |
| Interface | Boolean |
| Unit | |
| Range | True, False |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

ExtTriggerPolarity

This feature defines if the camera trigger gets active on the rising or falling edge of the external trigger signal.

| | |
|--------------|--------------------|
| Feature name | ExtTriggerPolarity |
| Access | Read/Write |
| Interface | IEnum |
| Unit | |
| Range | Raising, Falling |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

2.8 ImageBlitz (Custom Features)

The ImageBlitz is a camera internal trigger unit which generates trigger pulses in dependency of changes in a predefined area (ROI) of the camera sensor. At a user defined time the current content of the area gets saved and is subsequently compared against the same area in each new captured image for trigger generation. The trigger puls is used to stop cyclic recording in the same manner as described by the external trigger feature.

The size and position of the ROI can be freely defined within the sensor area but the total number of pixels within the trigger area may not exceed 20480 pixel. The trigger condition within the area is defined by a threshold for the amount of changes in the brightness of a pixel and the number of pixels (Release Condition) that have to change to release the trigger.

The pixels in the ROI are subpacket to sets of 10 pixels. If one or more pixel in a set exceeds the threshold, a counter is incremented by one. If the counter exceeds the release condition, a trigger signal is generated. The counter gets set to 0 for each new frame captured by the camera.

To ease the positioning of the ImageBlitz area and the definition of the threshold and release condition, the upper and lower border of the ROI can be displayed as dotted lines in the captured images if the camera is in live mode. To visualize the trigger point, the dotted lines gets inverted each time the trigger releases.

IBEnable

This feature enables/disables the Mikrotron ImageBlitz feature.

| | |
|--------------|---|
| Feature name | IBEnable |
| Access | Read/Write |
| Interface | IBoolean |
| Unit | |
| Range | True, False |
| Invalidation | IBWindowVisible |
| Precondition | CameraState = Idle or Live |
| Note | A transition from True->False saves the current content of the trigger ROI for the reference image. |

IBHeight

This feature defines the height of the ImageBlitz window in units of lines.

| | |
|--------------|--|
| Feature name | IBHeight |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 1...SensorHeight-IBOffset, Inc. 1 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IBEnable = True IBHeight*IBWidth <=20480 |
| Note | |

IBOffsetX

This feature defines the right offset of the ImageBlitz window starting from the first sensor row, in units of pixels.

| | |
|--------------|--|
| Feature name | IBOffsetX |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 0...SensorWidth-IBWidth, Inc. 16 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IBEnable = True |
| Note | |

IBOffsetY

This feature defines the offset of the ImageBlitz window starting from the first sensor line, in units of lines.

| | |
|--------------|--|
| Feature name | IBOffsetY |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 0...SensorHeight-IBHeight, Inc. 1 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IBEnable = True |
| Note | |

IBReleaseCondition

This feature defines the number of pixel to fulfill the ImageBlitz trigger condition.

| | |
|--------------|--|
| Feature name | IBReleaseCondition |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 1...100, Inc. 1 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IEnable = True |
| Note | |

IBThreshold

This feature defines the trigger threshold of the ImageBlitz feature.

The threshold is the steps of brightness a pixel must change to be counted to the release condition.

| | |
|--------------|--|
| Feature name | IBThreshold |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 1...255, Inc. 1 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IEnable = True |
| Note | |

IBWidth

This feature defines the width of the ImageBlitz window in units of pixels.

| | |
|--------------|--|
| Feature name | MultipleSlopsTripleRaw |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | 16...SenosrWidth-IBOffsetX, Inc. 16 |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IEnable = True IBHeight*IBWidth <=20480 |
| Note | |

IBWindowVisible

This feature shows/hides the ImageBlitz trigger window.

The upper and lower border of the trigger window is displayed as dotted lines in each captured image.

| | |
|--------------|--|
| Feature name | IBWindowVisible |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | |
| Invalidation | |
| Precondition | CameraState = Idle or Live and IBEnable = True |
| Note | Each time the trigger condition is fulfilled, the dotted lines of the ImageBlitz area gets inverted. |

2.9 Synchronisation (Custom Features)

The cameras have an output pin, which can be connected to a camera generated SYNC OUT signal or a ARM signal.

If SYNC OUT is selected, this output will carry a strobe that corresponds to the selected exposure time of the camera.

If ARM is selected, it will be active if the camera runs in circular recording mode.

The cameras can be synchronized by an internal or an external signal. The external synchronisation is used to run more than one cameras synchronized to each other by connecting them to the same external sync signal. The frequency of the external sync signal must be below the free-running frame rate of the cameras.

The sync in is also used to synchronize a slave camera to the strobe out signal of a master camera. Again, take attention that the frequency of the masters strobe signal is a little bit less than the free-running frequency of the slave camera.

SyncARMPolarity

This feature defines the polarity of the ARM signal.

| | |
|--------------|--------------------|
| Feature name | SyncARMPolarity |
| Access | Read/Write |
| Interface | IEnum |
| Unit | |
| Range | positive, negative |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

SyncInEnable

This feature enables the synchronisation input pin of the camera connector for ext. camera synchronisation.

| | |
|--------------|--------------------|
| Feature name | SyncInEnable |
| Access | Read/Write |
| Interface | IBoolean |
| Unit | |
| Range | True, False |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

SyncInPolarity

This feature defines if the camera is synchronised to the falling or raising edge of the sync in signal

| | |
|--------------|--------------------|
| Feature name | SyncInPolarity |
| Access | Read/Write |
| Interface | IEnum |
| Unit | |
| Range | raising, falling |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

SyncOutSelect

This feature defines the camera signal to connect to the Sync Out / ARM pin of the camera connector.

| | |
|--------------|---|
| Feature name | SyncOutSelect |
| Access | Read/Write |
| Interface | IEnum |
| Unit | |
| Range | ARM, SyncOut |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | <p>ARM = Signals that the camera is in circular recording mode.</p> <p>SyncOut = If set, the camera outputs a strobe signal which is equivalent to the exposure time of the camera.</p> |

2.10 Custom Features

CameraState

This feature returns the current transmission state of the camera.

The camera always is in one of the following states:

| | |
|--------------|---|
| Feature name | CameraState |
| Access | Read only |
| Interface | IEnumeration |
| Unit | |
| Range | Idle, Live, Streaming, Recording |
| Invalidation | <p>Idle→Live: -- Idle→Streaming:-- Idle→Recording: RecordNumberOfFrames, RecordNumberOfFramesPerSequences</p> <p>Live→Idle: -- Live→Recording: RecordNumberOfFrames, RecordNumberOfFramesPerSequences</p> <p>Recording→Idle: RecordNumberOfFrames, RecordNumberOfFramesPerSequences</p> <p>Streaming→Idle: --</p> |
| Precondition | |
| Note | |

IRIGBEnable

This feature enables/disables the processing of an IRIG-B signal connected to the according camera input pin (this feature is not available on all cameras).

| | |
|--------------|------------------------|
| Feature name | MultipleSlopsTripleRaw |
| Access | Read/Write |
| Interface | IBoolean |
| Unit | |
| Range | True, False |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | |

DecimationXY

This feature enables the sensor to skip every second row and column.

| | |
|--------------|--|
| Feature name | DecimationXY |
| Access | Read/Write |
| Interface | IBoolean |
| Unit | |
| Range | True, False |
| Invalidation | |
| Precondition | CameraState = Idle |
| Note | Width and Height must be halves or doubles after DecimationXY . Wait One Second after DecimationXY, Width and Height |

Temperature

This feature reads the temperature of the camera body as raw data.

| | |
|--------------|--|
| Feature name | Temperature |
| Access | Read |
| Interface | Integer |
| Unit | |
| Range | |
| Invalidation | |
| Precondition | |
| Note | EoSens Cube6, EoSens mini1, EoSens mini2 |

2.11 CustomData (Custom Features)

Data00, Data01, Data02, Data03

Data00 to Data03 can be used to store custom data in the non volatile memory of the camera. This preserves the data for later use, even if the camera power gets switched off.

| | |
|--------------|--------------------------------|
| Feature name | Data00, Data01, Data02, Data03 |
| Access | Read/Write |
| Interface | Integer |
| Unit | |
| Range | |
| Invalidation | |
| Precondition | |
| Note | |

2.12 UserSets

The User Set feature is used, to preserve current frame settings, even if the camera gets switched off. The frame values that gets stored are frame **Width**, **Height**, **OffsetX** and **OffsetY**. The saved frame settings can be restored later by the user or if the camera gets reset by a power off/on cycle.

UserSetSelector

This feature must be set to '**UserSet1**' before saving the current frame size by the feature **UserSetSave**. The User Set **Default** is reserved for internal use. It can not be used to save user settings!

| | |
|--------------|---|
| Feature name | UserSetSelector |
| Access | Read/Writer |
| Interface | IEnumeration |
| Unit | |
| Range | Default, UserSet1 |
| Invalidation | |
| Precondition | |
| Note | No data can be saved if User Set Default is selected. By default, the frame size is set to its maximum. |

UserSetLoad

This feature loads the User Set specified by **UserSetSelector** to the device and makes it active. The User Set includes the features **Width, Height, OffsetX, OffsetY**.

| | |
|--------------|-------------|
| Feature name | UserSetLoad |
| Access | Writer |
| Interface | ICommand |
| Unit | |
| Range | |
| Invalidation | |
| Precondition | |
| Note | |

UserSetSave

This feature saves the current frame settings to the User Set specified by **UserSetSelector**. The data gets saved in the non-volatile memory of the camera. The User Set includes the features **Width, Height, OffsetX, OffsetY**.

| | |
|--------------|--|
| Feature name | UserSetSave |
| Access | Writer |
| Interface | ICommand |
| Unit | |
| Range | |
| Invalidation | |
| Precondition | |
| Note | The Default feature can be not saved. By the first time of the saving of UserSet1, make sure that the UserSetDefaultSelector is Default. After the saving, the UserSetDefaultSelector can be set to UserSet1. |

UserSetDefaultSelector

This feature defines the User Set to Load when the camera gets reset. The User Set includes the features **Width, Height, OffsetX, OffsetY**.

| | |
|--------------|------------------------|
| Feature name | UserSetDefaultSelector |
| Access | Read/Writer |
| Interface | IEnumeration |
| Unit | |
| Range | Default, UserSet1 |
| Invalidation | |
| Precondition | |
| Note | |

3 General Programming Guide

3.1 Camera States

The feature **CameraState** indicates the current state of the camera.

The camera has 4 states:

IDLE

the camera is in IDLE state

LIVE

the camera is streaming live frames from the camera to the host

RECORDING

the camera is in recording mode. Recorded frames will be stored in the camera's internal buffer. If LIVE streaming was enabled when recording starts, a live image stream is sent to the host while recording is running.

STREAMING

the camera streams recorded frames from the camera's internal buffer to the host.

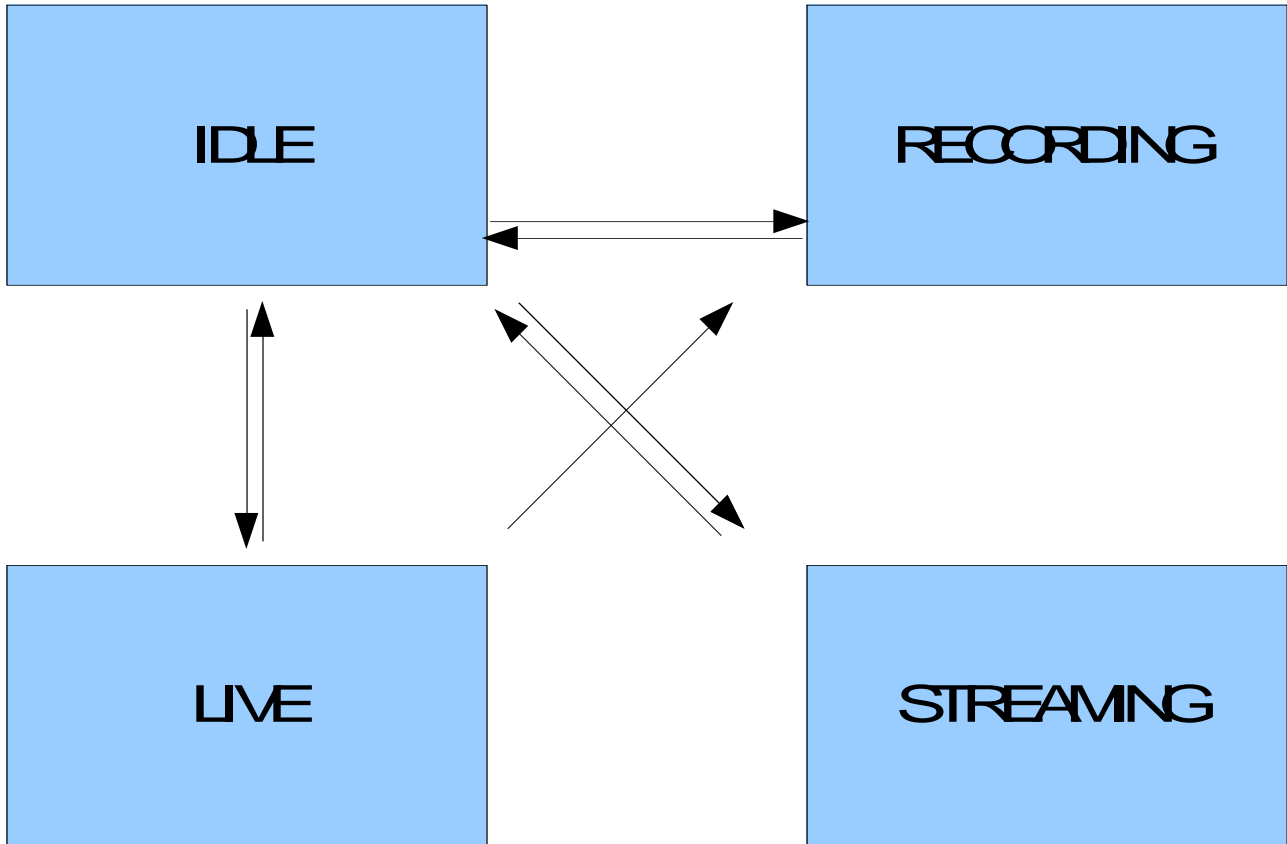
The accessibility of camera features depends on the current camera state. The table below shows these dependencies.

| | available | unavailable |
|------------------|--|-------------|
| IDLE | all states | - |
| LIVE | all „read only“ features ExposureTimeRaw GainRaw BlackLevelRaw MultipleStopsMode MultipleStopsDualRaw MultipleStopsTripleRaw AcquisitionStop RecordStart | others |
| RECORDING | all „read only“ features AcquisitionStop RecordStop | others |
| STREAMING | all „read only“ features AcquisitionStop | others |

3.2 State Transformations

The camera goes to the IDLE state automatically after the power-up.

The diagram below shows the state transitions between the different camera states:



Possible state transitions:

| Transition | Precondition | Transition trigger |
|------------------|----------------------------|---|
| IDLE → LIVE | AcquisitionSource = Live | AcquisitionStart |
| LIVE → IDLE | -- | AcquisitionStop |
| IDLE → RECORDING | -- | RecordStart or trigger button on camera (Cube only) |
| RECORDING → IDLE | -- | RecorStop or trigger on last sequence. |
| LIVE → RECORDING | -- | RecordStart or trigger butto on camera (Cube only) |
| IDLE → STREAMING | AcquisitionSource = Memory | AcquisitionStart and StreamSequencesStart or AcquisitionStart and StreamSingleFrame |
| STREAMING → IDLE | -- | AcquisitionStop or transmittion is finished |

4 Record Programming Guide

The camera can save recorded frames directly into the internal memory of the camera at full speed. The memory can be configured as one large buffer or can be divided in up to 16 regions of equal size, called sequences. Recording is controlled by the features **RecordStart**, **RecordStop** and trigger signals.

Trigger signals may come from : - the trigger line of the camera
 - the button on the back side of the camera (EoSens Cube only)
 - ImageBLITZ@

The table below gives you an overview of the possible recording modes of the camera. The method to start and stop recording depends on the used recording mode.

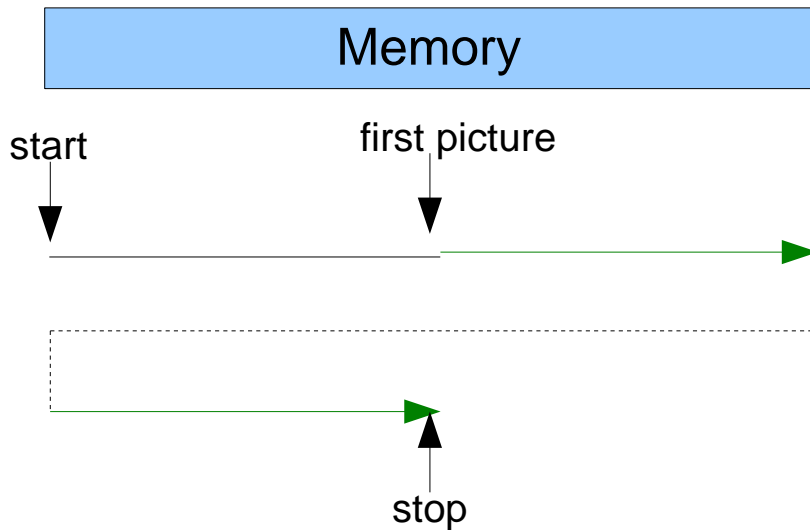
| Record Mode | Start | Stop |
|---|--------------------------------------|---|
| Continuous Recording one sequence | RecordStart | RecordStop or trigger signal. |
| Continuous Recording with more than one sequences | RecordStart | RecordStop or trigger signal. („Sequence jump“ only with triggers) |
| Single Recording | RecordStart | RecordStop |
| Triggered Recording (only one sequence) | trigger button on camera (Cube only) | trigger button on camera (Cube only) |

The different recording mode will be discussed in detail in the chapters below.

Note : Most features about record will be stored automatically in non volatile memory of camera after every recording. (RecordWhileTriggerActive, RecordBurstCnt, RecordStandby, RecordMode, RecordTrailer)(see 2.12 and 4.4).

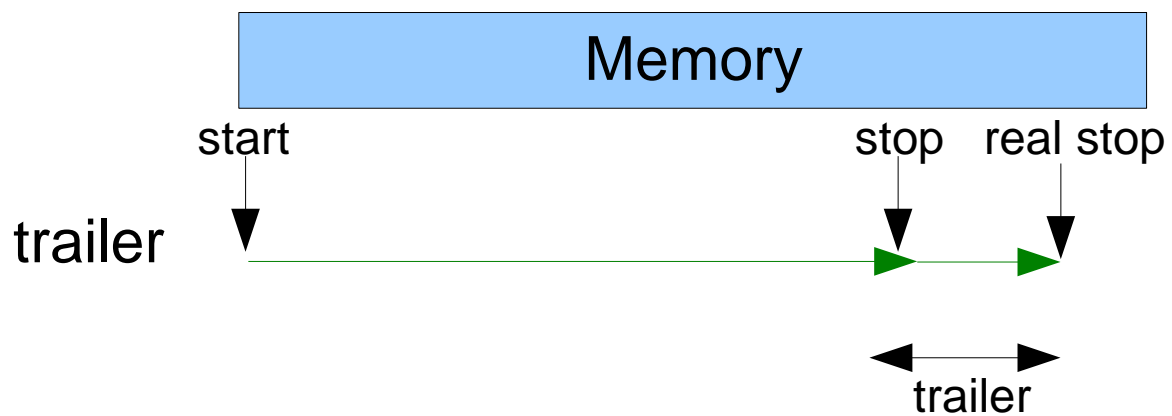
Continuous Recording, one sequence

In continuous mode the camera fills the internal buffer circularly with frames. It starts recording at the beginning of the buffer and fills it with frames until the end of the buffer is reached. After reaching the end of the buffer recording starts again at the beginning of the buffer and so on. Recording stops when the command „**RecordStop**“ is given or a trigger signal is detected.



The **RecordMode**, **RecordNumberOfSequences** and **RecordTrailer** features must be set before recording is started.

There is an additional feature for defining a record sequence, called '**RecordTrailer**'.



The value of the feature '**RecordTrailer**' defines the number of additional frames to record after a stop signal is detected. The trailer value is only meaningful in continuous recording mode with one or more sequences.

Example:

We want to record 100 pictures after a trigger event. So the feature **RecordTrailer** has to be set to 100 before recording gets started. The default value of the **RecordTrailer** is 0.

Recording is started with the feature **RecordStart**. As long as recording is running the Trigger-LED on the camera is blinking and the feature **CameraState** returns the state „Recording“.

Recording gets stopped with the feature **RecordStop** or a trigger signal. After the record stop event is given recording continues until the additional 100 frames (defined by **RecordTrailer**) gets captured and stored in the buffer. Now the camera state changes to „Idle“ and the trigger-LED will stop blinking.

Please note: If recording is stopped by an external trigger event (initiated by a signal on the ext. trigger line of the camera or by the trigger button of the camera), you have to give the camera an additional **RecordStop** command.

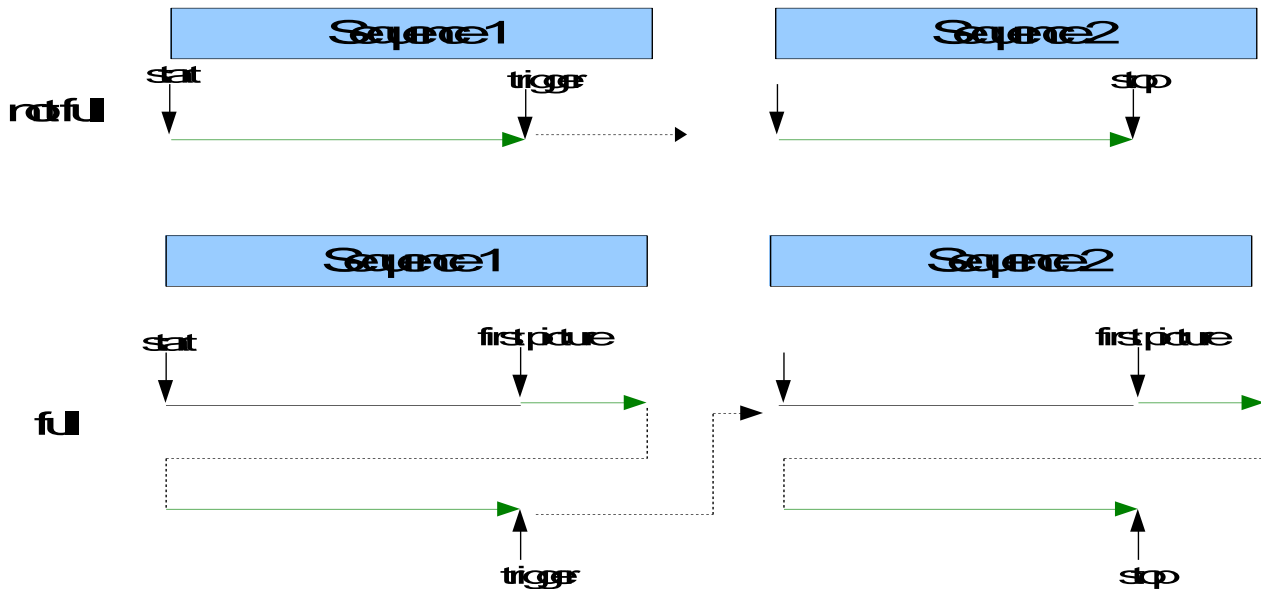
The programming flow for a continuous recording to one sequence is:

```
RecordMode = Continuous
RecordNumberOfSequences = One
RecordTrailer = x (Range: 0.....RecordMaxFramesPerSequence-1)
RecordStart
...
RecordStop or a trigger signal
(if RecordTrailer is unequal to 0, wait until the CameraState = Idle)
(if it stops with a trigger signal, give the RecordStop)
AcquisitionStop
Read the RecordNumberOfFrames and RecordNumberOfFramesPerSequence
```

4.1 Continuous Recording with more than one sequence

The camera internal buffer can be subdivided into 2, 4, 8, 16 record sequences. Each recording to a sequence buffer is done as described in the mode 'Continuous Recording, single buffer' above.

The difference to a single buffer recording is described by an example with 2 sequences in the example below:



The start of recording must be done per software (**RecordStart**), and the end of the recording can be done per software or a trigger signal. But if a trigger signal is used, you must give the camera a **RecordStop** after the **CameraState** = Idle. The jump from one sequence to the next can only be done by the trigger signals. But only one method can be used in once recording.

The programming flow of a continuous recording with more sequences is:

RecordMode = Continuous

RecordNumberOfSequences = Two, Four, Eight or Sixteen sequences

RecordTrailer = x (Range: 0.....RecordMaxFramesPerSequence-1)

RecordStart

...(jump with triggers)

RecordStop or a trigger signal

(if **RecordTrailer** is unequal to 0, wait until the **CameraState** = Idle)

(if it stops with a trigger signal, give the **RecordStop**)

AcquisitionStop.

Then the number of recorded frames can be read. You can read the sum of all recorded frames of all sequences (**RecordNumberOfFrames**) or the number of recorded frames in a specific sequence (**RecordNumberOfFramesPerSequence**). To get the number of recorded frames in a specific sequence, you have to select the sequence with the feature **StreamSequenceSelector** before.

4.2 Single Recording

Single recording means, that the recording starts at the beginning of the memory and stops automatically at the end of the buffer. Recording can be stopped prematurely using the command feature „**RecordStop**“.

If the camera is stopped automatically by reaching the end of the buffer, an additional **RecordStop** must be send to the camera.

Single recording mode is only available for one sequence. So, if the **RecordMode** is set to single

the **RecordNumberOfSequences** will be set to one automatically.

The single recording mode has three submodes:

No Trigger mode : after **RecordStart** the recording starts at the beginning of the memory and stops automatically at the end of the buffer.

While Trigger Active mode : after **RecordStart** the camera waits for a trigger. Images gets recorded as long as the the trigger stays active. The recording stops automatically if the buffer is full.

Burst mode : Before recording starts, the feature **RecordBurstCnt** must be set to the number of frames to record (comparable to the feature **RecordTrailer**). After RecordStart is initiated, the camera waits for the a trigger signal. The recording starts if a trigger is detected and records the number of **RecordBurstCnt** frames. Then the camera waits for the next trigger..... The recording stops automatically if the buffer is full.

The programming flow for the No Trigger mode is:

RecordMode = Single

RecordNumberOfSequences = One and unavailable (automatically)

RecordWhileTriggerActive = False

RecordBurstCnt = 0

RecordStart

...

recording stops automatically (**CameraState** = IDLE), then **RecordStop** must be given

AcquisitionStop

Read the number of recorded frames form the feature **RecordNumberOfFrames** which could be equal to **RecordNumberOfFramesPerSequence**.

The programming flow for the While Trigger Active mode is:

RecordMode = Single

RecordNumberOfSequences = One and unavailable (automatically)

RecordWhileTriggerActive = True

RecordBurstCnt = 0

RecordStart

...reording so long as trigger active

wait

...reording so long as trigger active

...

recording stop automatically (**CameraState** = IDLE), then **RecordStop**

Read the number of recorded frames form the feature **RecordNumberOfFrames** which could be equal to **RecordNumberOfFramesPerSequence**.

The programming flow for the Burst mode is:

RecordMode = Single

RecordNumberOfSequences = One and unavailable (automatically)

RecordWhileTriggerActive = False

RecordBurstCnt = x(unequal to 0)

RecordStart

...reording after one trigger x frames

...wait

...reording after one trigger x frames

...

recording stop automatically (**CameraState** = IDLE), then **RecordStop**

AcquisitionStop

Read the number of recorded frames form the feature **RecordNumberOfFrames** which could be equal to **RecordNumberOfFramesPerSequence**.

4.3 Triggered Recording

Triggered recording means, that recording is started with the trigger button of the camera(only for EoSens Cube6 and EoSens Cube7). Recording is done in continuous recording mode. Stopping recording is also done with the trigger button of the camera. There is no live picture available for triggered recording.

The programming flow of a triggered recording is:

RecordMode = Continuous

Camera must be in 'idle' state

Give the trigger button(The recording starts)

...

Give the trigger button(The recording stops, if **RecordTrailer** = 0)

...(if RecordTrailer = x)

Wait until **CameraState** gets 'Idle'. Now send a **RecordStop** to the camera(if there is connection with PC) or Wait until the Trigger-LED do not blink (if there is no connection with PC).

Read the **RecordNumberOfFrames** and **RecordNumberOfFramesPerSequence** to get the number of recorded frames.

4.4 RecordStandby (UserSets)

This mode is used to keep the images of a previously recorded sequence in the camera's memory for later use.

The feature **RecordStandby** must be set to **True** before a new recording gets started. The most parameters describing the recorded sequence will be stored in non volatile memory, with exception of the values for frame **Width**, **Height**, **OffsetX** and **OffsetY**. This values must be saved separately by setting the **UserSets** feature to **UserSet1** and saving the frame size in the camera by sending the command **UserSetSave**.

It is **very important** to save the frame size of the images in the camera buffer to **UserSet1**, because a number of record parameters gets calculated in dependency of this values (e.g. **RecordNumberOfFrames**, **RecordMaxFramesPerSequence**, **RecordMaxFrames**, **RecordNumberOfFramesPerSequence**). So, if you forget to save the frame values to the User Set, the reading of an previously stored sequence may fail after the camera leaves the standby mode.

It's a good practice to save the current frame size of the camera to **UserSet1** as soon as one of the parameters **Width**, **Height**, **OffsetX** or **OffsetY** gets changed if a recording shall be used after **RecordStandby**.

5 Streaming Programming Guide

After recording is stopped the internal buffer of the camera is filled with a number of recorded frames. To enable streaming from the camera buffer to the host, the feature **AcquisitionSource** must be set to '**Memory**'.

The maximum of streamrate is 30Mb/s, So if the picture is bigger, the framerate is smaller.

Reading recorded frames from the camera to the host can be done in two different modes:

Single Frame Mode

Before reading a frame in 'Single Frame Mode' you must activate streaming by performing the command **AcquisitionStart**. Now single frames can be read from the camera just by setting their frame number in the feature **StreamSingleFrame**. After setting the frame number, streaming of the frame data to the host computer will start immediately.

A frame is addressed by its frame number which is the index of the frame in the camera internal frame buffer. The frame number starts with 0 for the first frame and goes up to **RecordNumberOfFramesPerSequence-1** for the last frame. If you did recording in multi sequence mode, you must select the sequence you want to read from with the feature **StreamSequenceSelector** before reading.

The programming flow:

```
StreamSequenceSelector = 0 or 1 or....
Read the feature RecordNumberOfFramesPerSequence
AcquisitionSource = Memory
Perform AcquisitionStart
Set StreamSingleFrame to the requested frame number
...
AcquisitionStop
```

Frame Burst Mode

In 'Frame Burst Mode' you can read a number of consecutive frames from the camera by defining a frame start number with the feature **StreamStartNumber** and the number of the last frame to read by the feature **StreamEndNumber**.

The address of the frame start number has to be less or equal than the frame end number. The address range for **StreamStartNumber** is 0 up to the value of **StreamEndNumber**. The range **StreamEndNumber** is **StreamStartNumber** up to **RecordNumberOfFramesPerSequence-1**. Before reading you must activate streaming by performing the command **AcquisitionStart**. To start streaming the defined sequence perform the command **StreamSequenceStart**. Data transmission stops automatically after the last frame of the sequence. While data transmission is running, the camera state changes to 'streaming' and will be set to 'idle' after streaming ends. Select the sequence from where you want to read by setting the **StreamSequenceSelector** before reading,

The programming flow of a streaming is:

StreamSequenceSelector = 0 or 1 or....

Read the feature RecordNumberOfFramesPerSequence

StreamEndNumber = x (Range: StreamStartNumber ...RecordNumberOfFramesPerSequence-1)

StreamStartNumber = x (Range: 0.... StreamEndNumber)

AcquisitionSource = Memory

Perform AcquisitionStart

Perform StreamSequenceStart (CameraState gets Streaming)

The feature **CameraState** will be set automatically to Idle after streaming ends.

....

AcquisitionStop

6 ImageBLITZ® Programming Guide

Mikrotron's patented ImageBLITZ® Technology provides an extremely fast, accurate and sensitive trigger for image capture in high speed vision applications.

The trigger event has to be unique in such a way that it corresponds with a specific outstanding grey level change. Thus, the event can be identified by the ImageBLITZ® sensitive line segment, making all elaborate sensing device obsolete. Process synchronization takes care of itself.

Whenever the ImageBLITZ® detects a trigger condition, the camera captures an image and sends it to the frame grabber or directly to the application for processing , storage and later review or for documentation.

IBThreshold defines the absolute limit for the difference in the gray values within the image details. The value range is 1 to 255.

With a small value changings in the image detail let the ImageBLITZ trigger,with a big value the ImageBLITZ trigger gets less sensitive. Under normal conditions, this value should be between 10 and 70.

IBReleaseCondition defines the relative size of the area within the ImageBLITZ window that must be different from the reference frame before ImageBLITZ triggers. The value range is 0 to 100 percent. If you use a small value, e.g. 5, then ImageBLITZ triggers when 5% of the image details are different.

The ImageBLITZ reference frame is the first frame of a recording sequence In live mode, this is the first frame after AcquisitionStart, if IBEnable is active. If IBEnable is not active if AcquisitionStart is given, the reference frame is the first frame after IBEnable.In record mode the reference frame is the first frame after starting the recording.

The programming flow of ImageBLITZ® is:

AcquisitionSource = Live

AcquisitionStart

IBEnable = True(catch the reference frame)

IBWindowVisible = True

set up the features **IBWidth**, **IBOffsetX**, **IBHeight**, **IBOffsetY** to the desired value.

set up the features **IBThreshold** and **IBReleaseCondition**

...

7 DecimationXY mode (Subsampling)

With the feature **DecimationXY** is enabled the sensor skips every second row and column. Therefore the maximum width is 640 pixel(EoSens Cube6 &mini1) and the maximum height is 512 pixel(EoSens Cube6 &mini1). The advantage is that a lens with the same focal distance covers the identical image area but with a higher framerate. This feature is not available for EoSens Cube7 and mini2.

A ROI of 640x512 covers the whole sensor area and is equals to the reproduction scale of 1280x1024 in normal mode. Switching this feature on/off, halves/doubles the output image size.

The programming flow to enable DecimationXY mode:

```
make sure: CameraState = IDLE , Width = 1280, Height = 1024 , AcquisitionSource = Live
DecimationXY = True
Width = 640, wait 1s
Height = 512, wait 1s
(AcquisitionStart)
...
```

The programming flow to disable DecimationXY mode:

```
make sure: CameraState = IDLE , Width = 640, Height = 512 ,AcquisitionSource = Live
DecimationXY = False
Width = 1280, wait 1s
Height = 1024, wait 1s
(AcquisitionStart)
..
```

Note: Please do nothing between DecimationXY ,Width and Height

8 Feature overview

| Name | Type | Description | Values | Access |
|--------------------------------------|---------|--|---|--------|
| AcquisitionAndTriggerControls | | | | |
| AcquisitionFrameRateRaw | Integer | This feature sets the frame rate in dependency of the given frame size (in Hertz) | 1...max. acquisition rate in dependency of the current frame size, Inc. 1 | RW |
| AcquisitionMode | Enum | This feature controls the acquisition mode of the device. | Continuous | RW |
| AcquisitionSource | Enum | This feature defines the source of image acquisition (live image or image from camera image buffer) | Live, Memory | RW |
| AcquisitionStart | Command | This feature starts the Acquisition of the device. | | WO |
| AcquisitionStop | Command | This feature stops the Acquisition of the device at the end of the current Frame. | | WO |
| ExposureMode | Enum | This feature is used to set the operation mode of the Exposure (or shutter). | Timed | RW |
| ExposureTimeRaw | Integer | This feature is used to set the Exposure time in multiples of microseconds when ExposureMode is Timed. | 2.. max. exposure time in dependency of the current frame rate, Inc. 1 | RW |

| Name | Type | Description | Values | Access |
|----------------------------------|---------|---|---|--------|
| AnalogControls | | | | |
| BlackLevelRaw | Integer | This feature controls the analog black level as a raw integer value. | 50...200, Inc. 1 | RW |
| GainRaw | Integer | This feature controls the selected gain as a raw integer value. | EoSens Cube6 & mini1 100..400 Inc. 50 EoSens Cube7 & mini2 100,150,200 | RW |
| CustomFeatures | | | | |
| CameraState | Enum | This feature returns the current transmission state of the camera | Idle, Live, Streaming, Recording | RO |
| IRIGBEnable | Boolean | This feature enables/disables the processing of an IRIG-B signal connected to the according camera input pin (this feature is not available on all cameras) | On, Off | RW |
| Temperature | Integer | This feature reads the temperature of the camera body as raw data. | | RO |
| DecimationXY | Boolean | f this feature is enabled the sensor skips every second row and column. | On, Off | RW |
| CustomFeatures\ExtTrigger | | | | |
| ExtTriggerEnable | Boolean | This Feature enables/disables the input port for an external trigger signal | On, Off | RW |
| ExtTriggerPolarity | Enum | This feature defines if the camera trigger gets active on the rising or falling edge of the external trigger signal | raising, falling | RW |

| Name | Type | Description | Values | Access |
|----------------------------------|---------|---|-----------------------------------|--------|
| CustomFeatures\ImageBlitz | | | | |
| IBEnable | Boolean | This feature enables/disables the Mikrotron ImageBlitz feature | On, Off | RW |
| IBHeight | Integer | This feature defines the height of the ImageBlitz window in units of lines | 1...SensorHeight-IBOffset, Inc. 1 | RW |
| IBOffsetX | Integer | This feature defines the right offset of the ImageBlitz window starting from the first sensor row, in units of pixels | 0...SensorWidth-IBWidth, Inc. 16 | RW |
| IBOffsetY | Integer | This feature defines the offset of the ImageBlitz window starting from the first sensor line, in units of lines | 0...SensorHeight-IBHeight, Inc. 1 | RW |
| IBReleaseCondition | Integer | This feature defines the number of pixel to fulfill the ImageBlitz trigger condition | 1...100, Inc. 1 | RW |

| Name | Type | Description | Values | Access |
|-------------------------------------|---------|--|---|--------|
| IBThreshold | Integer | This feature defines the trigger threshold of the ImageBlitz feature | 1...255, Inc. 1 | RW |
| IBWidth | Integer | This feature defines the width of the ImageBlitz window in units of pixels | 16...SensorWidth-IBOffsetX, Inc. 16 | RW |
| IBWindowVisible | Boolean | This feature shows/hides the ImageBlitz trigger window | On, Off | RW |
| CustomFeatures\MultipleSlope | | | | |
| MultipleSlopeDualRaw | Integer | This feature sets the value for Dual Multiple Slope in percent of the exposure time | Min. and max. in dependency of the exposure time and MultipleSlopeTripleRaw if TripleSlope mode is active, Inc. 1 | RW |
| MultipleSlopeMode | Enum | This feature defines the Multiple Slope Mode | NormalShutter, DualSlope, TripleSlope | RW |
| MultipleSlopeTripleRaw | Integer | This feature sets the value for Triple Multiple Slope in percent of the exposure time(only for EoSens Cube6 & mini1) | Min. and max. in dependency of the exposure time and MultipleSlopeDualRaw, Inc. 1 | RW |
| CustomFeatures\Recording | | | | |
| RecordBurstCnt | Integer | This feature defines the number of consecutive frames to record after an external trigger or an image trigger occurs. Only available in single recording mode. | 0...1022, Inc. 1 | RW |
| RecordCameraBufferSize | Integer | This feature returns the size of the camera internal image buffer in Giga Bytes | | RO |
| RecordCurrentSequence Number | Integer | If recording is running, this feature returns the number of the currently active sequence | 0...15 | RO |

| Name | Type | Description | Values | Access |
|----------------------------------|---------|--|---|--------|
| RecordMaxFrames | Integer | This feature returns the max. possible frames of all sequences | | RO |
| RecordMaxFrames PerSequence | Integer | This feature returns the number of max. possible frames per sequence | | RO |
| RecordMode | Enum | This feature switches recording mode between continuous and single recording. | Single, Continuous | RW |
| RecordNumberOfFrames | Integer | This feature returns the number of recorded frames of all sequences | | RO |
| RecordNumberOfFrames PerSequence | Integer | This feature returns the number of recorded frames per sequence | | RO |
| RecordNumberOfSequences | Enum | This feature defines the number of different recording sequences | One, Two, Four, Eight, Sixteen | RW |
| RecordStandby | Boolean | If this feature is true, the camera enters standby mode after next/current recording ends | On, Off | RW |
| RecordStart | Command | This command starts recording to the camera image buffer | | WO |
| RecordStop | Command | This command stops recording to the camera image buffer | | WO |
| RecordTrailer | Integer | This feature defines the number of trailing frames after recording stop | 0...RecordMaxFrames PerSequence-1, Inc. 1 | RW |
| RecordWhileTriggerActive | Boolean | If this feature is true the camera stores frames until the external trigger or the image trigger is active. Only available in single recording mode. | On, Off | RW |

| Name | Type | Description | Values | Access |
|---------------------------------------|---------|--|--|--------|
| CustomFeatures\Streaming | | | | |
| StreamEndNumber | Integer | This feature defines the end frame number for reading frames from image buffer | StreamStartNumber...RecordNumber OfFramesPerSequence-1, Inc. 1 | RW |
| StreamSequenceSelector | Integer | This feature selects the current sequence for streaming | 0...15, Inc. 1 | RW |
| StreamSequenceStart | Command | This feature starts streaming of a sequence of images. | | WO |
| StreamSingleFrame | Integer | This feature reads a single frame from an image buffer | 0...RecordNumberOfFrames PerSequence-1, Inc. 1 | RW |
| StreamStartNumber | Integer | This feature defines the start frame number for reading frames from an image buffer | 0...StreamEndNumber, Inc. 1 | RW |
| CustomFeatures\Synchronisation | | | | |
| SyncARMPolarity | Enum | This feature defines the polarity of the ARM signal | positive, negative | RW |
| SyncInEnable | Boolean | This feature enables the synchronisation input pin of the camera connector for ext. camera synchronisation | On, Off | RW |
| SyncInPolarity | Enum | This feature defines if the camera is synchronised to the falling or raising edge of the sync in signal | raising, falling | RW |
| SyncOutSelect | Enum | This feature defines the camera signal to connec to the Sync Out / ARM pin of the camera connector | ARM, SyncOut | RW |

| Name | Type | Description | Values | Access |
|--------------------------|---------|---|---|--------|
| DeviceInformation | | | | |
| DeviceID | String | This feature stores a camera identifier. | Length: 16 | RO |
| DeviceManufacturerInfo | String | This feature provides extended manufacturer information about the device. | Length: 48 | RO |
| DeviceModelName | String | This feature provides the model of the device. | Length: 32 | RO |
| DeviceReset | Command | This command is used to reset the device and to put it in its power up state. | | WO |
| DeviceScanType | Enum | This feature specifies the scan type of the sensor. | Areascan | RO |
| DeviceUserID | String | This feature stores a user-programmable identifier. | Length: 16 | RW |
| DeviceVendorName | String | This feature provides the name of the manufacturer of the device. | Length: 32 | RO |
| DeviceVersion | String | This feature provides the version of the device. | Length: 32 | RO |
| ImageSizeControl | | | | |
| Height | Integer | This feature represents the actual image height expelled by the camera (in pixels). | 2...HeightMax - OffsetY, Inc. 2 | RW |
| HeightMax | Integer | This feature represents the maximum height (in pixels) of the image after vertical binning, decimation or any other function changing the vertical dimensions of the image. | EoSens Cube6 & mini1 : 1024 EoSens Cube7 & mini2 : 1710 | RO |
| OffsetX | Integer | This feature represents the horizontal offset from the origin to the AOI (in pixels). | 0...WidthMax-Width, EoSens Cube6 & mini1 :Inc. 96 EoSens Cube7 & mini2 :Inc. 64 | RW |

| Name | Type | Description | Values | Access |
|----------------------------------|---------|--|--|--------|
| OffsetY | Integer | This feature represents the vertical offset from the origin to the AOI (in pixels). | 0...HeightMax - Height, Inc: 2 | RW |
| PixelCoding | Enum | This feature indicates the coding of the pixels in the image. | Mono | RO |
| PixelColorFilter | Enum | This feature indicates the type of color filter that is applied to the image. | None | RO |
| PixelFormat | Enum | This feature indicates the format of the pixel to use during the acquisition. | Mono8 | RO |
| PixelSize | Enum | This feature indicates the total size in bits of a pixel of the image. | Bpp8 | RO |
| SensorDigitizationTaps | Enum | This feature represents the number of digitized samples outputted simultaneously by the camera A/D conversion stage. | 2 | RW |
| SensorHeight | Integer | This feature indicates the effective height of the sensor in pixels. | EoSens Cube6 & mini1 : 1024 EoSens Cube7 & mini2 : 1710 | RO |
| SensorWidth | Integer | This feature indicates the effective width of the sensor in pixels. | EoSens Cube6 & mini1 : 1280 EoSens Cube7 & mini2 : 1696 | RO |
| TestImageSelector | Enum | This feature selects the type of test image that is expelled by the camera. | Off, IPEngineTestPattern | RW |
| Width | Integer | This feature represents the actual image width expelled by the camera (in pixels). | 128...WidthMax-OffsetX, Inc. 16 | RW |
| WidthMax | Integer | This feature represents the maximum width (in pixels) of the image after horizontal binning, decimation or any other function changing the horizontal dimensions of the image. | EoSens Cube6 & mini1 : 1280 EoSens Cube7 & mini2 : 1696 | RO |
| CustomFeatures\CustomData | | | | |
| Data00 | Integer | This feature stores custom data in non volatile memory. | | RW |

| | | | | |
|------------------------|---------|---|-------------------|----|
| Data01 | Integer | This feature stores custom data in non volatile memory. | | RW |
| Data02 | Integer | This feature stores custom data in non volatile memory. | | RW |
| Data03 | Integer | This feature stores custom data in non volatile memory. | | RW |
| UserSets | | | | |
| UserSetSelector | Enum | This feature selects the feature User Set to load, save or configure | Default, UserSet1 | RW |
| UserSetLoad | Command | This feature loads the User Set specified by UserSetSelector to the device and makes it active. | | W |
| UserSetSave | Command | This feature save the User Set specified by UserSetSelector to the non-volatile memory of the device. | | W |
| UserSetDefaultSelector | Enum | This feature selects the feature User Set to load and make active when the device is reset. | Default, UserSet1 | RW |