GenICam

Standard Features Naming Convention

Version 1.5

Table of Contents

History

Version	Date	Changed by	Change
0.01	14.02.2006	Eric Carey, DALSA Coreco	Initial version based on the GenICam standard feature list document of the GigE Vision/GenICam joint sub-committee. This version is intended to be the official feature naming convention to be used for GigE Vision cameras. Original contributors: **Basler** (Fritz Dierks, Thies Moeller, Andreas Gäer), **Leutron Vision** (Jan Becvar), **DALSA Coreco** (Eric Carey), **Euresys** (Jean-Michel Wintgens), **MVTec** (Christoph Zierl), **National Instruments** (Chris Graf), **Stemmer** (Sascha Dorenbeck), **SICKIVP** (Mattias Johannesson), **JAI** (Ole Krogh Jørgensen), **Matrox** (Stephane Maurice)
0.02	16.03.2006	Stephane Maurice, Matrox	Define the new Acquisition, Trigger and I/O feature set. Introduced the notion of counters and grouped it with Timers in a separate section. Reviewed feature names for consistency and grouping.
draft 1.00	04.04.2006	Stephane Maurice, Matrox	Included modifications and corrections based on the feedbacks from version 0.02 to 0.9. Final Draft.

Version	Date	Changed by	Change
draft 1.00.01	06.06.2006	Stephane Maurice, Matrox	Changed PixelSize to Bpp8, Bpp10, … Removed all "_" in enumerations and all feature names.
draft 1.00.02	22.06.2006	Stephane Maurice, Matrox	Changed Software Trigger from TriggerMode to TriggerSource to permit 1394 DCAM feature compatibility. Removed ticks as standard unit for Raw time unit. Added AnyEdge as standard signal activation and event type. Added Line0 and UserOutput0 as standard optional names for enumeration. Added AcquisitionFrameRateRaw and AcquisitionLineRateRaw. Defined standard Event numbers that matches the GigEvision Event numbers.
draft 1.00.03	16.06.2007	Vincent Rowley, Pleora Technologies Inc.	Prepared Version 1.0. Removed the AIA logo. Fixed typos. Added a note with respect to how the GevMACAddress feature should be implemented. Added a note specifying that the GevCurrentIPConfiguration feature should not be used in production GenICam XML files since it will be deprecated in the next version of the present document. Fixed GevTimestampTickFrequency valid range.

Version	Date	Changed by	Change
draft 1.00.03 cont.	19.06.2007	Stephane Maurice, Matrox	Preparation for Version 1.0 continued: Added a note about the Selector usage specifying that they must not introduce side effect when their value is changed. Removed GiGEVision logo since the Standard Feature List is now part of the GenICam standard. Specified that features with big value such as GevMACAddress, GEVTimestampTickFrequency and GEVTimestampValue must be returned as a single 64 bit values.
Release 1.00.00	20.06.2007	Stephane Maurice, Matrox	Final release Version 1.00 Note: This release includes all the features as they were defined in the draft 1.00.02 referenced in the final GigE Vision specification version 1.00.
Version 1.01.01	04.07.2007	Vincent Rowley, Pleora Technologies Inc.	Added SensorTaps, SensorDigitizationTaps, GevCurrentIPConfigurationLLA, GevCurrentIPConfigurationDHCP, GevCurrentIPConfigurationPersistentIP and GevIPConfigurationStatus features. Deprecated GevCurrentIPConfiguration. Added OpenAccess to the list of valid values for the GevCCP feature.
Version 1.01.02	24.07.2007	Stephane Maurice Matrox	Added the PixelFormat description chapter and note about zero based user bits.
Release 1.1	2.10.2007	Stephane Maurice, Matrox	Final release Version 1.1

Version	Date	Changed by	Change
Version 1.1.01	10.09.2007	Thies Möller, Basler	Created chapter for File Access.
Version 1.1.02	12.01.2008	Stephane Maurice, Matrox Vincent Rowley , Pleora	Review and modification to the File Access features proposal.
Release 1.2	29.04.2008	Stephane Maurice, Matrox	SFNC 1.2 including the File Access features and corrections. Also removed the PixelFormat description chapter and GEV event numbers.
Version 1.2.01	17.07.2008	Karsten Ingeman Christensen, JAI	Merged with recommended visibility proposal from JAI and commented by Vincent Rowley, Pleora
Release 1.2.1	19.08.2008	Stephane Maurice, Matrox	SFNC 1.2.1 including the recommended visibility.
Version 1.2.12	28.10.2008	Stephane Maurice, Matrox Thies Möller, Basler	Matrox: Created draft for 1.3 including: minors corrections, deprecated Raw and Abs feature and deprecated GigEVision Event, Changed sections names and created according category features, added Root, Device, TLParamsLocked, PixelClock, Temperature features and made ICommand optionally readable, … Basler: Action command was added.
Version 1.2.13	05.05.2009	Stephane Maurice, Matrox	Deprecated all the GEVSupported… feature to regroup them in a selector. Added Color Transformation features. Action command reworked and moved in a separate section. Added Event data delivery features.

Version	Date	Changed by	Change
Version 1.2.14	20.05.2009	Stephane Maurice, Matrox	Deprecated Line0RisingEdge, …compound enumeration in CountersEventSource and created separate CounterEventActivation and CounterResetActivation features to be consistent with the trigger features. Made CounterValue and TimerValue Writable. Modified descriptions to be able to extract tooltips and descriptions for the reference SFNC XML. Added a VBA macro to be able to generate machine readable version of the SFNC. Added a VBA macro to be able to generate the Features summary (Chapter 2) automatically. Changed units to have a standard notation.
Release 1.3	11.08.2009	Stephane Maurice, Matrox	SFNC 1.3 release including the changes since version 1.2.1.
Draft 1.4a-c	05.01.2009 and 22.01.2010	Vincent Rowley, Pleora Technologies Inc.	Added GigE Vision 1.2 support. Added missing Bpp36 and Bpp48 enumeration entries for PixelSize feature. Added missing RawPacked enumeration entry for PixelCoding feature. Updated support level for GevSCPInterfaceIndex feature in order to be consistent with related features. Clarified text when necessary and fixed typos. Corrected some feature descriptions.

Version	Date	Changed by	Change
Release 1.4	17.03.2010	Stephane Maurice, Matrox	Minor fixes to remove mistakes. YUV422YUYVPacked was removed, changed all the ExposureTimeAuto to ExposureAuto. Corrected GevGVCPPendingAck and GevManifestSecondaryURL names. Added ChunkTimer and ChunkCounter to ChunkSelector. Updated VB macros.
Release 1.5	22.11.2010	Stephane Maurice, Matrox	- Added Camera Link related features. - ActionSelector now >0. - Added Bpp30 to PixelSize. -Added RGB16Packed, BGR16Packed, BGR10V1Packed and BGR10V2Packed to pixel Format. - DeviceUserID is now recommended to be an empty string. - GenICam Access section added. - DeviceSFNCVersion… features added. - Clarified and corrected points in Counter and Timer section. - Added new TimerReset feature and removed LevelHigh and LevelLow in CounterEventActivation of Counter and Timer section. - Updated Chunk section to correct inconsistencies and add missing items. - Minor fixes to remove mistakes. - Updated VB macros to extract the description of the enumerations and fix other minor parsing issues.

1 Introduction

The GenICam technology allows exposing arbitrary features of a camera through a unified API and GUI. Each feature can be defined in an abstract manner by its name, interface type, unit of measurement and behavior. The GenApi module of the GenICam standard defines how to write a camera description file that describes a specific camera’s mapping.

GenICam alone is sufficient if the user wants to write software for a specific camera only because all features of the camera are accessible through the GenICam provided API. However if the user wants to write **generic software** for a whole class of cameras then GenICam alone is not sufficient. In addition, the software vendors and the camera vendors have to agree on a common naming convention for the standard features.

For technical and historical reasons the different transport layer technologies (GigE, 1394, Camera Link, etc.) might require slightly different feature sets. This is why this document addresses mainly the cameras compliant to the GigE Vision standard. The naming convention is however targeting maximum reusability by other existing and future transport layer technologies. It provides the definitions of **standard use cases** and **standard features**. The goal is to cover and to standardize the naming convention used in all those basic use cases where the implementation by different vendors would be very similar anyway.

Features are tagged within this document according to the following list:

- M: **mandatory -** Must be implemented to achieve compliance with the GigE Vision standard

- R: recommended **-** This feature adds important aspects to the use case and should respect the naming convention.*

- O: optional - This feature is less critical. Nevertheless, it is considered and should respect the naming convention.

For additional details about the mandatory features please refer to the GigE Vision standard.

Recommended Visibility

According to the GenICam standard each feature can be assigned a "recommended visibility" using the <Visibility> element in the XML-files. The <Visibility> element defines the user level that should get access to the feature. Possible values are: Beginner, Expert, Guru and Invisible. The latter is required to make features show up in API, but not in the GUI.

The visibility does not affect the functionality of the features but is merely used by the GUI to decide which features to display based on the current user level. The purpose is mainly to insure that the GUI is not cluttered with information that is not intended at the current user level.

The following criteria's have been used for the assignment of recommended visibility:

- B: beginner - features that should be visible for ***all*** users via the GUI and API. This is the default visibility in the GenICam XML-files and will be used if the <Visibility> element is omitted. The number of features with "beginner" visibility should be limited to all **basic** features of the devices so the GUI display is well-arranged and is easy to use.

- E: expert - features that require a more in-depth knowledge of the camera functionality. This is the preferred visibility level for all advanced features in the cameras.

- G: guru – 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.

- I: invisible – features that should be kept hidden for the GUI users but still be available via the API.

1.1 Conventions

Selector

A selector is used to index which instance of the feature is accessed in situations where multiple instances of a feature exist (for instance, the analog gain for each separate channel for the red/green/blue component of a color camera). The selector is a separate feature that is typically an IEnumeration or an IInteger.

Features dependent on the Selector are expressed using the C language convention for arrays: a pair of brackets follows the feature name, like in SelectedFeature[Selector]. When the Selector is not present, one must deduce the feature is not an array.

Note that selectors must be used only to select the target features for subsequent changes. It is not allowed to change the behavior of a device in response to a change of a selector value.

Standard Units

The following abbreviations are used as standard units for features described in this document. Note that all units are using plain ASCII characters.

us	microseconds
ms	milliseconds
s	seconds
B	Bytes
Bps	Bytes per second
MBps	Mega Bytes per second
Mbps	Mega bit per second
Fps	Frame per second
dB	decibels
C	Celsius
Hz	Hertz

1.2 Acronyms

ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AIA	Automated Imaging Association
AOI	Area Of Interest
CRT	Cathode Ray Tube
DC	Direct Current
DHCP	Dynamic Host Configuration Protocol
EMVA	European Machine Vision Association
ID	Identifier
I/O	Input/Output
IP	Internet Protocol
LLA	Link-Local Address
LUT	Look-Up Table
M	Mandatory
O	Optional
R	Recommended or Read (depends on the context)
ROI	Region Of Interest
URL	Uniform Resource Locator
W	Write
XML	eXtensible Markup Language

2 Features Summary

This section provides a comprehensive list of the standard features covered by this document. The following sections provide more detailed explanation of each feature. In case of discrepancy, those sections describing the features in detail prevail.

2.1 Device Control

Contains the features related to the device and its sensor.

Table 2-1: Device Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
DeviceControl	R	ICategory	R	-	B	Category for Device information and control.
DeviceVendorName	R	IString	R	-	B	Name of the manufacturer of the device.
DeviceModelName	R	IString	R	-	B	Model of the device.
DeviceManufacturerInfo	R	IString	R	-	B	Manufacturer information about the device.
DeviceVersion	R	IString	R	-	B	Version of the device.
DeviceFirmwareVersion	R	IString	R	-	B	Version of the firmware in the device.
DeviceSFNCVersionMajor	R	IInteger	R	-	B	Major Version of the Standard Feature Naming Convention that was used to create the device`s XML.
DeviceSFNCVersionMinor	R	IInteger	R	-	B	Minor Version of the Standard Feature Naming Convention that was used to create the device`s XML.
DeviceSFNCVersionSubMinor	R	IInteger	R	-	B	Sub Minor Version of Standard Feature Naming Convention that was used to create the device`s XML.
DeviceManifestEntrySelector	O	IInteger	R/W	-	G	Selects the manifest entry to reference.
DeviceManifestXMLMajorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the major version number of the XML file of the selected manifest entry.
DeviceManifestXMLMinorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the minor version number of the XML file of the selected manifest entry.

DeviceManifestXMLSubMinorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the subminor version number of the XML file of the selected manifest entry.
DeviceManifestSchemaMajorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the major version number of the schema file of the selected manifest entry.
DeviceManifestSchemaMinorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the minor version number of the schema file of the selected manifest entry.
DeviceManifestPrimaryURL[DeviceManifestEntrySelector]	O	IString	R	-	G	Indicates the first URL to the XML device description file of the selected manifest entry.
DeviceManifestSecondaryURL[DeviceManifestEntrySelector]	O	IString	R	-	G	Indicates the second URL to the XML device description file of the selected manifest entry.
DeviceID	R	IString	R	-	E	Device Identifier (serial number).
DeviceUserID	O	IString	R/W	-	B	User-programmable Device Identifier.
DeviceReset	R	ICommand	W	-	G	Resets the device to its power up state.
DeviceRegistersStreamingStart	R	ICommand	(R)/W	-	G	Prepare the device for registers streaming without checking for consistency.
DeviceRegistersStreamingEnd	R	ICommand	(R)/W	-	G	Announce the end of registers streaming.
DeviceRegistersCheck	R	ICommand	(R)/W	-	E	Perform the validation of the current register set for consistency.
DeviceRegistersValid	R	IBoolean	R	-	E	Returns if the current register set is valid and consistent.
DeviceMaxThroughput	O	IInteger	R	Bps	E	Maximum bandwidth of the data that can be streamed out of the device.
DeviceTemperatureSelector	O	IEnumeration	R/W	-	E	Selects the location within the device, where the temperature will be measured.
DeviceTemperature[DeviceTemperatureSelector]	O	IFloat	R	C	E	Device temperature in degrees Celsius (C).
DeviceClockSelector	O	IEnumeration	R/(W)	-	E	Selects the clock frequency to access from the device.
DeviceClockFrequency[DeviceClockSelector]	O	IFloat	R/(W)	Hz	E	Returns the frequency in Hertz of the selected Clock.
DeviceSerialPortSelector	R	IEnumeration	R/(W)	-	E	Selects which device serial port to control.

DeviceSerialPortBaudRate[SerialPortSelector]	R	IEnumeration	R/(W)	-	E	This feature controls the baud rate used by the selected device's serial port.
DeviceScanType	R	IEnumeration	R/(W)	-	E	Scan type of the sensor of the device.

2.2 Image Format Control

Contains the features related to the format of the transmitted image.

Table 2-2: Image Format Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ImageFormatControl	R	ICategory	R	-	B	Category for Image Format Control features.
SensorWidth	R	IInteger	R	-	E	Effective width of the sensor in pixels.
SensorHeight	R	IInteger	R	-	E	Effective height of the sensor in pixels.
SensorTaps	O	IEnumeration	R/(W)	-	E	Number of taps of the camera sensor.
SensorDigitizationTaps	O	IEnumeration	R/(W)	-	E	Number of digitized samples outputted simultaneously by the camera A/D conversion stage.
WidthMax	R	IInteger	R	-	E	Maximum width (in pixels) of the image.
HeightMax	R	IInteger	R	-	E	Maximum height (in pixels) of the image.
Width	M	IInteger	R/(W)	-	B	Width of the Image provided by the device (in pixels).
Height	M	IInteger	R/(W)	-	B	Height of the image provided by the device (in pixels).
OffsetX	R	IInteger	R/W	-	B	Horizontal offset from the origin to the area of interest (in pixels).
OffsetY	R	IInteger	R/W	-	B	Vertical offset from the origin to the area of interest (in pixels).
LinePitch	R	IInteger	R/W	B	E	Total number of bytes between 2 successive lines.
BinningHorizontal	O	IInteger	R/W	-	E	Number of horizontal photo-sensitive cells to combine together.
BinningVertical	O	IInteger	R/W	-	E	Number of vertical photo-sensitive cells to combine together.

DecimationHorizontal	O	IInteger	R/W	-	E	Horizontal sub-sampling of the image.
DecimationVertical	O	IInteger	R/W	-	E	Vertical sub-sampling of the image.
ReverseX	R	IBoolean	R/W	-	E	Flip horizontally the image sent by the device.
ReverseY	R	IBoolean	R/W	-	E	Flip vertically the image sent by the device.
PixelFormat	M	IEnumeration	R/(W)	-	B	Format of the pixel provided by the device.
PixelCoding	R	IEnumeration	R/(W)	-	E	Coding of the pixels in the image.
PixelSize	R	IEnumeration	R/(W)	-	E	Total size in bits of a pixel of the image.
PixelColorFilter	R	IEnumeration	R/(W)	-	E	Type of color filter that is applied to the image.
PixelDynamicRangeMin	O	IInteger	R/W	-	E	Minimum value that can be returned during the digitization process.
PixelDynamicRangeMax	O	IInteger	R/W	-	E	Maximum value that will be returned during the digitization process.
TestImageSelector	O	IEnumeration	R/W	-	B	Selects the type of test image that is sent by the camera.

2.3 Acquisition Control

Contains the features related to image acquisition, including the triggering mode.

Table 2-3: Acquisition Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
AcquisitionControl	R	ICategory	R	-	B	Category for the acquisition and trigger control features.
AcquisitionMode	M	IEnumeration	R/W	-	B	Sets the acquisition mode of the device.
AcquisitionStart	M	ICommand	(R)/W	-	B	Starts the Acquisition of the device.
AcquisitionStop	M	ICommand	(R)/W	-	B	Stops the Acquisition of the device at the end of the current Frame.
AcquisitionAbort	R	ICommand	(R)/W	-	E	Aborts the acquisition immediately.
AcquisitionArm	O	ICommand	(R)/W	-	E	Arms the device before an AcquisitionStart command.
AcquisitionFrameCount	R	IInteger	R/W	-	B	Number of frames to acquire in MultiFrame Acquisition mode.
AcquisitionFrameRate	R	IFloat	R/W	Hz	B	Controls the acquisition rate (in Hertz) at which the frames are captured.
AcquisitionFrameRateAbs	R	IFloat	R/W	Hz	I	This feature is deprecated.
AcquisitionFrameRateRaw	O	IInteger	R/W	-	I	This feature is deprecated.
AcquisitionLineRate	R	IFloat	R/W	Hz	B	Controls the rate (in Hertz) at which the Lines in a Frame are captured.
AcquisitionLineRateAbs	R	IFloat	R/W	Hz	I	This feature is deprecated.
AcquisitionLineRateRaw	O	IInteger	R/W	-	I	This feature is deprecated.
AcquisitionStatusSelector	R	IEnumeration	R/W	-	E	Selects the internal acquisition signal to read using AcquisitionStatus.
AcquisitionStatus[AcquisitionStatusSelector]	R	IBoolean	R	-	E	Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.
TriggerSelector	R	IEnumeration	R/W	-	B	Selects the type of trigger to configure.

TriggerMode[TriggerSelector]	R	IEnumeration	R/W	-	B	Controls if the selected trigger is active.
TriggerSoftware[TriggerSelector]	R	ICommand	(R)/W	-	B	Generates an internal trigger.
TriggerSource[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the internal signal or physical input Line to use as the trigger source.
TriggerActivation[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the activation mode of the trigger.
TriggerOverlap[TriggerSelector]	R	IEnumeration	R/W	-	E	Specifies the type trigger overlap permitted with the previous frame.
TriggerDelay[TriggerSelector]	R	IFloat	R/W	us	E	Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.
TriggerDelayAbs[TriggerSelector]	R	IFloat	R/W	us	I	This feature is deprecated.
TriggerDelayRaw[TriggerSelector]	R	IInteger	R/W	-	I	This feature is deprecated.
TriggerDivider[TriggerSelector]	R	IInteger	R/W	-	E	Specifies a division factor for the incoming trigger pulses.
TriggerMultiplier[TriggerSelector]	R	IInteger	R/W	-	E	Specifies a multiplication factor for the incoming trigger pulses.
ExposureMode	R	IEnumeration	R/W	-	B	Sets the operation mode of the Exposure (or shutter).
ExposureTime	R	IFloat	R/W	us	B	Sets the Exposure time (in microseconds) when ExposureMode is Timed.
ExposureTimeAbs	R	IFloat	R/W	us	I	This feature is deprecated.
ExposureTimeRaw	O	IInteger	R/W	-	I	This feature is deprecated.
ExposureAuto	O	IEnumeration	R/W	-	B	Sets the automatic exposure mode when ExposureMode is Timed.

2.4 Digital I/O Control

Contains the features related to the control the general input and output pins of the device.

Table 2-4: Digital I/O Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
DigitalIOControl	R	ICategory	R	-	E	Category that contains the digital input and output control features.
LineSelector	R	IEnumeration	R/W	-	E	Selects the physical line (or pin) of the external device connector to configure.
LineMode[LineSelector]	O	IEnumeration	R/W	-	E	Controls if the physical Line is used to Input or Output a signal.
LineInverter[LineSelector]	R	IBoolean	R/W	-	E	Controls the invertion of the signal of the selected input or output Line.
LineStatus[LineSelector]	R	IBoolean	R	-	E	Returns the current status of the selected input or output Line.
LineStatusAll	O	IInteger	R	-	E	Returns the current status of all available Line signals at time of polling in a single bitfield.
LineSource[LineSelector]	R	IEnumeration	R/W	-	E	Selects which internal acquisition or I/O source signal to output on the selected Line.
LineFormat[LineSelector]	O	IEnumeration	R/W	-	E	Controls the current electrical format of the selected physical input or output Line.
UserOutputSelector	R	IEnumeration	R/W	-	E	Selects which bit of the User Output register will be set by UserOutputValue.
UserOutputValue[UserOutputSelector]	R	IBoolean	R/W	-	E	Sets the value of the bit selected by UserOutputSelector.
UserOutputValueAll	O	IInteger	R/W	-	E	Sets the value of all the bits of the User Output register.
UserOutputValueAllMask	O	IInteger	R/W	-	E	Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.

2.5 Counter and Timer Control

Contains the features related to the usage of programmable counters and timers.

Table 2-5: Counter and Timer Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
CounterAndTimerControl	R	ICategory	R	-	E	Category that contains the Counter and Timer control features.
CounterSelector	R	IEnumeration	R/W	-	E	Selects which counter to configure.
CounterEventSource[CounterSelector]	R	IEnumeration	R/W	-	E	Select the events that will be the source to increment the counter.
CounterEventActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the Activation mode Event Source signal.
CounterResetSource[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the signals that will be the source to reset the counter.
CounterResetActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the Activation mode of the Counter Reset Source signal.
CounterReset[CounterSelector]	R	ICommand	(R)/W	-	E	Does a software reset of the selected counter and starts it.
CounterValue[CounterSelector]	R	IInteger	R/W	-	E	Reads or writes the current value of the selected counter.
CounterValueAtReset[CounterSelector]	R	IInteger	R	-	E	Reads the value of the selected counter when it was reset by a trigger or by an explicit CounterReset command.
CounterDuration[CounterSelector]	R	IInteger	R/W	-	E	Sets the duration (or number of events) before the CounterEnd event is generated.
CounterStatus[CounterSelector]	R	IEnumeration	R	-	E	Returns the current state of the counter.
CounterTriggerSource[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the source to start the counter.
CounterTriggerActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the activation mode of the trigger to start the counter.
TimerSelector	R	IEnumeration	R/W	-	E	Selects which Timer to configure.
TimerDuration[TimerSelector]	R	IFloat	R/W	us	E	Sets the duration (in microseconds) of the Timer pulse.
TimerDurationAbs[TimerSelector]	R	IFloat	R/W	us	I	This feature is deprecated.

TimerDurationRaw[TimerSelector]	R	IInteger	R/W	-	I	This feature is deprecated.
TimerDelay[TimerSelector]	R	IFloat	R/W	us	E	Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before to start the Timer.
TimerDelayAbs[TimerSelector]	R	IFloat	R/W	us	I	This feature is deprecated.
TimerDelayRaw[TimerSelector]	R	IInteger	R/W	-	I	This feature is deprecated.
TimerReset[TimerSelector]	R	ICommand	(R)/W	-	E	Does a software reset of the selected timer and starts it.
TimerValue[TimerSelector]	R	IFloat	R/W	us	E	Reads or writes the current value (in microseconds) of the selected Timer.
TimerValueAbs[TimerSelector]	R	IFloat	R	us	I	This feature is deprecated.
TimerValueRaw[TimerSelector]	R	IInteger	R	-	I	This feature is deprecated.
TimerStatus[TimerSelector]	R	IEnumeration	R	-	E	Returns the current state of the Timer.
TimerTriggerSource[TimerSelector]	R	IEnumeration	R/W	-	E	Selects the source of the trigger to start the Timer.
TimerTriggerActivation[TimerSelector]	R	IEnumeration	R/W	-	E	Selects the activation mode of the trigger to start the Timer.

2.6 Event Control

Contains the features related to the generation of Event notifications by the device.

Table 2-6: Event Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
EventControl	R	ICategory	R	-	E	Category that contains Event control features.
EventSelector	R	IEnumeration	R/W	-	E	Selects which Event to signal to the host application.
EventNotification[EventSelector]	R	IEnumeration	R/W	-	E	Activate or deactivate the notification to the host application of the occurrence of the selected Event.

Name	Level	Interface	Access	Unit	Visibility	Description
EventFrameTriggerData	R	ICategory	R	-	E	Category that contains all the data features related to the FrameTrigger Event.
EventFrameTrigger	R	IInteger	R	-	E	Returns the unique Identifier of the FrameTrigger type of Event.
EventFrameTriggerTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the AquisitionTrigger Event.
EventFrameTriggerFrameID	R	IInteger	R	-	E	Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

Name	Level	Interface	Access	Unit	Visibility	Description
EventExposureEndData	R	ICategory	R	-	E	Category that contains all the data features related to the ExposureEnd Event.
EventExposureEnd	R	IInteger	R	-	E	Returns the unique identifier of the ExposureEnd type of Event.
EventExposureEndTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the ExposureEnd Event.
EventExposureEndFrameID	R	IInteger	R	-	E	Returns the unique Identifier of the Frame (or image) that generated the ExposureEnd Event.

Name	Level	Interface	Access	Unit	Visibility	Description
EventErrorData	R	ICategory	R	-	E	Category that contains all the data features related to the Error Event.
EventError	R	IInteger	R	-	E	Returns the unique identifier of the Error type of Event.
EventErrorTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the Error Event.
EventErrorFrameID	R	IInteger	R	-	E	If applicable, returns the unique Identifier of the Frame (or image) that generated the Error Event.
EventErrorCode	R	IInteger	R	-	E	Returns an error code for the error(s) that happened.

2.7 Analog Control

Contains the features related to the video signal conditioning in the analog domain.

Table 2-7: Analog Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
AnalogControl	O	ICategory	R	-	B	Category that contains the Analog control features.
GainSelector	O	IEnumeration	R/W	-	B	Selects which Gain is controlled by the various Gain features.
Gain[GainSelector]	O	IFloat	R/W	-	B	Controls the selected gain as an absolute physical value.
GainRaw[GainSelector]	O	IInteger	R/W	-	I	This feature is deprecated.
GainAbs[GainSelector]	O	IFloat	R/W	-	I	This feature is deprecated.
GainAuto[GainSelector]	O	IEnumeration	R/W	-	B	Sets the automatic gain control (AGC) mode.
GainAutoBalance	O	IEnumeration	R/W	-	B	Sets the mode for automatic gain balancing between the sensor color channels or taps.
BlackLevelSelector	O	IEnumeration	R/W	-	E	Selects which Black Level is controlled by the various Black Level features.
BlackLevel[BlackLevelSelector]	O	IFloat	R/W	-	E	Controls the analog black level as an absolute physical value.
BlackLevelRaw[BlackLevelSelector]	O	IInteger	R/W	-	I	This feature is deprecated.
BlackLevelAbs[BlackLevelSelector]	O	IFloat	R/W	-	I	This feature is deprecated.
BlackLevelAuto[BlackLevelSelector]	O	IEnumeration	R/W	-	E	Controls the mode for automatic black level adjustment.
BlackLevelAutoBalance	O	IEnumeration	R/W	-	E	Controls the mode for automatic black level balancing between the sensor color channels or taps.
WhiteClipSelector	O	IEnumeration	R/W	-	E	Selects which White Clip to control.
WhiteClip[WhiteClipSelector]	O	IFloat	R/W	-	E	Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.
WhiteClipRaw[WhiteClipSelector]	O	IInteger	R/W	-	I	This feature is deprecated.
WhiteClipAbs[WhiteClipSelector]	O	IFloat	R/W	-	I	This feature is deprecated.

BalanceRatioSelector	O	IEnumeration	R/W	-	E	Selects which Balance ratio to control.
BalanceRatio[BalanceRatioSelector]	O	IFloat	R/W	-	E	Controls ratio of the selected color component to a reference color component.
BalanceRatioAbs[BalanceRatioSelector]	O	IFloat	R/W	-	I	This feature is deprecated.
BalanceWhiteAuto	O	IEnumeration	R/W	-	E	Controls the mode for automatic white balancing between the color channels.
Gamma	O	IFloat	R/W	-	B	Controls the gamma correction of pixel intensity.

2.8 LUT Control

Contains the features related to the look-up table (LUT) control.

Table 2-8: Lut Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
LUTControl	O	ICategory	R	-	E	Category that includes the LUT control features.
LUTSelector	O	IEnumeration	R/W	-	E	Selects which LUT to control.
LUTEnable[LUTSelector]	O	IBoolean	R/W	-	E	Activates the selected LUT.
LUTIndex[LUTSelector]	O	IInteger	R/W	-	G	Control the index (offset) of the coefficient to access in the selected LUT.
LUTValue[LUTSelector][LUTIndex]	O	IInteger	R/W	-	G	Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.
LUTValueAll[LUTSelector]	O	IRegister	R/W	-	G	Accesses all the LUT coefficients in a single access without using individual LUTIndex.

2.9 GenICam Access

Contains the features related to GenIcam access.

Table 2-9: GenICam Access Summary

Name	Level	Interface	Access	Unit	Visibility	Description
Root	M	ICategory	R	-	B	Provides the Root of the GenICam features tree.
Device	M	IPort	R/W	-	I	Provides the default GenICam port of the Device.
TLParamsLocked	M	IInteger	R/W	-	I	Used by the Transport Layer to prevent critical features changes during acquisition.

2.10 Transport Layer Control

Contains the features related to the Transport Layer Control.

Table 2-10: Transport Layer Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
TransportLayerControl	R	ICategory	R	-	B	Category that contains the Transport layer control features.
PayloadSize	M	IInteger	R	B	E	Provides the number of bytes transferred for each image or chunk on the stream channel.
GevVersionMajor	R	IInteger	R	-	E	Major version of the specification.
GevVersionMinor	R	IInteger	R	-	E	Minor version of the specification.
GevDeviceModeIsBigEndian	O	IBoolean	R	-	G	Endianess of the device registers.
GevDeviceClass	O	IEnumeration	R	-	G	Returns the class of the device.
GevDeviceModeCharacterSet	O	IEnumeration	R	-	G	Character set used by all the strings of the bootstrap registers.
GevInterfaceSelector	O	IInteger	R/W	-	B	Selects which physical network interface to control.
GevMACAddress[GevInterfaceSelector]	O	IInteger	R	-	B	MAC address of the network interface.
GevSupportedOptionSelector	O	IEnumeration	R/W	-	E	Selects the GEV option to interrogate for existing support.
GevSupportedOption[GevSupportedOpt	O	IBoolean	R	-	E	Returns if the selected GEV option is supported.

ionSelector]						
GevSupportedIPConfigurationLLA[GevInterfaceSelector]	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedIPConfigurationDHCP[GevInterfaceSelector]	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedIPConfigurationPersistentIP[GevInterfaceSelector]	O	IBoolean	R	-	I	This feature is deprecated.
GevCurrentIPConfiguration[GevInterfaceSelector]	O	IEnumeration	R/W	-	I	This feature is deprecated.
GevCurrentIPConfigurationLLA[GevInterfaceSelector]	O	IBoolean	R/W	-	B	Controls whether the Link Local Address IP configuration scheme is activated on the given network interface.
GevCurrentIPConfigurationDHCP[GevInterfaceSelector]	O	IBoolean	R/W	-	B	Controls whether the DHCP IP configuration scheme is activated on the given network interface.
GevCurrentIPConfigurationPersistentIP[GevInterfaceSelector]	O	IBoolean	R/W	-	B	Controls whether the PersistentIP configuration scheme is activated on the given network interface.
GevCurrentIPAddress[GevInterfaceSelector]	O	IInteger	R	-	B	Reports the IP address for the given network interface.
GevCurrentSubnetMask[GevInterfaceSelector]	O	IInteger	R	-	B	Reports the subnet mask of the given interface.
GevCurrentDefaultGateway[GevInterfaceSelector]	O	IInteger	R	-	B	Reports the default gateway IP address to be used on the given network interface.
GevIPConfigurationStatus[GevInterfaceSelector]	O	IEnumeration	R	-	B	Reports the current IP configuration status.
GevFirstURL	O	IString	R	-	G	Indicates the first URL to the XML device description file.
GevSecondURL	O	IString	R	-	G	Indicates the second URL to the XML device description file.
GevNumberOfInterfaces	O	IInteger	R	-	E	Indicates the number of physical network interfaces supported by this device.
GevPersistentIPAddress[GevInterfaceSelector]	O	IInteger	R/W	-	B	Controls the Persistent IP address for this network interface.

GevPersistentSubnetMask[GevInterfaceSelector]	O	IInteger	R/W	-	B	Controls the Persistent subnet mask associated with the Persistent IP address on this network interface.
GevPersistentDefaultGateway[GevInterfaceSelector]	O	IInteger	R/W	-	B	Controls the persistent default gateway for this network interface.
GevLinkSpeed[GevInterfaceSelector]	O	IInteger	R	Mbps	E	Indicates the speed of transmission negotiated by the given network interface.
GevMessageChannelCount	O	IInteger	R	-	E	Indicates the number of message channels supported by this device.
GevStreamChannelCount	O	IInteger	R	-	E	Indicates the number of stream channels supported by this device.
GevSupportedOptionalCommandsUserDefinedName	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedOptionalCommandsSerialNumber	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedOptionalCommandsEVENTDATA	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedOptionalCommandsEVENT	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedOptionalCommandsPACKETRESEND	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedOptionalCommandsWRITEMEM	O	IBoolean	R	-	I	This feature is deprecated.
GevSupportedOptionalCommandsConcatenation	O	IBoolean	R	-	I	This feature is deprecated.
GevHeartbeatTimeout	O	IInteger	R/W	ms	G	Controls the current heartbeat timeout in milliseconds.
GevTimestampTickFrequency	O	IInteger	R	Hz	E	Indicates the number of timestamp ticks in 1 second (frequency in Hz).
GevTimestampControlLatch	O	ICommand	W	-	E	Latches the current timestamp counter into GevTimestampValue.
GevTimestampControlReset	O	ICommand	W	-	E	Resets the timestamp counter to 0.
GevTimestampValue	O	IInteger	R		E	Returns the latched 64-bit value of the timestamp counter.

GevDiscoveryAckDelay	O	IInteger	R/(W)	ms	E	Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.
GevGVCPExtendedStatusCodes	O	IBoolean	R/W	-	G	Enables the generation of extended status codes.
GevGVCPPendingAck	O	IBoolean	R/W	-	G	Enables the generation of PENDING_ACK.
GevGVCPHeartbeatDisable	O	IBoolean	R/W	-	E	Disables the GVCP heartbeat.
GevGVCPPendingTimeout	O	IInteger	R	-	G	Indicates the longest GVCP command execution time before a device returns a PENDING_ACK.
GevPrimaryApplicationSwitchoverKey	O	IInteger	W-O	-	G	Controls the key to use to authenticate primary application switchover requests.
GevCCP	O	IEnumeration	R/W	-	G	Controls the device access privilege of an application.
GevPrimaryApplicationSocket	O	IInteger	R	-	G	Returns the UDP source port of the primary application.
GevPrimaryApplicationIPAddress	O	IInteger	R	-	G	Returns the address of the primary application.
GevMCPHostPort	O	IInteger	R/W	-	G	Controls the port to which the device must send messages.
GevMCDA	O	IInteger	R/W	-	G	Controls the destination IP address for the message channel.
GevMCTT	O	IInteger	R/W	ms	G	Provides the transmission timeout value in milliseconds.
GevMCRC	O	IInteger	R/W	-	G	Controls the number of retransmissions allowed when a message channel message times out.
GevMCSP	O	IInteger	R	-	G	This feature indicates the source port for the message channel.
GevStreamChannelSelector	O	IInteger	R/W	-	E	Selects the stream channel to control.
GevSCCFGUnconditionalStreaming[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).
GevSCCFGExtendedChunkData[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Enables cameras to use the extended chunk data payload type for this stream channel.
GevSCPDirection[GevStreamChannelSelector]	O	IEnumeration	R	-	G	Reports the direction of the stream channel.
GevSCPInterfaceIndex[GevStreamChan	O	IInteger	R/W	-	G	Index of network interface to use.

nelSelector]						
GevSCPHostPort[GevStreamChannelSelector]	O	IInteger	R/W	-	G	Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.
GevSCPSFireTestPacket[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Sends a test packet.
GevSCPSDoNotFragment[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	The state of this feature is copied into the "do not fragment" bit of IP header of each stream packet.
GevSCPSBigEndian[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Endianess of multi-byte pixel data for this stream.
GevSCPSPacketSize[GevStreamChannelSelector]	R	IInteger	R/(W)	B	E	Specifies the stream packet size, in bytes, to send on the selected channel for a GVSP transmitter or specifies the maximum packet size supported by a GVSP receiver.
GevSCPD[GevStreamChannelSelector]	R	IInteger	R/W		E	Controls the delay (in timestamp counter unit) to insert between each packet for this stream channel.
GevSCDA[GevStreamChannelSelector]	O	IInteger	R/W	-	G	Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.
GevSCSP[GevStreamChannelSelector]	O	IInteger	R	-	G	Indicates the source port of the stream channel.
GevManifestEntrySelector	O	IInteger	R/W	-	I	This feature is deprecated.
GevManifestXMLMajorVersion[GevManifestEntrySelector]	O	IInteger	R	-	I	This feature is deprecated.
GevManifestXMLMinorVersion[GevManifestEntrySelector]	O	IInteger	R	-	I	This feature is deprecated.
GevManifestXMLSubMinorVersion[GevManifestEntrySelector]	O	IInteger	R	-	I	This feature is deprecated.
GevManifestSchemaMajorVersion[GevManifestEntrySelector]	O	IInteger	R	-	I	This feature is deprecated.
GevManifestSchemaMinorVersion[Gev	O	IInteger	R	-	I	This feature is deprecated.

ManifestEntrySelector]						
GevManifestPrimaryURL[GevManifestEntrySelector]	O	IString	R	-	I	This feature is deprecated.
GevManifestSecondaryURL[GevManifestEntrySelector]	O	IString	R	-	I	This feature is deprecated.
ClConfiguration	R	IEnumeration	R/(W)	-	B	This Camera Link specific feature describes the configuration used by the camera.
ClTimeSlotsCount	O	IEnumeration	R/(W)	-	O	This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.
DeviceTapGeometry	R	IEnumeration	R/(W)	-	G	This Camera Link specific device tap geometry feature describes the geometrical properties characterizing the taps of a Camera Link camera as seen from the frame grabber or acquisition card.

2.11 User Set Control

Contains the features related to the User Set Control to save and load the user device settings.

Table 2-11: User Set Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
UserSetControl	R	ICategory	R	-	B	Category that contains the User Set control features.
UserSetSelector	R	IEnumeration	R/W	-	B	Selects the feature User Set to load, save or configure.
UserSetLoad[UserSetSelector]	R	ICommand	(R)/W	-	B	Loads the User Set specified by UserSetSelector to the device and makes it active.
UserSetSave[UserSetSelector]	R	ICommand	(R)/W	-	B	Save the User Set specified by UserSetSelector to the non-volatile memory of the device.
UserSetDefaultSelector	O	IEnumeration	R/W	-	B	Selects the feature User Set to load and make active when the device is reset.

2.12 Chunk Data Control

Contains the features related to the Chunk Data Control.

Table 2-12: Chunk Data Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ChunkDataControl	R	ICategory	R	-	E	Category that contains the Chunk Data control features.
ChunkModeActive	R	IBoolean	R/W	-	E	Activates the inclusion of Chunk data in the payload of the image.
ChunkSelector	R	IEnumeration	R/W	-	E	Selects which Chunk to enable or control.
ChunkEnable[ChunkSelector]	R	IBoolean	R/W	-	E	Enables the inclusion of the selected Chunk data in the payload of the image.
ChunkImage	R	IRegister	R	-	G	Returns the entire image data included in the payload.
ChunkOffsetX	R	IInteger	R	-	E	Returns the OffsetX of the image included in the payload.
ChunkOffsetY	R	IInteger	R	-	E	Returns the OffsetY of the image included in the payload.
ChunkWidth	R	IInteger	R	-	E	Returns the Width of the image included in the payload.
ChunkHeight	R	IInteger	R	-	E	Returns the Height of the image included in the payload.
ChunkPixelFormat	R	IEnumeration	R	-	E	Returns the PixelFormat of the image included in the payload.
ChunkPixelDynamicRangeMin	R	IInteger	R	-	E	Returns the minimum value of dynamic range of the image included in the payload.
ChunkPixelDynamicRangeMax	R	IInteger	R	-	E	Returns the maximum value of dynamic range of the image included in the payload.
ChunkDynamicRangeMin	R	IInteger	R	-	I	This feature is deprecated.
ChunkDynamicRangeMax	R	IInteger	R	-	I	This feature is deprecated.
ChunkTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event.
ChunkLineStatusAll	R	IInteger	R	-	E	Returns the status of all the I/O lines at the time of the FrameStart internal event.

ChunkCounterSelector	R	IEnumeration	R/W	-	E	Selects which counter to retrieve data from.
ChunkCounterValue[ChunkCounterSelector]	R	IInteger	R	-	E	Returns the value of the selected Chunk counter at the time of the FrameStart event.
ChunkCounter[ChunkCounterSelector]	R	IInteger	R	-	I	This feature is deprecated.
ChunkTimerSelector	R	IEnumeration	R/W	-	E	Selects which Timer to retrieve data from.
ChunkTimerValue[ChunkTimerSelector]	R	IFloat	R	us	E	Returns the value of the selected Timer at the time of the FrameStart internal event.
ChunkTimer[ChunkTimerSelector]	R	IFloat	R	us	I	This feature is deprecated.
ChunkExposureTime	R	IFloat	R	us	E	Returns the exposure time used to capture the image.
ChunkGainSelector	R	IEnumeration	R/W	-	E	Selects which Gain to retrieve data from.
ChunkGain[ChunkGainSelector]	R	IFloat	R	-	E	Returns the gain used to capture the image.
ChunkBlackLevelSelector	R	IEnumeration	R/W	-	E	Selects which Black Level to retrieve data from.
ChunkBlackLevel[ChunkBlackLevelSelector]	R	IFloat	R	-	E	Returns the black level used to capture the image included in the payload.
ChunkLinePitch	R	IInteger	R	B	E	Returns the LinePitch of the image included in the payload.
ChunkFrameID	R	IInteger	R	-	E	Returns the unique Identifier of the frame (or image) included in the payload.

2.13 File Access Control

Contains the features related to the File that provides all the sevices necessary for generic file access of a device.

Table 2-13: File Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
FileAccessControl	R	ICategory	R	-	G	Category that contains the File Access control features.
FileSelector	R	IEnumeration	R/(W)	-	G	Selects the target file in the device.

FileOperationSelector[FileSelector]	R	IEnumeration	R/W	-	G	Selects the target operation for the selected file in the device.
FileOperationExecute[FileSelector][FileOperationSelector]	R	ICommand	(R)/W	-	G	Executes the operation selected by FileOperationSelector on the selected file.
FileOpenMode[FileSelector]	R	IEnumeration	R/(W)	-	G	Selects the access mode in which a file is opened in the device.
FileAccessBuffer	R	IRegister	R/(W)	-	G	Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
FileAccessOffset[FileSelector][FileOperationSelector]	R	IInteger	R/(W)	B	G	Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.
FileAccessLength[FileSelector][FileOperationSelector]	R	IInteger	R/W	B	G	Controls the Length of the mapping between the device file storage and the FileAccessBuffer.
FileOperationStatus[FileSelector][FileOperationSelector]	R	IEnumeration	R	-	G	Represents the file operation execution status.
FileOperationResult[FileSelector][FileOperationSelector]	R	IInteger	R	-	G	Represents the file operation result.
FileSize[FileSelector]	R	IInteger	R	B	G	Represents the size of the selected file in bytes.

2.14 Color Transformation Control

Contains the features related to the control of the color transformation.

Table 2-14: Color Transformation summary

Name	Level	Interface	Access	Unit	Visibility	Description
ColorTransformationControl	R	ICategory	R	-	E	Category that contains the Color Transformation control features.
ColorTransformationSelector	O	IEnumeration	R/W	-	E	Selects which Color Transformation module is controlled by the various Color Transformation features.
ColorTransformationEnable[ColorTransformationSelector]	O	IBoolean	R/W	-	E	Activates the selected Color Transformation module.
ColorTransformationValueSelector[Col	O	IEnumeration	R/W	-	E	Selects the Gain factor or Offset of the Transformation matrix to access

orTransformationSelector]						in the selected Color Transformation module.
ColorTransformationValue[ColorTransformationSelector][ColorTransformationValueSelector]	O	IFloat	R/W	-	E	Represents the value of the selected Gain factor or Offset inside the Transformation matrix.

2.15 Action Control

Contains the features related to the control of the Action command mechanism.

Table 2-15: Action Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ActionControl	R	ICategory	R	-	G	Category that contains the Action control features.
ActionDeviceKey	O	IInteger	W-O	-	G	Provides the device key that allows the device to check the validity of action commands.
ActionSelector	O	IInteger	R/W	-	G	Selects to which Action Signal further Action settings apply.
ActionGroupMask[ActionSelector]	O	IInteger	R/W	-	G	Provides the mask that the device will use to validate the action on reception of the action protocol message.
ActionGroupKey[ActionSelector]	O	IInteger	R/W	-	G	Provides the key that the device will use to validate the action on reception of the action protocol message.

3 Device Control

Device control features provides general information and control for the device (camera) and its sensor. This is mainly used to identify the device during the enumeration process and to obtain information about the sensor resolution. Other information and controls pertaining to the general state of the device are also included in this category.

3.1 DeviceControl

Name	DeviceControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	-

Category for Device information and control.

3.2 DeviceVendorName

Name	DeviceVendorName
Level	Recommended
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

Name of the manufacturer of the device.

For GigE Vision bootstrap registers, this string has a maximum length of 32 bytes (including the NULL-terminating character).

3.3 DeviceModelName

Name	DeviceModelName
Level	Recommended

Interface	IString
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

Model of the device.

For GigE Vision bootstrap registers, this string has a maximum length of 32 bytes (including the NULL-terminating character).

3.4 DeviceManufacturerInfo

Name	DeviceManufacturerInfo
Level	Recommended
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

Manufacturer information about the device.

For GigE Vision bootstrap registers, this string has a maximum length of 48 bytes (including the NULL-terminating character).

3.5 DeviceVersion

Name	DeviceVersion
Level	Recommended
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

Version of the device.

For GigE Vision bootstrap registers, this string has a maximum length of 32 bytes (including the NULL-terminating character).

3.6 DeviceFirmwareVersion

Name	DeviceFirmwareVersion
Level	Recommended
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

Version of the firmware in the device.

This information is not provided by the standard GigE Vision bootstrap registers. It must be part of device-specific registers.

3.7 DeviceSFNCVersionMajor

Name	DeviceSFNCVersionMajor
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	>0

Major Version of the Standard Feature Naming Convention that was used to create the device`s XML.

3.8 DeviceSFNCVersionMinor

Name	DeviceSFNCVersionMinor
Level	Recommended
Interface	IInteger

Access	Read
Unit	-
Recommended Visibility	Beginner
Values	>=0

Minor Version of the Standard Feature Naming Convention that was used to create the device`s XML

3.9 DeviceSFNCVersionSubMinor

Name	DeviceSFNCVersionSubMinor
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	>=0

Sub Minor Version of Standard Feature Naming Convention that was used to create the device`s XML.

3.10 DeviceManifestEntrySelector

Name	DeviceManifestEntrySelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥1

Selects the manifest entry to reference.

3.11 DeviceManifestXMLMajorVersion

Name	DeviceManifestXMLMajorVersion[DeviceManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the major version number of the XML file of the selected manifest entry.

3.12 DeviceManifestXMLMinorVersion

Name	DeviceManifestXMLMinorVersion[DeviceManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the minor version number of the XML file of the selected manifest entry.

3.13 DeviceManifestXMLSubMinorVersion

Name	DeviceManifestXMLSubMinorVersion[DeviceManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the subminor version number of the XML file of the selected manifest entry.

3.14 DeviceManifestSchemaMajorVersion

Name	DeviceManifestSchemaMajorVersion[DeviceManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the major version number of the schema file of the selected manifest entry.

3.15 DeviceManifestSchemaMinorVersion

Name	DeviceManifestSchemaMinorVersion[DeviceManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the minor version number of the schema file of the selected manifest entry.

3.16 DeviceManifestPrimaryURL

Name	DeviceManifestPrimaryURL[DeviceManifestEntrySelector]
Level	Optional
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

Indicates the first URL to the XML device description file of the selected manifest entry.

3.17 DeviceManifestSecondaryURL

Name	DeviceManifestSecondaryURL[DeviceManifestEntrySelector]
Level	Optional
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

Indicates the second URL to the XML device description file of the selected manifest entry.

3.18 DeviceID

Name	DeviceID
Level	Recommended
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Expert
Values	Any NULL-terminated string

Device Identifier (serial number).

GigE Vision bootstrap registers provide a string with up to 16 bytes to store the serial number of the camera.

3.19 DeviceUserID

Name	DeviceUserID
Level	Optional
Interface	IString
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

User-programmable Device Identifier.

When this feature is present, it must be writable. The recommended factory default value is an empty string.

3.20 DeviceReset

Name	DeviceReset
Level	Recommended
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Guru
Values	-

Resets the device to its power up state.

3.21 DeviceRegistersStreamingStart

Name	DeviceRegistersStreamingStart
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Guru

Values	-

Prepare the device for registers streaming without checking for consistency.

If the camera implements this feature, GenApi guarantees using it to announce register streaming.

If the feature is present, but currently not writable (locked), the application must not start register streaming and must avoid switching the access mode and range verification off until the feature becomes writable again.

3.22 DeviceRegistersStreamingEnd

Name	DeviceRegistersStreamingEnd
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Guru
Values	-

Announce the end of registers streaming. This will do a register set validation for consistency and activate it. This will also update the **DeviceRegistersValid** flag.

3.23 DeviceRegistersCheck

Name	DeviceRegistersCheck
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Expert
Values	-

Perform the validation of the current register set for consistency. This will update the **DeviceRegistersValid** flag.

3.24 DeviceRegistersValid

Name	DeviceRegistersValid
Level	Recommended
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Expert
Values	True False

Returns if the current register set is valid and consistent.

3.25 DeviceMaxThroughput

Name	DeviceMaxThroughput
Level	Optional
Interface	IInteger
Access	Read
Unit	Bps
Recommended Visibility	Expert
Values	>0

Maximum bandwidth of the data that can be streamed out of the device. This can be used to estimate if the network connection can sustain transfer of free-running images from the camera at its maximum speed.

3.26 DeviceTemperatureSelector

Name	DeviceTemperatureSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert

Values	Sensor Mainboard Device-specific

Selects the location within the device, where the temperature will be measured.

3.27 DeviceTemperature

Name	DeviceTemperature[DeviceTemperatureSelector]
Level	Optional
Interface	IFloat
Access	Read
Unit	C
Recommended Visibility	Expert
Values	Device-specific

Device temperature in degrees Celsius (C). It is measured at the location selected by DeviceTemperatureSelector.

3.28 DeviceClockSelector

Name	DeviceClockSelector
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	Sensor SensorDigitization CameraLink Device-specific

Selects the clock frequency to access from the device.

DeviceClockSelector can take one of the following values:

- Sensor: Clock frequency of the image sensor of the camera.

- SensorDigitization: Clock frequency of the camera A/D conversion stage.
- CameraLink: Speed of the Camera Link clock.

3.29 DeviceClockFrequency

Name	DeviceClockFrequency[DeviceClockSelector]
Level	Optional
Interface	IFloat
Access	Read/(Write)
Unit	Hz
Recommended Visibility	Expert
Values	>=0

Returns the frequency in Hertz of the selected Clock.

3.30 DeviceSerialPortSelector

Name	DeviceSerialPortSelector
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	CameraLink Device-specific

Selects which device serial port to control.

3.31 DeviceSerialPortBaudRate

Name	DeviceSerialPortBaudRate[SerialPortSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)

Unit	-
Recommended Visibility	Expert
Values	Baud9600 Baud19200 Baud38400 Baud57600 Baud115200 Baud230400 Baud460800 Baud921600 …

This feature controls the baud rate used by the selected device's serial port. Typical values listed should be used whenever possible. Arbitrary values can also be used by defining new enumeration entries.

3.32 DeviceScanType

Name	DeviceScanType
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	Areascan Linescan

Scan type of the sensor of the device.

Typically, this feature is not writable. But some cameras might allow switching between linescan and areascan.

DeviceScanType can take any of the following values:

- **Areascan**: 2D sensor
- **Linescan**: 1D sensor

4 Image Format Control

This section describes how to influence and determine the image size and format. It also provides the necessary information to acquire and to display the image data. It assumes that the camera expels a single rectangular image.

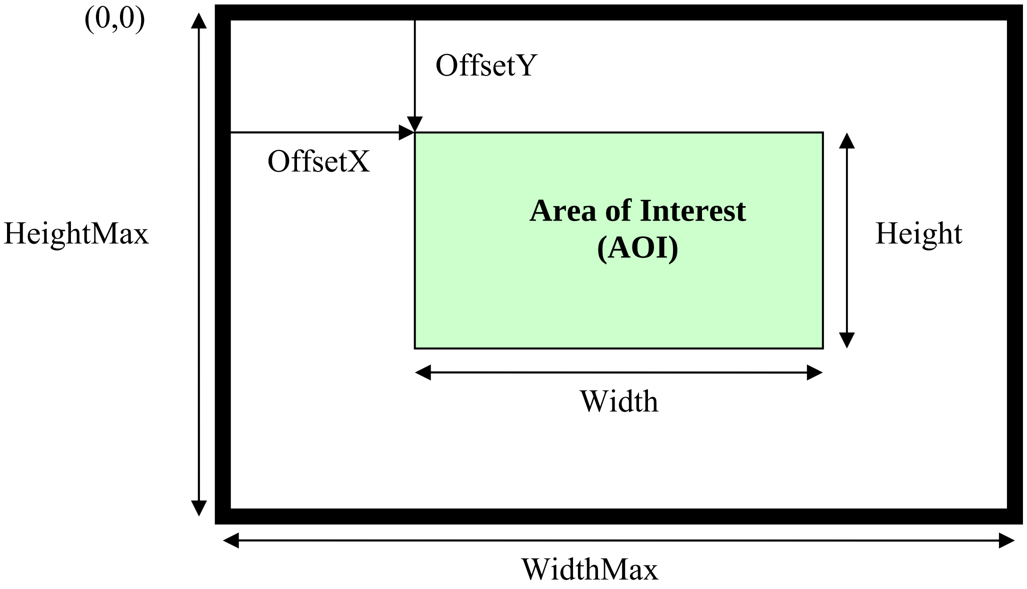

Figure 4-1: Image size and defining an area of interest (AOI)

The sensor provides **SensorWidth** time **SensorHeight** pixels.

Using **BinningHorizontal** and/or **BinningVertical** or **DecimationHorizontal** and/or **DecimationVertical** the image is shrunk to**WidthMax** time **HeightMax** pixels.

In addition the features **ReverseX** and **ReverseY** can be used to flip the image respectively along the X-axis or Y-axis. The flipping is done before the AOI is applied.

Within the shrunk image the user can set an area of interest (AOI) using the features **OffsetX**, **OffsetY**, **Width**, and **Height**. The resulting image expelled by the camera has **Width** time **Height** pixels. **OffsetX** and **OffsetY** are given with respect to the upper left corner of the image which has the coordinate (0, 0), see
Figure 4-1.

All measures are given in the unit [pixel]. As a result the values should not change if the **PixelFormat** changes. For monochrome cameras each pixel corresponds to one gray value. For color camera in raw mode (Bayer pattern, etc.) each pixel corresponds to one pixel in the color mask. For color cameras in RGB mode each pixel corresponds to one RGB triplet. For color cameras in YUV mode each pixel corresponds to one Y value with the associated color information.

The feature **Height** describes the height of the image in lines. The pixels within a line are contiguous. The lines however may be not contiguous, e.g. in order to yield a DWORD alignment. **LinePitch** gives the number of bytes separating the starting pixels of two consecutive lines.

Each pixel in the image has a format defined by **PixelFormat**. For details see GigEVision specification **PixelFormat** (section 25.2 of GigE Vision Specification).

Because the **PixelFormat** feature contains a mix of informations specified by the user and informations provided by the camera, it is suitable for describing the whole pixel settings but might be less practical when individual setting must be set or inquired. Therefore a second set of features exists composed of the individual components of **PixelFormat.** Those features are **PixelCoding**, **PixelSize**, **PixelColorFilter**, **PixelDynamicRangeMin** and **PixelDynamicRangeMax.**

Even if the **PixelFormat** might allow for, e.g. 16 bits per pixel, the real image data might provide only a certain range of value (e.g. 12 bits per pixel because the camera is equipped with a 12 bit analog to digital converter only). In that case, **PixelDynamicRangeMin** and **PixelDynamicRangeMax** specify the lower and upper limits of the pixel values in the image. In general, **PixelDynamicRangeMin** should be zero and **PixelDynamicRangeMax** should be a power of two ($\left[0, 2^{DataDepth} - 1\right]$). There should be no missing codes in the range.

4.1 ImageFormatControl

Name	ImageFormatControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	-

Category for Image Format Control features.

4.2 SensorWidth

Name	SensorWidth
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Effective width of the sensor in pixels.

Its value must be greater than 0.

4.3 SensorHeight

Name	SensorHeight
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Effective height of the sensor in pixels.

Its value must be greater than 0. For linescan sensor, this value is 1.

4.4 SensorTaps

Name	SensorTaps
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	One Two Three Four Eight Ten … Device-specific

Number of taps of the camera sensor.

4.5 SensorDigitizationTaps

Name	SensorDigitizationTaps
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	One Two Three Four Eight Ten … Device-specific

Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

4.6 WidthMax

Name	WidthMax
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Maximum width (in pixels) of the image. The dimension is calculated after horizontal binning, decimation or any other function changing the horizontal dimension of the image.

4.7 HeightMax

Name	HeightMax
Level	Recommended
Interface	IInteger

Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Maximum height (in pixels) of the image. This dimension is calculated after vertical binning, decimation or any other function changing the vertical dimension of the image.

4.8 Width

Name	Width
Level	Mandatory
Interface	IInteger
Access	Read/(Write)
Unit	-
Recommended Visibility	Beginner
Values	>0

Width of the Image provided by the device (in pixels).

4.9 Height

Name	Height
Level	Mandatory
Interface	IInteger
Access	Read/(Write)
Unit	-
Recommended Visibility	Beginner
Values	>0

Height of the image provided by the device (in pixels).

4.10 OffsetX

Name	OffsetX
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥0

Horizontal offset from the origin to the area of interest (in pixels).

4.11 OffsetY

Name	OffsetY
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥0

Vertical offset from the origin to the area of interest (in pixels).

4.12 LinePitch

Name	LinePitch
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	B
Recommended Visibility	Expert
Values	≥0

Total number of bytes between 2 successive lines. This feature is used to facilitate alignment of image data.

This might be useful if the system has specific limitations, such as having the lines aligned on 32-bit boundaries.

4.13 BinningHorizontal

Name	BinningHorizontal
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	>0

Number of horizontal photo-sensitive cells to combine together. This increases the intensity (or signal to noise ratio) of the pixels and reduces the horizontal resolution (width) of the image.

A value of 1 indicates that no horizontal binning is performed by the camera.

4.14 BinningVertical

Name	BinningVertical
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	>0

Number of vertical photo-sensitive cells to combine together. This increases the intensity (or signal to noise ratio) of the pixels and reduces the vertical resolution (height) of the image.

A value of 1 indicates that no vertical binning is performed by the camera.

4.15 DecimationHorizontal

Name	DecimationHorizontal
Level	Optional

Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

Horizontal sub-sampling of the image. This reduces the horizontal resolution (width) of the image by the specified horizontal decimation factor.

This might be done by pixel dropping or by first applying a horizontal low-pass filter before pixel dropping.

A value of 1 indicates that the camera performs no horizontal decimation.

4.16 DecimationVertical

Name	DecimationVertical
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

Vertical sub-sampling of the image. This has the net effect of reducing the vertical resolution (height) of the image by the specified vertical decimation factor.

This might be implemented by pixel dropping or by first applying a vertical low-pass filter before pixel dropping.

A value of 1 indicates that the camera performs no vertical decimation.

4.17 ReverseX

Name	ReverseX
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended	Expert

Visibility	
Values	True False

Flip horizontally the image sent by the device. The AOI is applied after the flipping.

4.18 ReverseY

Name	ReverseY
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Flip vertically the image sent by the device. The AOI is applied after the flipping.

4.19 PixelFormat

Name	PixelFormat
Level	Mandatory
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Beginner
Values	Mono8 Mono8Signed Mono10 Mono10Packed Mono12 Mono12Packed Mono14 Mono16 BayerGR8 BayerRG8 BayerGB8 BayerBG8

	BayerGR10 BayerRG10 BayerGB10 BayerBG10 BayerGR12 BayerRG12 BayerGB12 BayerBG12 BayerGR10Packed BayerRG10Packed BayerGB10Packed BayerBG10Packed BayerGR12Packed BayerRG12Packed BayerGB12Packed BayerBG12Packed BayerGR16 BayerRG16 BayerGB16 BayerBG16 RGB8Packed BGR8Packed RGBA8Packed BGRA8Packed RGB10Packed BGR10Packed RGB12Packed BGR12Packed RGB16Packed BGR16Packed RGB10V1Packed BGR10V1Packed RGB10V2Packed BGR10V2Packed RGB12V1Packed RGB565Packed BGR565Packed YUV411Packed YUV422Packed YUV444Packed YUYVPacked RGB8Planar RGB10Planar RGB12Planar RGB16Planar Device-specific

Format of the pixel provided by the device. It represents all the informations provided by **PixelCoding, PixelSize**, **PixelColorFilter** but combined in one single value.

The values of the enumeration and the pixel formatting correspond to the GigE Vision specification (when applicable).

4.20 PixelCoding

Name	PixelCoding
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	Mono MonoSigned MonoPacked RGBPacked BGRPacked RGBAPacked BGRAPacked RGBPlanar YUV411Packed YUV422Packed YUV444Packed YUYVPacked Raw RawPacked

Coding of the pixels in the image. Raw gives the data in the native format of the sensor.

Raw is mainly used for Bayer sensor. This value must always be coherent with the **PixelFormat** feature.

4.21 PixelSize

Name	PixelSize
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-

Recommended Visibility	Expert
Values	Bpp8 Bpp10 Bpp12 Bpp14 Bpp16 Bpp24 Bpp30 Bpp32 Bpp36 Bpp48 Bpp64

Total size in bits of a pixel of the image.

This value must always be coherent with the **PixelFormat** feature.

4.22 PixelColorFilter

Name	PixelColorFilter
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	None BayerRG BayerGB BayerGR BayerBG

Type of color filter that is applied to the image.

This value must always be coherent with the **PixelFormat** feature.

4.23 PixelDynamicRangeMin

Name	PixelDynamicRangeMin
Level	Optional
Interface	IInteger
Access	Read/Write

Unit	-
Recommended Visibility	Expert
Values	Device-specific

Minimum value that can be returned during the digitization process. This corresponds to the darkest value of the camera. For color camera, this returns the smallest value that each color component can take.

4.24 PixelDynamicRangeMax

Name	PixelDynamicRangeMax
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Maximum value that will be returned during the digitization process. This corresponds to the brightest value of the camera. For color camera, this returns the biggest value that each color component can take.

4.25 TestImageSelector

Name	TestImageSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Black White GreyHorizontalRamp GreyVerticalRamp GreyHorizontalRampMoving GreyVerticalRampMoving

	HorzontalLineMoving VerticalLineMoving ColorBar FrameCounter Device-specific

Selects the type of test image that is sent by the camera.

TestImageSelector can take any of the following values:

- **Off**: Image is coming from the sensor.
- **Black**: Image is filled with the darkest possible image.
- **White**: Image is filled with the brightest possible image.
- **GreyHorizontalRamp**: Image is filled horizontally with an image that goes from the darkest possible value to the brightest.
- **GreyVerticalRamp**: Image is filled vertically with an image that goes from the darkest possible value to the brightest.
- **GreyHorizontalRampMoving**: Image is filled horizontally with an image that goes from the darkest possible value to the brightest and that moves horizontally from left to right at each frame.
- **GreyVerticalRampMoving**: Image is filled vertically with an image that goes from the darkest possible value to the brightest and that moves verticaly from top to bottom at each frame.
- **HorizontalLineMoving**: A moving horizontal line is superimposed on the live image.
- **VerticalLineMoving**: A moving vertical line is superimposed on the live image.
- **ColorBar**: Image is filled with stripes of color including White, Black, Red, Green, Blue, Cyan, Magenta and Yellow.
- **FrameCounter**: A frame counter is superimposed on the live image.

Other values are device-specific and represent particular test images digitally generated by the camera.

5 Acquisition Control

The Acquisition Control section describes all features related to image acquisition, including the trigger and exposure control. It describes the basic model for acquisition and the typical behavior of the device.

An **Acquisition** is defined as the capture of a sequence of one or many **Frame**(s) (see Figure 5-1**).** The transfer of the frame(s) of an **Acquisition,** starts with the beginning of the transfer of the first frame and ends with completion of the transfer of the last one.

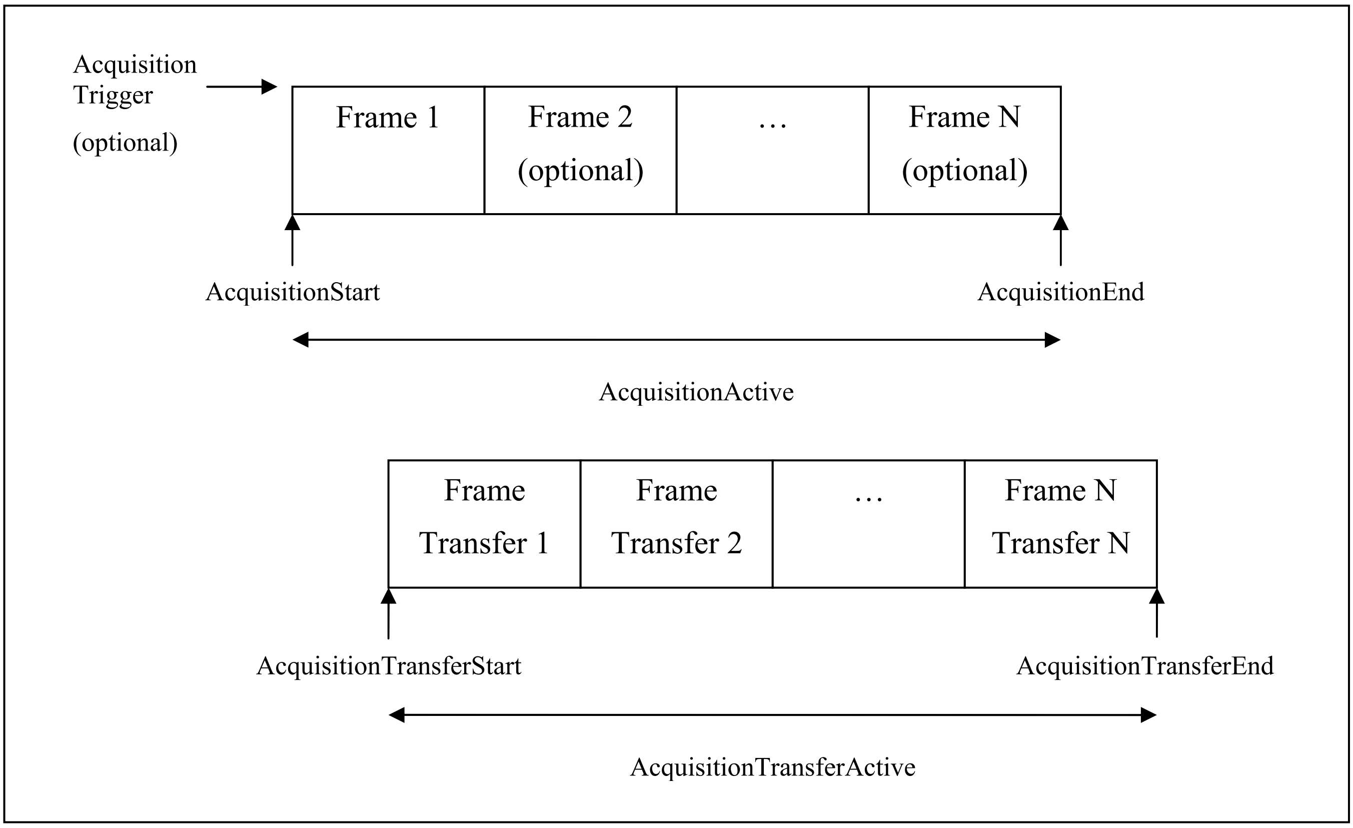

Figure 5-1: Acquisition signals definitions

A **Frame** is defined as the capture of **Width** pixels x **Height** lines. A **Frame** starts with an optional **Exposure** period and ends with the completion of the sensor read out. Generally, a transfer period will start during the sensor read out and will finish sometime after it but it is not considered as part of the Frame (see Figure 5-2).

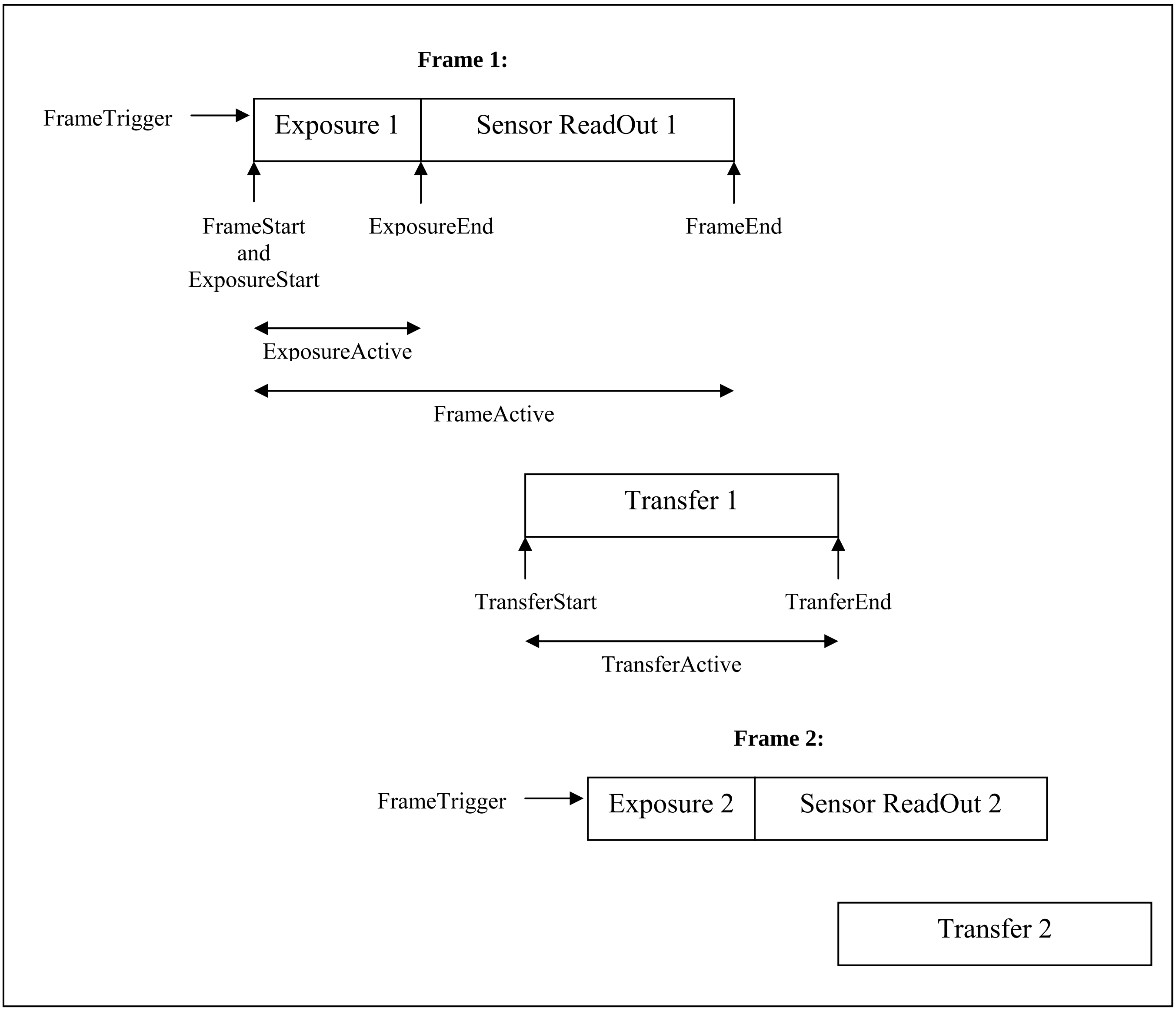

Figure 5-2: Frame signals definitions

For Line Scan acquisition, the definition of **Frame** stays the same but the exposure and read out are done for each line of the virtual Frame (see Figure 5-3).

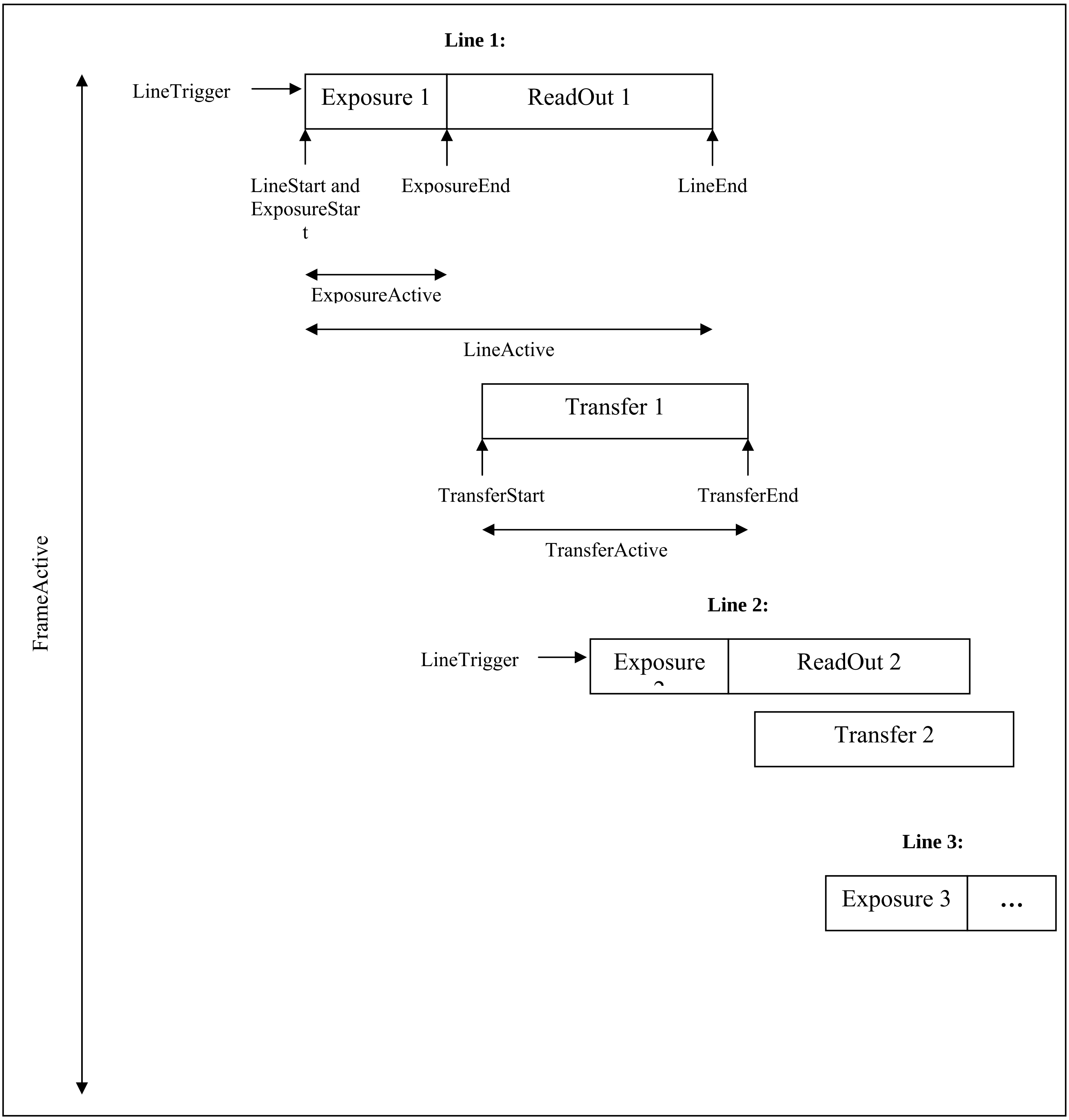

Figure 5-3: Frame signals definitions in Line Scan mode

Acquisition Control features:

The **AcquisitionMode** controls the mode of acquisition for the device. This mainly affects the number of frames captured in the Acquisition (**SingleFrame, MultiFrame, Continuous**).

The optional **AcquisitionArm** command is used to verify and freeze all parameters relevant for the image data capture. It prepares the device for the **AcquistionStart**.

The **AcquisitionStart** command is used to start the Acquisition.

The **AcquisitionStop** command will stop the Acquisition at the end of the current Frame**.** It can be used in any acquisition mode and if the camera is waiting for a trigger, the pending Frame will be cancelled.

The **AcquisitionAbort** command can be used to abort an Acquisition at any time. This will end the capture immediately without completing the current Frame.

AcquisitionFrameCount controls the number of frames that will be captured when **AcquisitionMode** is **MultiFrame**.

AcquisitionFrameRate controls the rate at which the Frames are captured when **TriggerMode** is **Off**.

AcquisitionLineRate controls the rate at which the Lines in each Frame are captured. This is generally useful for line scan cameras.

AcquisitionStatusSelector and **AcquisitionStatus** can be used to read the status of the internal acquisition signals. The standard acquisition signals Status are: **AcquisitionTriggerWait, AcquisitionActive, AcquisitionTransfer, FrameTriggerWait, FrameActive, FrameTransfer, ExposureActive** (see Figure 5-1 and Figure 5-2),

See the Acquisition and Trigger Examples section at the end for more complete use cases of the acquisition and trigger features in conjunction with other related sections such as I/O and analog controls.

5.1 AcquisitionControl

Name	AcquisitionControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	-

Category for the acquisition and trigger control features.

5.2 AcquisitionMode

Name	AcquisitionMode
Level	Mandatory
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	SingleFrame MultiFrame Continuous

Sets the acquisition mode of the device. It defines mainly the number of frames to capture during an acquisition and the way the acquisition stops.

AcquisitionMode can take any of the following values:

- **SingleFrame**: One frame is captured.
- **MultiFrame**: The number of frames specified by **AcquisitionFrameCount** is captured.
- **Continuous**: Frames are captured continuously until stopped with the **AcquisitionStop** command.

5.3 AcquisitionStart

Name	AcquisitionStart
Level	Mandatory
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Beginner
Values	-

Starts the Acquisition of the device. The number of frames captured is specified by **AcquisitionMode.**

The Acquisition might be conditioned by the various trigger (see **Trigger**… features). An AcquisitionStart command must be sent to the device before the acquisition related triggers become effective.

Note that unless the **AcquisitionArm** was executed since the last feature change, the **AcquisitionStart** command must validate all the current features for consistency before starting the

Acquisition. This validation will not be repeated for the subsequent acquisitions unless a feature is changed in the device.

If the AcquisitionStart feature is currently not writable (locked), the application must not start the acquisition and must avoid to use the feature until the feature becomes writable again.

5.4 AcquisitionStop

Name	AcquisitionStop
Level	Mandatory
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Beginner
Values	-

Stops the Acquisition of the device at the end of the current Frame**.** It is mainly used when **AcquisitionMode** is **Continuous** but can be used in any acquisition mode.

If the camera is waiting for a trigger, the pending Frame will be cancelled. If no Acquisition is in progress, the command is ignored.

5.5 AcquisitionAbort

Name	AcquisitionAbort
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Expert
Values	-

Aborts the acquisition immediately. This will end the capture without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored.

5.6 AcquisitionArm

Name	AcquisitionArm
Level	Optional
Interface	ICommand

Access	(Read)/Write
Unit	-
Recommended Visibility	Expert
Values	-

Arms the device before an **AcquisitionStart** command. This optional command validates all the current features for consistency and prepares the device for a fast start of the Acquisition.

If not used explicitly, this command will be automatically executed at the first **AcquisitionStart** but will not be repeated for the subsequent ones unless a feature is changed in the device.

5.7 AcquisitionFrameCount

Name	AcquisitionFrameCount
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥1

Number of frames to acquire in MultiFrame Acquisition mode.

The minimum allowable value is 1.

5.8 AcquisitionFrameRate

Name	AcquisitionFrameRate
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	Hz
Recommended Visibility	Beginner
Values	Device-specific

Controls the acquisition rate (in Hertz) at which the frames are captured.

TriggerMode must be **Off** for the Frame trigger.

5.9 AcquisitionFrameRateAbs (Deprecated)

Name	AcquisitionFrameRateAbs
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	Hz
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. It controls the rate (in Hertz) at which the Frames are captured when **TriggerMode** is **Off** for the Frame trigger.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

5.10 AcquisitionFrameRateRaw (Deprecated)

Name	AcquisitionFrameRateRaw
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. It controls the rate (in device specific unit) at which the Frames are captured when **TriggerMode** is **Off** for the Frame trigger.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

5.11 AcquisitionLineRate

Name	AcquisitionLineRate
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	Hz

Recommended Visibility	Beginner
Values	Device-specific

Controls the rate (in Hertz) at which the Lines in a Frame are captured.

TriggerMode must be **Off** for the Line trigger.

This is generally useful for line scan camera only.

5.12 AcquisitionLineRateAbs (Deprecated)

Name	AcquisitionLineRateAbs
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	Hz
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. It controls the rate (in Hertz) at which the Lines in a Frame are captured when **TriggerMode** is **Off** for the Line trigger.

This is generally useful for line scan camera only.

This feature can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

5.13 AcquisitionLineRateRaw (Deprecated)

Name	AcquisitionLineRateRaw
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. It controls the rate (in device specific unit) at which the Lines in a Frame are captured when **TriggerMode** is **Off** for the Line trigger.

This is generally useful for line scan camera only.

This feature can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

5.14 AcquisitionStatusSelector

Name	AcquisitionStatusSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	AcquisitionTriggerWait AcquisitionActive AcquisitionTransfer FrameTriggerWait FrameActive FrameTransfer ExposureActive

Selects the internal acquisition signal to read using AcquisitionStatus**.**

AcquisitionStatusSelector can take any of the following values (see Figure 5-1 and Figure 5-2):

- **AcquisitionTriggerWait**: Device is currently waiting for a trigger for the capture of one or many frames.
- **AcquisitionActive**: Device is currently doing an acquisition of one or many frames.
- **AcquisitionTransfer**: Device is currently transferring an acquisition of one or many frames.
- **FrameTriggerWait**: Device is currently waiting for a Frame trigger.
- **FrameActive**: Device is currently doing the capture of a frame.
- **FrameTransfer**: Device is currently transferring a frame.
- **ExposureActive**: Device is doing the Exposure of a frame.

5.15 AcquisitionStatus

Name	AcquisitionStatus[AcquisitionStatusSelector]
Level	Recommended

Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Expert
Values	True False

Reads the state of the internal acquisition signal selected using **AcquisitionStatusSelector.**

Trigger Control features:

The Trigger Control section describes all features related to image acquisition using trigger(s).

One or many **Trigger**(s) can be used to control the start of an **Acquisition** (see Figure 5-1)**,** of a **Frame** of an Acquisition (see Figure 5-2) or each **Line** of a Frame (for line scan devices)**.** It can also be used to control the exposure duration at the beginning of a frame.

TriggerSelector is used to select which type of trigger to configure. The standard trigger types are: **AcquisitionStart, AcquisitionEnd, AcquisitionActive, FrameStart, FrameEnd, FrameActive, LineStart, ExposureStart, ExposureEnd** and **ExposureActive.**

TriggerMode activate/desactivate trigger operation. It can be **Off** or **On.**

TriggerSource specifies the physical input **Line** or internal signal to use for the selected trigger. Standard trigger sources are: **Software, Line0, Line1, …, Timer1Start, Timer1End, , …, Counter1Start, Counter1End, …, UserOutput0, UserOutput1, …, Action1, Action2, …**

With a **Software** trigger source, the **TriggerSoftware** command can be used by an application to generate an internal trigger signal.

With the hardware trigger sources, **TriggerActivation** specifies the activation mode of the trigger. This can be a **RisingEdge, FallingEdge, AnyEdge, LevelHigh** or **LevelLow**.

TriggerOverlap specifies the type of trigger overlap permitted with the previous frame. This defines when a valid trigger will be accepted (or latched) for a new frame. This can be **Off** for no overlap, **ReadOut** to accept a trigger immediately after the exposure period or **PreviousFrame** to accept (latch) a trigger that happened at any time after the start of the previous frame.

TriggerDelay specifies the delay to apply after the trigger reception before to effectively activate it.

TriggerDivider and **TriggerMultiplier** are used to control the ratio of triggers that are accepted.

Note that, a trigger is considered valid after the Delay, Dividers, ...

For example to setup a hardware triggered acquisition that will start the capture of each frame on the rising edge of the signal coming from the physical input Line 1, the following pseudo-code can be used:

```
Camera.TriggerSelector     = FrameStart;
Camera.TriggerMode         = On;
Camera.TriggerActivation   = RisingEdge;
Camera.TriggerSource       = Line1;
```

See also Chapter 14: Acquisition and Trigger Examples for more complete use cases of the acquisition and trigger features in conjunction with other related sections such as I/O and analog controls.

5.16 TriggerSelector

Name	TriggerSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	AcquisitionStart AcquisitionEnd AcquisitionActive FrameStart FrameEnd FrameActive LineStart ExposureStart ExposureEnd ExposureActive

Selects the type of trigger to configure.

TriggerSelector can take any of the following values (see Figure 5-1 and Figure 5-2):

- **AcquisitionStart**: Selects a trigger that starts the Acquisition of one or many frames according to **AcquisitionMode**.
- **AcquisitionEnd**: Selects a trigger that ends the Acquisition of one or many frames according to **AcquisitionMode**.

- **AcquisitionActive**: Selects a trigger that controls the duration of the Acquisition of one or many frames. The Acquisition is activated when the trigger signal becomes active and terminated when it goes back to the inactive state.
- **FrameStart**: Selects a trigger starting the capture of one frame.
- **FrameEnd**: Selects a trigger ending the capture of one frame (mainly used in line scan mode).
- **FrameActive**: Selects a trigger controlling the duration of one frame (mainly used in line scan mode).
- **LineStart**: Selects a trigger starting the capture of one Line of a Frame (mainly used in line scan mode).
- **ExposureStart**: Selects a trigger controlling the start of the exposure of one Frame (or Line).
- **ExposureEnd**: Selects a trigger controlling the end of the exposure of one Frame (or Line).
- **ExposureActive**: Selects a trigger controlling the duration of the exposure of one frame (or Line).

5.17 TriggerMode

Name	TriggerMode[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off On

Controls if the selected trigger is active.

It can take any of the following values:

- **Off**: Disables the selected trigger.
- **On**: Enable the selected trigger.

5.18 TriggerSoftware

Name	TriggerSoftware[TriggerSelector]
Level	Recommended
Interface	ICommand

Access	(Read)/Write
Unit	-
Recommended Visibility	Beginner
Values	-

Generates an internal trigger. **TriggerSource** must be set to **Software.**

5.19 TriggerSource

Name	TriggerSource[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Software Line0 (If 0 based), Line1, Line2, … Timer1Start, Timer2Start, … Timer1End, Timer2End, … Counter1Start, Counter2Start, … Counter1End, Counter2End, … UserOutput0, UserOutput1, UserOutput2, ... Action1 , Action2, … *CC1, CC2, CC3, CC4,* …

Specifies the internal signal or physical input **Line** to use as the trigger source. The selected trigger must have its **TriggerMode** set to **On**.

TriggerSource can take any of the following values:

- **Software**: Specifies that the trigger source will be generated by software using the **TriggerSoftware** command.
- **Line0** (If 0 based)**, Line1, Line2,** …: Specifies which physical line (or pin) and associated I/O control block to use as external source for the trigger signal.
- **Timer1Start, Timer2Start, …, Timer1End, Timer2End,** …: Specifies which Timer signal to use as internal source for the trigger.

- **Counter1Start, Counter2Start, …, Counter1End, Counter2End,** …: Specifies which of the Counter signal to use as internal source for the trigger.
- **UserOutput0, UserOutput1, UserOutput2,** …: Specifies which User Output bit signal to use as internal source for the trigger.
- **Action 1, Action2,** …: Specifies which Action command to use as internal source for the trigger.
- **CC1, CC2, CC3, CC4**: Index of the Camera Link physical line and associated I/O control block to use. This ensures a **direct** mapping between the lines on the frame grabber and on the camera. Applicable to CameraLink Product only.

5.20 TriggerActivation

Name	TriggerActivation[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Specifies the activation mode of the trigger.

TriggerActivation can take any of the following values:

- **RisingEdge**: Specifies that the trigger is considered valid on the rising edge of the source signal.
- **FallingEdge**: Specifies that the trigger is considered valid on the falling edge of the source signal.
- **AnyEdge**: Specifies that the trigger is considered valid on the falling or rising edge of the source signal.
- **LevelHigh**: Specifies that the trigger is considered valid as long as the level of the source signal is high.
- **LevelLow**: Specifies that the trigger is considered valid as long as the level of the source signal is low.

5.21 TriggerOverlap

Name	TriggerOverlap[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off ReadOut PreviousFrame

Specifies the type trigger overlap permitted with the previous frame. This defines when a valid trigger will be accepted (or latched) for a new frame.

It can take any of the following values:

- **Off**: No trigger overlap is permitted.
- **ReadOut**: Trigger is accepted immediately after the exposure period.
- **PreviousFrame**: Trigger is accepted (latched) at any time during the capture of the previous frame.

5.22 TriggerDelay

Name	TriggerDelay[TriggerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Expert
Values	Device-specific

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

5.23 TriggerDelayAbs (Deprecated)

Name	TriggerDelayAbs[TriggerSelector]
Level	Recommended

Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. It specifies the absolute delay in microseconds (us) to apply after the trigger reception before effectively activating it. **TriggerDelayRaw** must reflect the state of **TriggerDelayAbs** when they are both supported.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

5.24 TriggerDelayRaw (Deprecated)

Name	TriggerDelayRaw[TriggerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. **TriggerDelayRaw** specifies the delay in device-specific unit to apply after the trigger reception before effectively activating it. **TriggerDelayAbs** must reflect the state of **TriggerDelayRaw** when they are both supported**.**

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

5.25 TriggerDivider

Name	TriggerDivider[TriggerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended	Expert

Visibility	
Values	Device-specific

Specifies a division factor for the incoming trigger pulses.

5.26 TriggerMultiplier

Name	TriggerMultiplier[TriggerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Specifies a multiplication factor for the incoming trigger pulses. It is used generally used in conjunction with **TriggerDivider** to control the ratio of triggers that are accepted.

Exposure Control features:

The Exposure Control section describes all features related to the exposure of the photosensitive cells (shutter control) during image acquisition.

The Exposure of the photosensitive cells during Frame or Line acquisition can be in 3 differents modes.

- **ExposureMode** can be **Off** to disable the Shutter and let it open.
- **ExposureMode** can be **Timed** to have a timed exposure and allow programing the duration using the **ExposureTime or ExposureAuto** features.

  For example to have a fixed exposure time of 1 milisecond, use the following pseudo code:

```
Camera.ExposureMode = Timed;
Camera.ExposureTime = 1000;
```

- **ExposureMode** can be **TriggerWidth** to use the width of the current Frame or Line trigger signal(s) to control exposure duration.
- **ExposureMode** can be **TriggerControlled** to use one or more trigger signal(s) to control the exposure duration independently from the current Frame or Line triggers (See **ExposureStart, ExposureEnd** and **ExposureActive** of the **TriggerSelector** feature).

For example: To use 2 hardware triggers respectively starting and stopping the Exposure, use the following pseudo code:

```
Camera.ExposureMode    = TriggerControlled;
Camera.TriggerSelector = ExposureStart;
Camera.TriggerMode     = On;
Camera.TriggerSource   = Line1;
Camera.TriggerSelector = ExposureEnd;
Camera.TriggerMode     = On;
Camera.TriggerSource   = Line2;
```

5.27 ExposureMode

Name	ExposureMode
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Timed TriggerWidth TriggerControlled

Sets the operation mode of the Exposure (or shutter).

ExposureMode can take any of the following values:

- **Off**: Disables the Exposure and let the shutter open.
- **Timed**: Timed exposure. The exposure duration time is set using the **ExposureTime** or **ExposureAuto** features and the exposure starts with the FrameStart (see Figure 5-2).
- **TriggerWidth**: Uses the width of the current Frame or Line trigger signal(s) pulse to control the exposure duration. Note that if the Frame or Line **TriggerActivation** is RisingEdge or LevelHigh, the exposure duration will be the time the trigger stays High. If **TriggerActivation** is FallingEdge or LevelLow, the exposure time will last as long as the trigger stays Low.
- **TriggerControlled**: Uses one or more trigger signal(s) to control the exposure duration independently from the current Frame or Line triggers. See **ExposureStart, ExposureEnd** and **ExposureActive** of the **TriggerSelector** feature.

Note also that **ExposureMode** as priority over the Exposure Trigger settings defined using **TriggerSelector=Exposure…** and defines which trigger (if any) is active.

For example, if:

```
ExposureMode = Timed;
ExposureTime = 200;
```

Then the Exposure will be controlled using the **ExposureTime** Feature, even if the following code is done:

```
TriggerSelector   = ExposureActive;
TriggerMode       = On;
TriggerActivation = LevelHigh;
TriggerSource     = Line1;
```

But simply by adding:

```
ExposureMode = TriggerControlled;
```

The Exposure duration will become controlled by the length of the positive pulse on physical Line 1.

5.28 ExposureTime

Name	ExposureTime
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Beginner
Values	≥0

Sets the Exposure time (in microseconds) when **ExposureMode** is **Timed**. This controls the duration where the photosensitive cells are exposed to light.

5.29 ExposureTimeAbs (Deprecated)

Name	ExposureTimeAbs
Level	Recommended

Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. It is used to set the Exposure time (in microseconds) when **ExposureMode** is **Timed**. This controls the duration where the photosensitive cells are exposed to light.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

ExposureTimeRaw must reflect the value of **ExposureTimeAbs** when they are both supported.

5.30 ExposureTimeRaw (Deprecated)

Name	ExposureTimeRaw
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. It can used to set the Exposure time in device-specific unit when **ExposureMode** is **Timed**. This controls the duration where the photosensitive cells are exposed to light.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

ExposureTimeAbs must reflect the value of **ExposureTimeRaw** when they are both supported.

5.31 ExposureAuto

Name	ExposureAuto
Level	Optional

Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Once Continuous Device-specific

Sets the automatic exposure mode when **ExposureMode** is **Timed**. The exact algorithm used to implement this control is device-specific.

Some other device-specific features might be used to allow the selection of the algorithm.

ExposureAuto can take any of the following values:

- **Off**: Exposure duration is manually controlled using **ExposureTime.**
- **Once**: Exposure duration is adapted once by the device. Once it has converged, it returns to the **Off** state.
- **Continuous**: Exposure duration is constantly adapted by the device to maximize the dynamic range.

On top of the previous standard values, a device might also provide device-specific values.

6 Digital I/O Control

Digital I/O covers the features required to control the general Input and Output signals of the device. This includes input and output control signals for Triggers Timers, counters and also static signals such as user configurable input or output bits.

The Digital I/O Control section models each I/O **Line** as a physical line that comes from the device connector and that goes into an **I/O Control Block** permitting to condition and to monitor the incoming or outgoing signal.

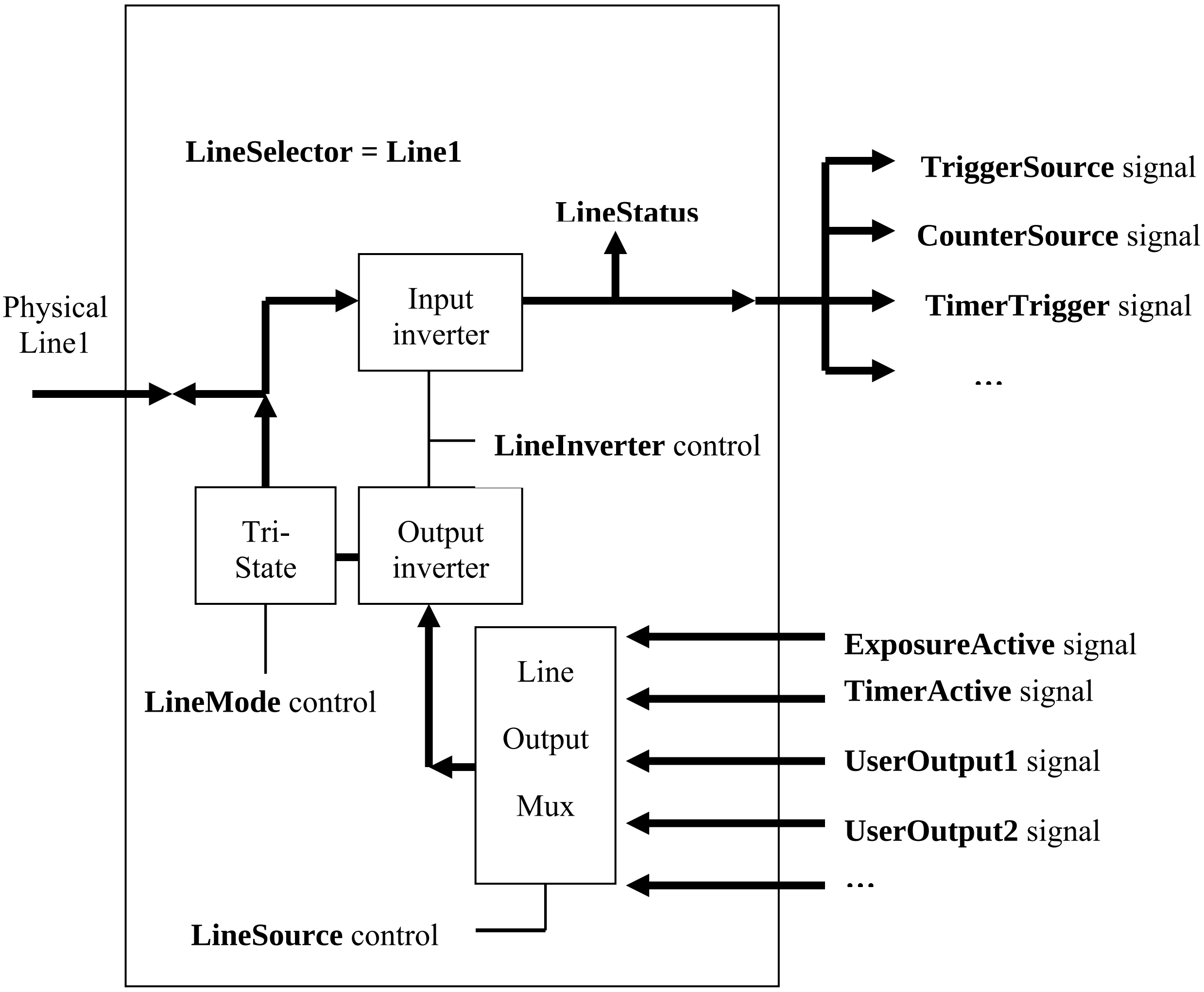

Figure 6-1: I/O Control

I/O Lines:

For a Digital I/O, when the full **I/O Control Block** is implemented, each physical **Line** (or pin) selected using **LineSelector** can be configured as Input or Output using **LineMode**. For an input or output Line, it is possible to read the Status of the Line with **LineStatus** and the incoming or outgoing signal can also be inverted using **LineInverter**. For an Output signal, the source of the signal is controlled using **LineSource** (See Figure 6-1).

For example, to output an inverted pulse coming from the Timer 1 on the physical Line 2 of the device connector, use the following code:

```
LineSelector = Line2;
LineMode     = Output;
LineInverter = True;
LineSource   = Timer1Active;
```

Note that all the features of an I/O control block are optional. Typically, an Input only line will report the **LineMode** as **Input** (read-only) and will implement only the **LineSelector**, **LineInverter** and **LineStatus** features (top half in Figure 6-1). An Output only line will report the **LineMode** as **Output** (read-only) and will implement only the **LineSelector**, **LineInverter** and **LineSource** features (bottom half of Figure 6-1). Even a hard-wired input or output line is just particular case where all the features are read-only.

The electrical format of the physical Line (TTL. LVDS, Opto-Coupled…) can be read or controlled (if supported) using **LineFormat**.

Note also that the Status of all the Lines can be monitored in one single access using **LineStatusAll.**

UserOutput:

One possible source for Output lines is the User Output bit register.

Using **LineSource**, each of the bits of the User Output register can be directed to a physical output Line after going trough the I/O control block (See Figure 6-1

UserOutputSelector and **UserOutputValue** are used to set any individual bit of the User Output register. **UserOutputValueAll** and **UserOutputValueAllMask** can be used to set all or many of the User Output bits in one access.

6.1 DigitalIOControl

Name	DigitalIOControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains the digital input and output control features.

6.2 LineSelector

Name	LineSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Line0 (If 0 based), Line1, Line2, … *CC1, CC2, CC3, CC4,* …

Selects the physical line (or pin) of the external device connector to configure.

When a Line is selected, all the other Line features will be applied to its associated I/O control block and will condition the resulting input or output signal.

LineSelector can take any of the following values:

- **Line0** (If 0 based)**, Line1, Line2, ...**: Index of the physical line and associated I/O control block to use.
- **CC1, CC2, CC3, CC4**: Index of the Camera Link physical line and associated I/O control block to use. This ensures a **direct** mapping between the lines on the frame grabber and on the camera. Applicable to CameraLink Product only.

6.3 LineMode

Name	LineMode[LineSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Input Output

Controls if the physical Line is used to Input or Output a signal.

When a Line supports input and output mode, the default state is Input to avoid possible electrical contention.

LineMode can take any of the following values:

- **Input**: The selected physical line is used to Input an electrical signal.
- **Output**: The selected physical line is used to Output an electrical signal.

6.4 LineInverter

Name	LineInverter[LineSelector]
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	False True

Controls the invertion of the signal of the selected input or output Line.

LineInverter can take any of the following values:

- **False**: The Line signal is not inverted.
- **True**: The Line signal is inverted.

6.5 LineStatus

Name	LineStatus[LineSelector]
Level	Recommended
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Expert
Values	False True

Returns the current status of the selected input or output Line.

The status of the signal is taken after the input Line inverter of the I/O control block.

LineStatus can take any of the following values:

- **True**: The level of the Line signal is High.
- **False**: The level of the Line signal is Low.

6.6 LineStatusAll

Name	LineStatusAll
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Returns the current status of all available Line signals at time of polling in a single bitfield. The order is Line0 (If 0 based), Line1, Line2,...

6.7 LineSource

Name	LineSource[LineSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTriggerWait AcquisitionActive FrameTriggerWait FrameActive ExposureActive Timer1Active, Timer2Active, … Counter1Active, Counter2Active, … UserOutput0, UserOutput1, UserOutput2, …

Selects which internal acquisition or I/O source signal to output on the selected Line. **LineMode** must be **Output.**

LineSource can take any of the following values (see Figure 6-1):

- **Off**: Line output is disabled (Tri-State).
- **AcquisitionTriggerWait**: Device is currently waiting for a trigger for the capture of one or many Frames.
- **AcquisitionActive**: Device is currently doing an acquisition of one or many Frames.
- **FrameTriggerWait**: Device is currently waiting for a Frame trigger.
- **FrameActive**: Device is currently doing the capture of a Frame.
- **ExposureActive**: Device is doing the exposure of a Frame (or Line).
- **Timer1Active, Timer2Active,** …: The chosen Timer is in active state.
- **Counter1Active, Counter2Active,** …: The chosen counter is in active state (counting).
- **UserOutput0, UserOutput1, UserOutput2,** …: The chosen User Output Bit state as defined by its current **UserOutputValue**.

6.8 LineFormat

Name	LineFormat[LineSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	NoConnect TriState TTL LVDS RS422 OptoCoupled

Controls the current electrical format of the selected physical input or output **Line**.

LineFormat can take any of the following values:

- **NoConnect**: The Line is not connected.
- **TriState:** The Line is currently in Tri-State mode (Not driven).
- **TTL:** The Line is currently accepting or sending TTL level signals.

- **LVDS:** The Line is currently accepting or sending LVDS level signals.
- **RS422:** The Line is currently accepting or sending RS422 level signals.
- **OptoCoupled**: The Line is opto-coupled.

6.9 UserOutputSelector

Name	UserOutputSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	UserOutput0, UserOutput1, UserOutput2, …

Selects which bit of the User Output register will be set by **UserOutputValue**.

UserOutputSelector can take any of the following values (If 0 based):

- **UserOutput0:** Selects the bit 0 of the User Output register.
- **UserOutput1**: Selects the bit 1 of the User Output register.
- **UserOutput2**: Selects the bit 2 of the User Output register.
- …

6.10 UserOutputValue

Name	UserOutputValue[UserOutputSelector]
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Sets the value of the bit selected by UserOutputSelector.

UserOutputValue can take any of the following values:

- **True**: Sets the bit to High.
- **False**: Sets the bit to Low.

6.11 UserOutputValueAll

Name	UserOutputValueAll
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Sets the value of all the bits of the User Output register. It is subject to the **UserOutputValueAllMask**.

UserOutputValueAll can take any binary value and each bit set to one will set the corresponding User Output register bit to high. Note that the UserOutputs are numbered from 0 to N (If 0 based). This means that Bit 0 of **UserOutputValueAll** corresponds to the UserOutput0.

6.12 UserOutputValueAllMask

Name	UserOutputValueAllMask
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Sets the write mask to apply to the value specified by **UserOutputValueAll** before writing it in the User Output register. If the **UserOutputValueAllMask** feature is present, setting the user Output register using **UserOutputValueAll** will only change the bits that have a corresponding bit in the mask set to one.

UserOutputValueAllMask can take any binary value. Each bit set to one will enable writing of the corresponding User Output register bit and each bit set to zero will prevent it.

Note that **UserOutputValueAllMask** is ignored when an individual bit is set using **UserOutputValue**.

7 Counter and Timer Control

This section lists all features that relates to control and monitoring of Counters and Timers.

A Counter is used to count internal events (FrameStart, FrameTrigger, …), I/O external events (Input Line rising edge, …) and even clock ticks. It can be reset, read or written at anytime. Counters and Timers can also be cascaded to increase their range if necessary.

Timers are readable and can be used to measure the duration of internal or external signals. A Timer can also be used to generate a timed strobe pulse with an optional delay before activation.

For example,to generate a 300 us strobe pulse coming from the Timer 1 when a rising edge trigger is detected on the physical Line 2 of the device connector, use the following code:

```
TimerSelector          = Timer1;
TimerDuration          = 300;
TimerTriggerActivation = RisingEdge;
TimerTriggerSource     = Line2;
```

To set the destination output Line of the Timer pulse, see for example **Timer1Active** entry of the **LineSource** feature.

Note that Counters and Timers can also be used to generate an Event when a predetermined maximum count (or duration) is reached. See the **EventSelector** feature.

7.1 CounterAndTimerControl

Name	CounterAndTimerControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains the Counter and Timer control features.

7.2 CounterSelector

Name	CounterSelector
Level	Recommended

Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Counter1, Counter2, …

Selects which counter to configure.

CounterSelector can take any of the following values:

- **Counter1**: Selects the first counter.
- **Counter2**: Selects the second counter.
- …

7.3 CounterEventSource

Name	CounterEventSource[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd LineStart LineEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … Counter1End, Counter2End, … Timer1End, Timer2End, … TimestampTick, Action1, Action2, … Line0RisingEdge (Deprecated) (If 0 based), Line1RisingEdge

	(Deprecated), …

Select the events that will be the source to increment the counter.

CounterSource can take any of the following values (see Figure 5-1, Figure 5-2 and Figure 5-3):

- **Off**: Counter is stopped.
- **AcquisitionTrigger**: Counts the number of Acquisition Trigger.
- **AcquisitionStart**: Counts the number of Acquisition Start.
- **AcquisitionEnd**: Counts the number of Acquisition End.
- **FrameTrigger**: Counts the number of Frame Trigger.
- **FrameStart**: Counts the number of Frame start.
- **FrameEnd**: Counts the number of Frame end.
- **LineStart**: Counts the number of Line start.
- **LineEnd**: Counts the number of Line end.
- **ExposureStart**: Counts the number of Exposure start.
- **ExposureEnd**: Counts the number of Exposure end.
- **Line1, Line2,** …: Counts the number of transitions on the chosen I/O Line.
- **Counter1End, Counter2End,** …: Counts the number of Counter end when counter are cascaded.
- **Timer1End, Timer2End,** …: Counts the number of Timer pulses generated.
- **TimestampTick**: Counts the number of clock ticks of the Timestamp clock. Can be used to create a programmable timer.
- **Action1, Action2, …** : Counts the number of assertions of the chosen action signal.
- **Line1RisingEdge, Line2RisingEdge,** …: **(Deprecated, see CounterEventActivation)** Counts the number of rising edge transitions on the chosen I/O Line.

7.4 CounterEventActivation

Name	CounterEventActivation[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert

Values	RisingEdge FallingEdge AnyEdge

Selects the Activation mode Event Source signal.

CounterEventActivation can take any of the following values:

- **RisingEdge**: Counts on the Rising Edge of the signal.
- **FallingEdge**: Counts on the Falling Edge of the signal.
- **AnyEdge**: Counts on the Falling or rising Edge of the selected signal.

7.5 CounterResetSource

Name	CounterResetSource[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off CounterTrigger AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … Counter1End, Counter2End, … Timer1End, Timer2End, … UserOutput0, UserOutput1, UserOutput2,…Action1, Action2, …

Selects the signals that will be the source to reset the counter.

CounterResetSource can take any of the following values:

- **Off**: Disable the Counter Reset trigger.
- **CounterTrigger**: Resets with the reception of a trigger on the **CounterTriggerSource**.
- **AcquisitionTrigger**: Resets with the reception of the Acquisition Trigger.

- **AcquisitionStart**: Resets with the reception of the Acquisition Start.
- **AcquisitionEnd**: Resets with the reception of the Acquisition End.
- **FrameTrigger**: Resets with the reception of the Frame Trigger.
- **FrameStart**: Resets with the reception of the Frame Start.
- **FrameEnd**: Resets with the reception of the Frame End.
- **ExposureStart**: Resets with the reception of the Exposure Start.
- **ExposureEnd**: Resets with the reception of the Exposure End.
- **Line0** (If 0 based)**, Line1, Line 2,** …: Resets by the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2,** …: Resets by the chosen User Output bit.
- **Counter1End, Counter2End,** …: Resets with the reception of the Counter end.
- **Timer1End, Timer2End,** …: Resets with the reception of the Timer end.
- **Action1, Action2**, … :. Resets on assertions of the chosen action signal (Broadcasted signal on the transportlayer).

Note that the value of the Counter at time of reset is automatically latched and reflected in **CounterValueAtReset.**

7.6 CounterResetActivation

Name	CounterResetActivation[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Selects the Activation mode of the Counter Reset Source signal.

CounterResetActivation can take any of the following values:

- **RisingEdge**: Resets the counter on the Rising Edge of the signal.
- **FallingEdge**: Resets the counter on the Falling Edge of the signal.
- **AnyEdge**: Resets the counter on the Falling or rising Edge of the selected signal.

- **LevelHigh**: Resets the counter as long as the selected signal level is High.
- **LevelLow**: Resets the counter as long as the selected signal level is Low.

7.7 CounterReset

Name	CounterReset[CounterSelector]
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Expert
Values	-

Does a software reset of the selected counter and starts it. The counter starts counting events immediately after the reset unless a Counter trigger is active. CounterReset can be used to reset the Counter independently from the CounterResetSource. To disable the counter temporarily, set **CounterEventSource** to **Off**.

Note that the value of the Counter at time of reset is automatically latched and reflected in the **CounterValueAtReset.**

7.8 CounterValue

Name	CounterValue[CounterSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

Reads or writes the current value of the selected counter.

Writing to CounterValue is typically used to set the start value.

7.9 CounterValueAtReset

Name	CounterValueAtReset[CounterSelector]
Level	Recommended

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Reads the value of the selected counter when it was reset by a trigger or by an explicit **CounterReset** command.

It represents the last counter value latched before to reset the counter.

7.10 CounterDuration

Name	CounterDuration[CounterSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

Sets the duration (or number of events) before the **CounterEnd** event is generated.

When the counter reaches the **CounterDuration** value, a **CounterEnd** event is generated, the **CounterActive** signal becomes inactive and the counter stops counting until a new trigger happens or it is explicitly reset with **CounterReset.**

7.11 CounterStatus

Name	CounterStatus[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Expert
Values	CounterIdle CounterTriggerWait

	CounterActive CounterCompleted CounterOverflow

Returns the current state of the counter**.**

CounterStatus can take any of the following values:

- **CounterIdle**: The counter is idle. **CounterEventSource** is **Off**.
- **CounterTriggerWait**: The counter is waiting for a start trigger.
- **CounterActive**: The counter is counting for the specified duration.
- **CounterCompleted**: The counter reached the **CounterDuration** count.
- **CounterOverflow**: The counter reached its maximum possible count.

7.12 CounterTriggerSource

Name	CounterTriggerSource[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … UserOutput0, UserOutput1, UserOutput2,… Counter1End, Counter2End, … Timer1End, Timer2End, … Action1, Action2, …

Selects the source to start the counter. **CounterTriggerSource** can take any of the following values:

- **Off**: Disables the Counter trigger.

- **AcquisitionTrigger**: Starts with the reception of the Acquisition Trigger.
- **AcquisitionStart**: Starts with the reception of the Acquisition Start.
- **AcquisitionEnd**: Starts with the reception of the Acquisition End.
- **FrameTrigger**: Starts with the reception of the Frame Trigger.
- **FrameStart**: Starts with the reception of the Frame start.
- **FrameEnd**: Starts with the reception of the Frame end.
- **ExposureStart**: Starts with the reception of the Exposure start.
- **ExposureEnd**: Starts with the reception of the Exposure end.
- **Line0** (If 0 based)**, Line1, Line2,** …: Starts when the specified CounterTriggerActivation condition is met on the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2,** …: Specifies which User Output bit signal to use as internal source for the trigger.
- **Counter1End, Counter2End,** …: Starts with the reception of the Counter end. **Timer1End, Timer2End, …:** Starts with the reception of the Timer end.
- **Action1, Action2, …: Starts** with the assertion of the chosen action signal.

7.13 CounterTriggerActivation

Name	CounterTriggerActivation[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Selects the activation mode of the trigger to start the counter.

CounterTriggerActivation can take any of the following values:

- **RisingEdge**: Starts counting on the Rising Edge of the selected trigger signal.
- **FallingEdge**: Starts counting on the Falling Edge of the selected trigger signal.
- **AnyEdge**: Starts counting on the Falling or rising Edge of the selected trigger signal.
- **LevelHigh**: Counts as long as the selected trigger signal level is High.

- **LevelLow**: Counts as long as the selected trigger signal level is Low.

7.14 TimerSelector

Name	TimerSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Timer1, Timer2, …

Selects which Timer to configure.

TimerSelector can take any of the following values:

- **Timer1:** Selects the first Timer.
- **Timer2:** Selects the second Timer.

7.15 TimerDuration

Name	TimerDuration[TimerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Expert
Values	≥0

Sets the duration (in microseconds) of the Timer pulse.

When the Timer reaches the **TimerDuration** value, a **TimerEnd** event is generated, the **TimerActive** signal becomes low and the Timer stops counting until a new trigger happens or it is explicitly reset with **TimerReset**.

7.16 TimerDurationAbs (Deprecated)

Name	TimerDurationAbs[TimerSelector]

Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. It sets the duration (in microseconds) of the Timer pulse.

When the Timer reaches the **TimerDurationAbs** value, a **TimerEnd** event is generated, the **TimerActive** signal becomes low and the Timer stops counting until a new trigger happens or it is explicitly reset with **TimerReset**.

TimerDurationRaw must reflect the state of **TimerDurationAbs** when they are both supported**.**

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

7.17 TimerDurationRaw (Deprecated)

Name	TimerDurationRaw[TimerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. It sets the duration in device-specific unit of the Timer pulse.

When the Timer reaches the **TimerDurationRaw** value, a **TimerEnd** event is generated, the **TimerActive** signal becomes low and the Timer stops counting until a new trigger happens or it is explicitly reset with **TimerReset**.

TimerDurationAbs must reflect the state of **TimerDurationRaw** when they are both supported**.**

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

7.18 TimerDelay

Name	TimerDelay[TimerSelector]

Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Expert
Values	≥0

Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before to start the Timer.

7.19 TimerDelayAbs (Deprecated)

Name	TimerDelayAbs[TimerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. This feature sets the duration (in microseconds) of the delay to apply after the reception of a trigger before to start the Timer.

TimerDelayRaw must reflect the state of **TimerDelayAbs** when they are both supported.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

7.20 TimerDelayRaw (Deprecated)

Name	TimerDelayRaw[TimerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible

Values	≥0

This feature is deprecated. It sets the duration in device-specific unit of the delay to apply after the reception of a trigger before to start the Timer.

TimerDelayAbs must reflect the state of **TimerDelayRaw** when they are both supported.

- It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

7.21 TimerReset

Name	TimerReset[TimerSelector]
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Expert
Values	-

Does a software reset of the selected timer and starts it. The timer starts counting events immediately after the reset unless a timer trigger is active.

7.22 TimerValue

Name	TimerValue[TimerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Expert
Values	≥0

Reads or writes the current value (in microseconds) of the selected Timer.

Writing **TimerValue** is typically used to set the start value.

7.23 TimerValueAbs(Deprecated)

Name	TimerValueAbs[TimerSelector]

Level	Recommended
Interface	IFloat
Access	Read
Unit	us
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. It returns the current value (in microseconds) of the selected Timer.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

7.24 TimerValueRaw (Deprecated)

Name	TimerValueRaw[TimerSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. This feature is used to read the current value in device-specific unit of the selected Timer.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

7.25 TimerStatus

Name	TimerStatus[TimerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Expert

Values	TimerIdle TimerTriggerWait TimerActive TimerCompleted

Returns the current state of the Timer**.**

TimerStatus can take any of the following values:

- **TimerIdle**: The Timer is idle. **TimerTriggerSource** is **Off**.
- **TimerTriggerWait**: The Timer is waiting for a start trigger.
- **TimerActive**: The Timer is counting for the specified duration.
- **TimerCompleted**: The Timer reached the **TimerDuration** count.

7.26 TimerTriggerSource

Name	TimerTriggerSource[TimerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2,… UserOutput0, UserOutput1, UserOutput2,…Counter1End, Counter2End, … Timer1End, Timer2End, … Action1, Action2, …

Selects the source of the trigger to start the Timer.

TimerTriggerSource can take any of the following values:

- **Off**: Disables the Timer trigger.

- **AcquisitionTrigger**: Starts with the reception of the Acquisition Trigger.
- **AcquisitionStart**: Starts with the reception of the Acquisition Start.
- **AcquisitionEnd**: Starts with the reception of the Acquisition End.
- **FrameTrigger**: Starts with the reception of the Frame Trigger.
- **FrameStart**: Starts with the reception of the Frame start.
- **FrameEnd**: Starts with the reception of the Frame end.
- **ExposureStart**: Starts with the reception of the Exposure start.
- **ExposureEnd**: Starts with the reception of the Exposure end.
- **Line0** (If 0 based)**, Line1, Line2,** …: Starts when the specidfied TimerTriggerActivation condition is met on the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2,** …: Specifies which User Output bit signal to use as internal source for the trigger.
- **Counter1End, Counter2End,** …: Starts with the reception of the counter end.
- **Timer1End, Timer2End,** …: Starts with the reception of the Timer end when Timer are cascaded. Note that a timer can retrigger itself to achieve a free running Timer.
- **Action1, Action2, …:** Starts with the assertion of the chosen action signal.

7.27 TimerTriggerActivation

Name	TimerTriggerActivation[TimerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Selects the activation mode of the trigger to start the Timer.

TimerTriggerActivation can take any of the following values:

- **RisingEdge**: Starts counting on the Rising Edge of the selected trigger signal.
- **FallingEdge**: Starts counting on the Falling Edge of the selected trigger signal.
- **AnyEdge**: Starts counting on the Falling or Rising Edge of the selected trigger signal.

- **LevelHigh**: Counts as long as the selected trigger signal level is High.
- **LevelLow**: Counts as long as the selected trigger signal level is Low.

8 Event Control

This section describes how to control the generation of Events to the host application. An Event is a message that is sent to the host application to notify it of the occurrence of an internal event.

Events are typically used to synchronize the host application with some Events happening in the device. A typical use in machine vision is a host application that waits to be notified of the CCD exposure end to move the inspected part on a conveyer belt.

EventSelector selects which particular Event to control. There are 4 typical sources of events: Acquisition, Timer, Counter and I/O lines.

The standard Acquisition related Events are: **AcquisitionTrigger, AcquisitionStart, AcquisitionEnd, AcquisitionTransferStart, AcquisitionTransferEnd, AcquisitionError, FrameTrigger, FrameStart, FrameEnd, FrameTransferStart, FrameTransferEnd, ExposureStart, ExposureEnd** (see Figure 5-1, Figure 5-2 and Figure 5-3).

The standard Counters and Timers related Events are: **Counter1Start, Counter1End, Counter2Start, Counter2End, … Timer1Start, Timer End, Timer2Start, Timer2End,** …

The standard I/O line Events are: **Line0RisingEdge , Line0FallingEdge, Line0AnyEdge, Line1RisingEdge, Line1FallingEdge,** … Note that the event signal is monitored at the same place as **LineStatus** in the I/O control block (See Figure 6-1). This means that event is checked against the condition after the input inverter.

EventNotification is used to enable or disable the notification of the occurrence of the internal event selected by **EventSelector**. If **EventNotification** is **Off,** no event of the selected type is generated.

For each of the events listed in the **EventSelector** enumeration, there must be a corresponding feature with a standard name (ex: **EventExposureEnd**). The controlling application can rely on this event identifier to register a callback function to be notified that the event happened.

Also for each Event in **EventSelector**, there should be one category grouping all the related data members (Ex: **EventExposureEndData**).

The other data members in that category should also follow the naming convention described below (Ex: **EventExposureEndTimestamp**).

The recommended optional data members are:

- Timestamp: Unique timestamp of the Event.
- FrameID: Unique ID of the Frame (or image) that generated the Event.
- Followed by any other data related to this particular event (Ex: StreamingChannel for GEV, …).

Therefore, the naming convention for the Event related features is:

For each Event member of the **EventSelector** (Ex: **ExposureEnd**):

- You should provide an ICategory named:
Event prefix + “**EventName**” + **Data** postfix (Ex: **EventExposureEndData**)

- You must provide an IInteger Event feature that will be used to register the callback and is named:
Event prefix +“**EventName**” (Ex: **EventExposureEnd**).

- You should provide for each optional data member a corresponding feature named:
Event prefix + “**EventName**”+”**DataMember”** (Ex: **EventExposureEndTimestamp**).

For the **ExposureEnd** member of **EventSelector,** this would give:

ICategory **EventExposureEndData**

IInteger **EventExposureEnd**

IInteger **EventExposureEndTimeStamp**

IInteger **EventExposureEndFrameID**

…

With the above naming convention, for each Event listed in **EventSelector:**

- A user always knows the name of the Feature to use to register a call back on that Event.
- The user can take the parent of this feature to find the corresponding Event category.
- In this Event category, the user will find all the features related to this Event.

For example, to do a continuous acquisition and be notified at the end of the exposure period of each frame to move the part and also get the timestamp, the following pseudo-code can be used:

```
Register(Camera.EventExposureEnd, CallbackDataObject, CallbackFunctionPtr)

Camera.EventSelector     = ExposureEnd;
Camera.EventNotification = On;
Camera.AcquisitionMode   = Continuous;
Camera.AcquisitionStart();
…
// In the callback of the ExposureEnd event, gets the event timestamp:
Timestamp = Camera.EventExposureEndTimestamp;
…
Camera.AcquisitionStop();
```

Here below, in addition to **EventControl**, **EventSelector** and **EventNotification** should be listed all the categories and data related features for each Event listed in the **EventSelector** enumeration feature.

For simplicity, all the categories and their data members are not listed explicitly in that document but a precise naming convention for the categories and their member is provided above instead.

Below, the detailled features for the members of the **EventSelector** are only listed for 3 typically recommended events: **FrameTrigger, ExposureEnd** and **Error**.

All the other members of the **EventSelector** feature should follow the exact same pattern for their features naming and category if they are present in a device.

8.1 EventControl

Name	EventControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains Event control features.

8.2 EventSelector

Name	EventSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	AcquisitionTrigger AcquisitionStart AcquisitionEnd AcquisitionTransferStart AcquisitionTransferEnd

	AcquisitionError FrameTrigger FrameStart FrameEnd FrameTransferStart FrameTransferEnd ExposureStart ExposureEnd Counter1Start, … Counter1End, … Timer1Start, … Timer1End, … Line0RisingEdge (If 0 based), Line1RisingEdge, Line2RisingEdge , … Line0FallingEdge (If 0 based), Line1FallingEdge, Line2FallingEdge , … Line0AnyEdge (If 0 based), Line1AnyEdge, Line2AnyEdge , … Error Device-specific Errors (Deprecated)

Selects which Event to signal to the host application.

EventSelector can take any of the following values (see Figure 5-1, Figure 5-2, Figure 5-3 and Figure 6-1):

- **AcquisitionTrigger:** Device just received a trigger for the Acquisition of one or many Frames.
- **AcquisitionStart**: Device just started the Acquisition of one or many Frames.
- **AcquisitionEnd**: Device just completed the Acquisition of one or many Frames.
- **AcquisitionTransferStart**: Device just started the transfer of one or many Frames.
- **AcquisitionTransferEnd**: Device just completed the transfer of one or many Frames.
- **AcquisitionError**: Device just detected an error during the active Acquisition.
- **FrameTrigger**: Device just received a trigger for the capture of one Frame.
- **FrameStart**: Device just started the capture of one Frame.
- **FrameEnd**: Device just completed the capture of one Frame.
- **FrameTransferStart**: Device just started the transfer of one Frame.
- **FrameTransferEnd**: Device just completed the transfer of one Frame.
- **ExposureStart**: Device just started the exposure of one Frame (or Line).

- **ExposureEnd**: Device just completed the exposure of one Frame (or Line).
- **Counter1Start**: The event will be generated when counter 1 starts counting.
- **Counter1End**: The event will be generated when counter 1 ends counting.
- **Timer1Start**: The event will be generated when Timer 1 starts counting.
- **Timer1End**: The event will be generated when Timer 1 ends counting.
- **Line1RisingEdge**: The event will be generated when a Rising Edge is detected on the Line 1.
- **Line1FallingEdge**: The event will be generated when a Falling Edge is detected on the Line 1.
- **Line1AnyEdge**: The event will be generated when a Falling or Rising Edge is detected on the Line 1.
- **Error:** The event will be generated when the device encounter an error.
- **Errors: (Deprecated)** The event will be generated when the device encounter an error.
- …

The value of the enumeration entry should correspond to the unique Identifier of the type of Event.

8.3 EventNotification

Name	EventNotification[EventSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off On GigEVisionEvent (Deprecated)

Activate or deactivate the notification to the host application of the occurrence of the selected Event.

EventNotification can take any of the following values:

- **Off**: The selected Event notification is disabled.
- **On**: The selected Event notification is enabled.
- **GigEVisionEvent (Deprecated)**: Map to On for standard GigE Vision event notifications. This enumeration is deprecated but could be included for backward compatibility with version 1.0 to 1.2.1 of this specification.

8.4 Frame Trigger Event (Example #1)

Below the recommended features for the Frame Trigger Event handling.

8.4.1 EventFrameTriggerData

Name	EventFrameTriggerData
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains all the data features related to the FrameTrigger Event.

8.4.2 EventFrameTrigger

Name	EventFrameTrigger
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the unique Identifier of the FrameTrigger type of Event. It can be used to register a callback function to be notified of the event occurrence. Its value uniquely identify the type event received.

8.4.3 EventFrameTriggerTimestamp

Name	EventFrameTriggerTimestamp
Level	Recommended

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the Timestamp of the AquisitionTrigger Event. It can be used to determine precisely when the event occured.

8.4.4 EventFrameTriggerFrameID

Name	EventFrameTriggerFrameID
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

8.5 Exposure End Event (Example #2)

Below the recommended features for the Exposure End Event handling.

8.5.1 EventExposureEndData

Name	EventExposureEndData
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert

Values	-

Category that contains all the data features related to the ExposureEnd Event.

8.5.2 EventExposureEnd

Name	EventExposureEnd
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the unique identifier of the ExposureEnd type of Event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received.

8.5.3 EventExposureEndTimestamp

Name	EventExposureEndTimestamp
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the Timestamp of the ExposureEnd Event. It can be used to determine precisely when the event occured.

8.5.4 EventExposureEndFrameID

Name	EventExposureEndFrameID
Level	Recommended

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the unique Identifier of the Frame (or image) that generated the ExposureEnd Event.

8.6 Error Event (Example #3)

Below the recommended features for the Error Event handling.

8.6.1 EventErrorData

Name	EventErrorData
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains all the data features related to the Error Event.

8.6.2 EventError

Name	EventError
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the unique identifier of the Error type of Event. It can be used to register a callback function to be notified of the Error event occurrence. Its value uniquely identify that the event received was an Error.

8.6.3 EventErrorTimestamp

Name	EventErrorTimestamp
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Returns the Timestamp of the Error Event. It can be used to determine when the event occured.

8.6.4 EventErrorFrameID

Name	EventErrorFrameID
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

If applicable, returns the unique Identifier of the Frame (or image) that generated the Error Event.

8.6.5 EventErrorCode

Name	EventErrorCode
Level	Recommended
Interface	IInteger
Access	Read

Unit	-
Recommended Visibility	Expert
Values	-

Returns an error code for the error(s) that happened.

9 Analog Control

Features in this section describes how to influence the analog features of an image, such as gain, black level, white clip and gamma.

The **Gain**, **BlackLevel** and **Gamma** features will transform the original pixel value Y to a new value Y' according to the following formula:

$$Y' = [(Y + BlackLevel) \cdot Gain]^{Gamma}$$

For some color cameras in Raw or RGB mode, the red/blue channel can be white balanced with respect to the green channel using the Red and blue **BalanceRatio** gain. For cameras in YUV mode the U/V channel can be balanced with respect to the Y channel using the U and V **BalanceRatio**, according to:

$$B' = B(BlueBalanceRatio \cdot Gain)$$

Other color camera controls each color channel gain independently, in which case, the Red, Green and Blue **Gain** features can be used for white balancing.

The automatic functions **GainAuto**, **BlackLevelAuto**, **BalanceWhiteAuto**, **GainAutoTapBalance** and **BlackLevelAutoTapBalance** can be used to auto-adjust a device once or continuously and to turn the function on and off.

Most of the automatic functions have 3 possible values: {**Off**, **Once**, **Continuous**}.

- **Off**: The automatic adjustement is disabled (ie. Manual control).
- **Once**: The automatic adjustement is performed once by the device. The affected features report the effective values. If necessary, the feature is automatically set to “Off” after the adjustment.
- **Continuous**: The automatic adjustement is continuously done by the device. The affected features report their effective values.

When a device has a specific auto-adjustment capability, it should have a corresponding feature allowing the necessary enumerations.

9.1 AnalogControl

Name	AnalogControl
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	-

Category that contains the Analog control features.

9.2 GainSelector

Name	GainSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	All Red Green Blue Y U V Tap1, Tap2, … AnalogAll AnalogRed AnalogGreen AnalogBlue AnalogY AnalogU AnalogV AnalogTap1, AnalogTap2, … DigitalAll DigitalRed

	DigitalGreen DigitalBlue DigitalY DigitalU DigitalV DigitalTap1, DigitalTap2, …

Selects which Gain is controlled by the various Gain features.

In general, there are 2 types of gain that can exist in a camera, analog or digital. Some camera will implement one or other or both. This is why there are 3 possible sets of gain.

The first one, without the **Analog** or **Digital** prefix, is to be used when only one type of gain is implemented. This permits to have an implementation independent way to set the gain.

The second and the third, with the **Analog** and **Digital** prefix, is to be used when both types of gain are implemented. This permits to have independent control over each one.

The possible values for **GainSelector** are:

- **All**: Gain will be applied to all channels or taps.
- **Red**: Gain will be applied to the red channel.
- **Green**: Gain will be applied to the green channel.
- **Blue**: Gain will be applied to the blue channel.
- **Y**: Gain will be applied to Y channel.
- **U**: Gain will be applied to U channel.
- **V**: Gain will be applied to V channel.
- **Tap1**: Gain will be applied to Tap 1.
- **Tap2**: Gain will be applied to Tap 2.
- …
- **AnalogAll**: Gain will be applied to all analog channels or taps.
- **AnalogRed**: Gain will be applied to the red analog channel.
- **AnalogGreen**: Gain will be applied to the green analog channel.
- **AnalogBlue**: Gain will be applied to the blue analog channel.
- **AnalogY**: Gain will be applied to Y analog channel.
- **AnalogU**: Gain will be applied to U analog channel.
- **AnalogV**: Gain will be applied to V analog channel.
- **AnalogTap1**: Analog gain will be applied to Tap 1.
- **AnalogTap2**: Analog gain will be applied to Tap 2.
- …

- **DigitalAll**: Gain will be applied to all digital channels or taps.
- **DigitalRed**: Gain will be applied to the red digital channel.
- **DigitalGreen**: Gain will be applied to the green digital channel.
- **DigitalBlue**: Gain will be applied to the blue digital channel.
- **DigitalY**: Gain will be applied to Y digital channel.
- **DigitalU**: Gain will be applied to U digital channel.
- **DigitalV**: Gain will be applied to V digital channel.
- **DigitalTap1**: Digital gain will be applied to Tap 1.
- **DigitalTap2**: Digital gain will be applied to Tap 2.
- …

9.3 Gain

Name	Gain[GainSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Device-specific

Controls the selected gain as an absolute physical value. This is an amplification factor applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

For color or multi-tap cameras, **GainSelector** indicates the color channel or tap to control.

9.4 GainRaw (Deprecated)

Name	GainRaw[GainSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible

Values	Device-specific

This feature is deprecated. It controls the selected gain as a raw integer value. This is an amplification factor applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

GainAbs must reflect the value put in **GainRaw** when both features are supported.

For color or multi-tap cameras, **GainSelector** indicates the color channel or tap to control.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

9.5 GainAbs (Deprecated)

Name	GainAbs[GainSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. This feature controls the selected gain as an absolute physical value. This is an amplification factor applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

GainRaw must reflect the value put in **GainAbs** when both features are supported.

For color or multi-tap cameras, **GainSelector** indicates the color channel or tap to control.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

9.6 GainAuto

Name	GainAuto[GainSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-

Recommended Visibility	Beginner
Values	Off Once Continuous Device-specific

Sets the automatic gain control (AGC) mode. The exact algorithm used to implement AGC is device-specific.

Some other device-specific features might be used to allow the selection of the algorithm.

GainAuto can take any of the following values:

- **Off**: Gain is manually controlled using **Gain.**
- **Once**: Gain is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Gain is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.7 GainAutoBalance

Name	GainAutoBalance
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Once Continuous Device-specific

Sets the mode for automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

GainAutoBalance can take any of the following values:

- **Off**: Gain tap balancing is manually controlled using **Gain** .

- **Once**: Gain tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Gain tap balancing is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.8 BlackLevelSelector

Name	BlackLevelSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	All Red Green Blue Y U V Tap1, Tap2, …

Selects which Black Level is controlled by the various Black Level features.

The possible values for **BlackLevelSelector** are:

- **All**: Black Level will be applied to all channels or taps.
- **Red**: Black Level will be applied to the red channel.
- **Green**: Black Level will be applied to the green channel.
- **Blue**: Black Level will be applied to the blue channel.
- **Y**: Black Level will be applied to Y channel.
- **U**: Black Level will be applied to U channel.
- **V**: Black Level will be applied to V channel.
- **Tap1**: Black Level will be applied to Tap 1.
- **Tap2**: Black Level will be applied to Tap 2.
- …

9.9 BlackLevel

Name	BlackLevel[BlackLevelSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Controls the analog black level as an absolute physical value. This represents a DC offset applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

For color or multi-tap cameras, **BlackLevelSelector** indicates which channel access.

9.10 BlackLevelRaw (Deprecated)

Name	BlackLevelRaw[BlackLevelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. It controls the analog black level as a raw integer value. This represents a DC offset applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

BlackLevelAbs must reflect the value put in **BlackLevelRaw** when both features are supported.

For color or multi-tap cameras, **BlackLevelSelector** indicates which channel to access.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

9.11 BlackLevelAbs (Deprecated)

Name	BlackLevelAbs[BlackLevelSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. This feature controls the analog black level as an absolute physical value. This represents a DC offset applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

BlackLevelRaw must reflect the value put in **BlackLevelAbs** when both feature are supported.

For color or multi-tap cameras, **BlackLevelSelector** indicates which channel access.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

9.12 BlackLevelAuto

Name	BlackLevelAuto[BlackLevelSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off Once Continuous Device-specific

Controls the mode for automatic black level adjustment. The exact algorithm used to implement this adjustment is device-specific.

Some other device-specific features might be used to allow the selection of the algorithm.

BlackLevelAuto can take any of the following values:

- **Off**: Analog black level is manually controlled using **BlackLevel**.
- **Once**: Analog black level is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Analog black level is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.13 BlackLevelAutoBalance

Name	BlackLevelAutoBalance
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off Once Continuous Device-specific

Controls the mode for automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

BlackLevelAutoBalance can take any of the following values:

- **Off**: Black level tap balancing is manually controlled using **BlackLevel**.
- **Once**: Black level tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Black level tap balancing is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.14 WhiteClipSelector

Name	WhiteClipSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-

Recommended Visibility	Expert
Values	All Red Green Blue Y U V Tap1, Tap2, …

Selects which White Clip to control.

The possible values for **WhiteClipSelector** are:

- **All**: White Clip will be applied to all channels or taps.
- **Red**: White Clip will be applied to the red channel.
- **Green**: White Clip will be applied to the green channel.
- **Blue**: White Clip will be applied to the blue channel.
- **Y**: White Clip will be applied to Y channel.
- **U**: White Clip will be applied to U channel.
- **V**: White Clip will be applied to V channel.
- **Tap1**: White Clip will be applied to Tap 1.
- **Tap2**: White Clip will be applied to Tap 2.
- …

9.15 WhiteClip

Name	WhiteClip[WhiteClipSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value. The video signal will never exceed the white clipping point: it will saturate at that level.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

For color or multi-tap cameras, **WhiteClipTapSelector** indicates the channel to control.

9.16 WhiteClipRaw (Deprecated)

Name	WhiteClipRaw[WhiteClipSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. Controls the maximal intensity taken by the video signal before being clipped as a raw integer value. The video signal will never exceed the white clipping point: it will saturate at that level.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

WhiteClipAbs must reflect the value put in **WhiteClipRaw** when both feature are supported.

For color or multi-tap cameras, **WhiteClipTapSelector** indicates the channel to control.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

9.17 WhiteClipAbs (Deprecated)

Name	WhiteClipAbs[WhiteClipSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	Device-specific

This feature is deprecated. Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value. The video signal will never exceed the white clipping point: it will saturate at that level.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

WhiteClipRaw must reflect the value put in **WhiteClipAbs** when both feature are supported.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

For color or multi-tap cameras, **WhiteClipTapSelector** indicates the channel to control.

9.18 BalanceRatioSelector

Name	BalanceRatioSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Red Green Blue Y U V Tap1, Tap2, …

Selects which Balance ratio to control.

The possible values for **BalanceRatioSelector** are:

- **Red**: Balance Ratio will be applied to the red channel.
- **Green**: Balance Ratio will be applied to the green channel.
- **Blue**: Balance Ratio will be applied to the blue channel.
- **Y**: Balance Ratio will be applied to Y channel.
- **U**: Balance Ratio will be applied to U channel.
- **V**: Balance Ratio will be applied to V channel.
- **Tap1**: Balance Ratio will be applied to Tap 1.
- **Tap2**: Balance Ratio will be applied to Tap 2.
- …

9.19 BalanceRatio

Name	BalanceRatio[BalanceRatioSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	>0.0

Controls ratio of the selected color component to a reference color component. It is used for white balancing.

For example, the Color balance is realized by the following formula:

C_w = **BalanceRatio** x C

where

C_w is the intensity of selected color component after white balancing.

BalanceRatio is the white balance coefficient.

C is the intensity of the color component before white balancing.

9.20 BalanceRatioAbs (Deprecated)

Name	BalanceRatioAbs[BalanceRatioSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	>0.0

This feature is deprecated. It controls the ratio of the selected color component to a reference color component. This feature is used for white balancing.

For example, the Color balance is realized by the following formula:

C_w = **BalanceRatioAbs** x C

where

C_w is the intensity of selected color component after white balancing.

BalanceRatioAbs is the white balance coefficient.

C is the intensity of the color component before white balancing.

It can be included as an invisible feature for backward compatibility with version 1.0 to 1.2.1 of this specification.

9.21 BalanceWhiteAuto

Name	BalanceWhiteAuto
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off Once Continuous Device-specific

Controls the mode for automatic white balancing between the color channels. The white balancing ratios are automatically adjusted.

BalanceWhiteAuto can take any of the following values:

- **Off**: White balancing is manually controlled using **BalanceRatioSelector** and **BalanceRatio**.
- **Once**: White balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: White balancing is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.22 Gamma

Name	Gamma
Level	Optional

Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	>0.0

Controls the gamma correction of pixel intensity. This is typically used to compensate for non-linearity of the display system (such as CRT).

Gamma correction is realized by the following formula:

$$Y' = Y^{\mathbf{Gamma}}$$

where

Y' is the new pixel intensity

Y is the original pixel intensity

Gamma is the correction factor

The realization of the gamma correction can be implemented using a LUT. Therefore, it is possible that some LUT functionality is not available when gamma correction is activated.

10 LUT Control

Features in this section describe the Look-up table (LUT) realated features.

10.1 LUTControl

Name	LUTControl
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that includes the LUT control features.

10.2 LUTSelector

Name	LUTSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Luminance Red Green Blue Device-specific

Selects which LUT to control.

It is typically not available when only a single LUT is supported.

10.3 LUTEnable

Name	LUTEnable[LUTSelector]
Level	Optional

Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Activates the selected LUT.

10.4 LUTIndex

Name	LUTIndex[LUTSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Control the index (offset) of the coefficient to access in the selected LUT.

10.5 LUTValue

Name	LUTValue[LUTSelector][LUTIndex]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	Device-specific

Returns the Value at entry **LUTIndex** of the LUT selected by **LUTSelector**.

10.6 LUTValueAll

Name	LUTValueAll[LUTSelector]
Level	Optional
Interface	IRegister
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	Device-specific

Accesses all the LUT coefficients in a single access without using individual **LUTIndex**.

11 GenICam Access

This section provides the necessary features to use the GenICam feature tree.

Note: In case of discrepancy between the features described in this section and the "GenICam Standard text" the SFNC document prevail.

11.1 Root

Name	Root
Level	Mandatory
Interface	ICategory
Access	Read
Unit	-
Mandatory Visibility	Beginner
Values	-

Provides the Root of the GenICam features tree.

11.2 Device

Name	Device
Level	Mandatory
Interface	IPort
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	-

Provides the default GenICam port of the Device.

Note: **Device** is the name of the standard port that is used to connect the node map to the transport layer and access the control port of the device. **Device** is a port node (not a feature node) and is generally not accessed by the end user directly. **Device** must not be included in the root feature tree.

11.3 TLParamsLocked

Name	TLParamsLocked

Level	Mandatory
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	-

Used by the Transport Layer to prevent critical features changes during acquisition.

TLParamsLocked can take any of the following values:

- 0 : No features are locked.
- 1 : Transport Layer and Device critical features are locked and cannot be changed.

12 Transport Layer Control

This section provides the Transport Layer control features.

The generic features are under the TransportLayerCategory and Transport layer specific features are under their respective sub category.

The GigEVision category:

This category lists the features necessary to access GigE Vision bootstrap registers and other information related to the GigE Vision transport medium. Note most of these registers are mapped according to GigE Vision specification.

In most situations, these registers are directly handled by the framework managing the transport layer on the PC and are not directly visible to user (for example, deciding which UDP port number to use for a stream channel).

More information about exact meaning of these features is found in the GigE Vision specification. The GigE Vision specification shall have precedence over this list in case of discrepancy.

Convention for the GEV section:

- All GigE Vision features start with the "Gev" prefix.
- GigE Vision registers are 32-bit. If a GigE Vision register has multiple fields within this 32-bit, then they are separated in multiple features.

Note: If the user has configured the camera front end, he can read from the back end which **PayloadSize** will be transferred for each image. This number covers all kind of data coming with the image, e.g. stamps etc. If the user allocates **PayloadSize** for each buffer he is insured that each frame will fit into his target buffers.

12.1 TransportLayerControl

Name	TransportLayerControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	-

Category that contains the Transport layer control features.

12.2 PayloadSize

Name	PayloadSize
Level	Mandatory
Interface	IInteger
Access	Read
Unit	B
Recommended Visibility	Expert
Values	>0

Provides the number of bytes transferred for each image or chunk on the stream channel. This includes any end-of-line, end-of-frame statistics or other stamp data. This is the total size of data payload for a data block.

This is mainly used by the application software to determine size of image buffers to allocate (largest buffer possible for current mode of operation).

For example, an image with no statistics or stamp data as **PayloadSize** equals to (width x height x pixel size) in bytes. It is strongly recommended to retrieve **PayloadSize** from the camera instead of relying on the above formula.

For GEV, UDP and GVSP headers are not considered. Data leader and data trailer are not included.

12.3 GevVersionMajor

Name	GevVersionMajor
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Major version of the specification.

For instance, GigE Vision version 1.0 would have the major version set to 1.

12.4 GevVersionMinor

Name	GevVersionMinor
Level	Recommended

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Minor version of the specification.

For instance, GigE Vision version 1.0 would have the minor version set to 0.

12.5 GevDeviceModelIsBigEndian

Name	GevDeviceModeIsBigEndian
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Guru
Values	True False

Endianess of the device registers.

It might be used to interpret multi-byte data for READMEM and WRITEMEM commands. Note this bit has no effect on the endianess of the GigE Vision protocol headers: they are always big-endian.

12.6 GevDeviceClass

Name	GevDeviceClass
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Guru
Values	Transmitter Receiver

	Transceiver Peripheral

Returns the class of the device.

Note: The **GevDeviceClass** feature returns **Transmitter** for cameras.

12.7 GevDeviceModeCharacterSet

Name	GevDeviceModeCharacterSet
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Guru
Values	UTF8

Character set used by all the strings of the bootstrap registers.

12.8 GevInterfaceSelector

Name	GevInterfaceSelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥0

Selects which physical network interface to control.

12.9 GevMACAddress

Name	GevMACAddress[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read

Unit	-
Recommended Visibility	Beginner
Values	≥0

MAC address of the network interface.

This feature must return a 64-bit value representing the full MAC address of the device i.e. the high and low parts.

12.10 GevSupportedOptionSelector

Name	GevSupportedOptionSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	IPConfigurationLLA IPConfigurationDHCP IPConfigurationPersistentIP StreamChannelSourceSocket MessageChannelSourceSocket CommandsConcatenation WriteMem PacketResend Event EventData PendingAck Action PrimaryApplicationSwitchover ExtendedStatusCodes DiscoveryAckDelay DiscoveryAckDelayWritable TestData

	ManifestTable CCPApplicationSocket LinkSpeed HeartbeatDisable SerialNumber UserDefinedName StreamChannel0BigAndLittleEndian StreamChannel0IPReassembly StreamChannel0UnconditionalStreaming StreamChannel0ExtendedChunkData StreamChannel1BigAndLittleEndian StreamChannel1IPReassembly StreamChannel1UnconditionalStreaming StreamChannel1ExtendedChunkData StreamChannel2BigAndLittleEndian StreamChannel2IPReassembly StreamChannel2UnconditionalStreaming StreamChannel2ExtendedChunkData …

Selects the GEV option to interrogate for existing support.

Note: The IP reassembly options (**StreamChannel0IPReassembly**, **StreamChannel1IPReassembly**, …) are only applicable to GVSP receiver stream channels.

12.11 GevSupportedOption

Name	GevSupportedOption[GevSupportedOptionSelector]
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Expert
Values	True

	False

Returns if the selected GEV option is supported.

12.12 GevSupportedIPConfigurationLLA (Deprecated)

Name	GevSupportedIPConfigurationLLA[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if Link Local Address IP configuration scheme is supported by the given network interface.

12.13 GevSupportedIPConfigurationDHCP (Deprecated)

Name	GevSupportedIPConfigurationDHCP[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if DHCP IP configuration scheme is supported by the given network interface.

12.14 GevSupportedIPConfigurationPersistentIP (Deprecated)

Name	GevSupportedIPConfigurationPersistentIP[GevInterfaceSelector]
Level	Optional

Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if PersistentIP configuration scheme is supported by the given network interface.

12.15 GevCurrentIPConfiguration (Deprecated)

Name	GevCurrentIPConfiguration[GevInterfaceSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	PersistentIP DHCP LLA

This feature is deprecated. GevSupportedOption should be used instead. It reports the current IP Configuration scheme. Note that this feature doesn't provision more that one simultaneous IP configuration and should not be used.

12.16 GevCurrentIPConfigurationLLA

Name	GevCurrentIPConfigurationLLA[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	True

Controls whether the Link Local Address IP configuration scheme is activated on the given network interface.

12.17 GevCurrentIPConfigurationDHCP

Name	GevCurrentIPConfigurationDHCP[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	True False

Controls whether the DHCP IP configuration scheme is activated on the given network interface.

12.18 GevCurrentIPConfigurationPersistentIP

Name	GevCurrentIPConfigurationPersistentIP[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	True False

Controls whether the PersistentIP configuration scheme is activated on the given network interface.

12.19 GevCurrentIPAddress

Name	GevCurrentIPAddress[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended	Beginner

Visibility	
Values	≥0

Reports the IP address for the given network interface.

12.20 GevCurrentSubnetMask

Name	GevCurrentSubnetMask[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	≥0

Reports the subnet mask of the given interface.

12.21 GevCurrentDefaultGateway

Name	GevCurrentDefaultGateway[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	≥0

Reports the default gateway IP address to be used on the given network interface.

12.22 GevIPConfigurationStatus

Name	GevIPConfigurationStatus[GevInterfaceSelector]
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-

Recommended Visibility	Beginner
Values	None PersistentIP DHCP LLA ForceIP

Reports the current IP configuration status.

12.23 GevFirstURL

Name	GevFirstURL
Level	Optional
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

Indicates the first URL to the XML device description file. The First URL is used as the first choice by the application to retrieve the XML device description file.

12.24 GevSecondURL

Name	GevSecondURL
Level	Optional
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

Indicates the second URL to the XML device description file. This URL is an alternative if the application was unsuccessful to retrieve the device description file using the first URL.

12.25 GevNumberOfInterfaces

Name	GevNumberOfInterfaces
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Indicates the number of physical network interfaces supported by this device.

12.26 GevPersistentIPAddress

Name	GevPersistentIPAddress[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥0

Controls the Persistent IP address for this network interface. It is only used when the device boots with the Persistent IP configuration scheme.

12.27 GevPersistentSubnetMask

Name	GevPersistentSubnetMask[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥0

Controls the Persistent subnet mask associated with the Persistent IP address on this network interface. It is only used when the device boots with the Persistent IP configuration scheme.

12.28 GevPersistentDefaultGateway

Name	GevPersistentDefaultGateway[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	≥0

Controls the persistent default gateway for this network interface. It is only used when the device boots with the Persistent IP configuration scheme.

12.29 GevLinkSpeed

Name	GevLinkSpeed[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	Mbps
Recommended Visibility	Expert
Values	>0

Indicates the speed of transmission negotiated by the given network interface.

12.30 GevMessageChannelCount

Name	GevMessageChannelCount
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert

Values	0 or 1

Indicates the number of message channels supported by this device.

12.31 GevStreamChannelCount

Name	GevStreamChannelCount
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	0 to 512

Indicates the number of stream channels supported by this device.

12.32 GevSupportedOptionalCommandsUserDefinedName (Deprecated)

Name	GevSupportedOptionalCommandsUserDefinedName
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the User-defined name register is supported.

12.33 GevSupportedOptionalCommandsSerialNumber (Deprecated)

Name	GevSupportedOptionalCommandsSerialNumber
Level	Optional

Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the Serial number register is supported.

12.34 GevSupportedOptionalCommandsEVENTDATA (Deprecated)

Name	GevSupportedOptionalCommandsEVENTDATA
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the EVENTDATA_CMD and EVENTDATA_ACK are supported.

12.35 GevSupportedOptionalCommandsEVENT (Deprecated)

Name	GevSupportedOptionalCommandsEVENT
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the EVENT_CMD and EVENT_ACK are supported.

12.36 GevSupportedOptionalCommandsPACKETRESEND (Deprecated)

Name	GevSupportedOptionalCommandsPACKETRESEND
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the PACKETRESEND_CMD is supported.

12.37 GevSupportedOptionalCommandsWRITEMEM (Deprecated)

Name	GevSupportedOptionalCommandsWRITEMEM
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the WRITEMEM_CMD and WRITEMEM_ACK are supported.

12.38 GevSupportedOptionalCommandsConcatenation (Deprecated)

Name	GevSupportedOptionalCommandsConcatenation
Level	Optional

Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	True False

This feature is deprecated. GevSupportedOption should be used instead. It indicates if the Multiple operations in a single message are supported.

12.39 GevHeartbeatTimeout

Name	GevHeartbeatTimeout
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	ms
Recommended Visibility	Guru
Values	>0

Controls the current heartbeat timeout in milliseconds.

12.40 GevTimestampTickFrequency

Name	GevTimestampTickFrequency
Level	Optional
Interface	IInteger
Access	Read
Unit	Hz
Recommended Visibility	Expert
Values	≥0

Indicates the number of timestamp ticks in 1 second (frequency in Hz).

This is a 64 bits number.

12.41 GevTimestampControlLatch

Name	GevTimestampControlLatch
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Expert
Values	-

Latches the current timestamp counter into GevTimestampValue.

12.42 GevTimestampControlReset

Name	GevTimestampControlReset
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Expert
Values	-

Resets the timestamp counter to 0.

12.43 GevTimestampValue

Name	GevTimestampValue
Level	Optional
Interface	IInteger
Access	Read
Unit	
Recommended Visibility	Expert
Values	≥0

Returns the latched 64-bit value of the timestamp counter.

It is necessary to latch the 64-bit timestamp value to guaranty its integrity when performing the two 32-bit read accesses to retrieve the higher and lower 32-bit portions.

12.44 GevDiscoveryAckDelay

Name	GevDiscoveryAckDelay
Level	Optional
Interface	IInteger
Access	Read/(Write)
Unit	ms
Recommended Visibility	Expert
Values	≥0 and < 1000

Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

12.45 GevGVCPExtendedStatusCodes

Name	GevGVCPExtendedStatusCodes
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

Enables the generation of extended status codes.

12.46 GevGVCPPendingAck

Name	GevGVCPPendingAck
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended	Guru

Visibility	
Values	True False

Enables the generation of PENDING_ACK.

12.47 GevGVCPHeartbeatDisable

Name	GevGVCPHeartbeatDisable
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Disables the GVCP heartbeat.

12.48 GevGVCPPendingTimeout

Name	GevGVCPPendingTimeout
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the longest GVCP command execution time before a device returns a PENDING_ACK.

12.49 GevPrimaryApplicationSwitchoverKey

Name	GevPrimaryApplicationSwitchoverKey
Level	Optional
Interface	IInteger

Access	Write-Only
Unit	-
Recommended Visibility	Guru
Values	>= 0

Controls the key to use to authenticate primary application switchover requests.

12.50 GevCCP

Name	GevCCP
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	OpenAccess ExclusiveAccess ControlAccess ControlAccessSwitchoverActive

Controls the device access privilege of an application.

Only one application is allowed to control the device. This application is able to write into device's registers. Other applications can read device's register only if the controlling application does not have the exclusive privilege.

12.51 GevPrimaryApplicationSocket

Name	GevPrimaryApplicationSocket
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Returns the UDP source port of the primary application.

12.52 GevPrimaryApplicationIPAddress

Name	GevPrimaryApplicationIPAddress
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	≥0

Returns the address of the primary application.

12.53 GevMCPHostPort

Name	GevMCPHostPort
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Controls the port to which the device must send messages. Setting this value to 0 closes the message channel.

12.54 GevMCDA

Name	GevMCDA
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru

Values	≥0

Controls the destination IP address for the message channel.

12.55 GevMCTT

Name	GevMCTT
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	ms
Recommended Visibility	Guru
Values	>0

Provides the transmission timeout value in milliseconds.

12.56 GevMCRC

Name	GevMCRC
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Controls the number of retransmissions allowed when a message channel message times out.

12.57 GevMCSP

Name	GevMCSP
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended	Guru

Visibility	
Values	≥0

This feature indicates the source port for the message channel.

12.58 GevStreamChannelSelector

Name	GevStreamChannelSelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

Selects the stream channel to control.

12.59 GevSCCFGUnconditionalStreaming

Name	GevSCCFGUnconditionalStreaming[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).

12.60 GevSCCFGExtendedChunkData

Name	GevSCCFGExtendedChunkData[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean

Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

Enables cameras to use the extended chunk data payload type for this stream channel.

12.61 GevSCPDirection

Name	GevSCPDirection[GevStreamChannelSelector]
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Guru
Values	Transmitter Receiver

Reports the direction of the stream channel.

12.62 GevSCPInterfaceIndex

Name	GevSCPInterfaceIndex[GevStreamChannelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	0 to 3

Index of network interface to use.

Specific streams might be hard-coded to a specific network interfaces. Therefore this field might not be programmable on certain devices. It is read-only for this case.

12.63 GevSCPHostPort

Name	GevSCPHostPort[GevStreamChannelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream. Setting this value to 0 closes the stream channel.

12.64 GevSCPSFireTestPacket

Name	GevSCPSFireTestPacket[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

Sends a test packet. When this feature is set, the device will fire one test packet.

The “don’t fragment” bit of IP header must be set for this test packet.

12.65 GevSCPSDoNotFragment

Name	GevSCPSDoNotFragment[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended	Guru

Visibility	
Values	True False

The state of this feature is copied into the "do not fragment" bit of IP header of each stream packet. It can be used by the application to prevent IP fragmentation of packets on the stream channel.

12.66 GevSCPSBigEndian

Name	GevSCPSBigEndian[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

Endianess of multi-byte pixel data for this stream.

This is an optional feature. A device that does not support this feature must support little-endian and always leave that bit clear.

12.67 GevSCPSPacketSize

Name	GevSCPSPacketSize[GevStreamChannelSelector]
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	B
Recommended Visibility	Expert
Values	>0

Specifies the stream packet size, in bytes, to send on the selected channel for a GVSP transmitter or specifies the maximum packet size supported by a GVSP receiver.

This does not include data leader and data trailer and the last data packet which might be of smaller size (since packet size is not necessarily a multiple of block size for stream channel).

If a device cannot support the requested packet size, then it must not fire a test packet when requested to do so.

12.68 GevSCPD

Name	GevSCPD[GevStreamChannelSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	
Recommended Visibility	Expert
Values	≥0

Controls the delay (in timestamp counter unit) to insert between each packet for this stream channel. This can be used as a crude flow-control mechanism if the application or the network infrastructure cannot keep up with the packets coming from the device.

12.69 GevSCDA

Name	GevSCDA[GevStreamChannelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

12.70 GevSCSP

Name	GevSCSP[GevStreamChannelSelector]
Level	Optional
Interface	IInteger
Access	Read

Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the source port of the stream channel.

12.71 GevManifestEntrySelector (Deprecated)

Name	GevManifestEntrySelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Invisible
Values	≥1

This feature is deprecated. See section 2.1 Device Control for an equivalent. Selects the manifest entry to reference.

12.72 GevManifestXMLMajorVersion (Deprecated)

Name	GevManifestXMLMajorVersion[GevManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the major version number of the XML file of the selected manifest entry.

12.73 GevManifestXMLMinorVersion (Deprecated)

Name	GevManifestXMLMinorVersion[GevManifestEntrySelector]
Level	Optional

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the minor version number of the XML file of the selected manifest entry.

12.74 GevManifestXMLSubMinorVersion (Deprecated)

Name	GevManifestXMLSubMinorVersion[GevManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the subminor version number of the XML file of the selected manifest entry.

12.75 GevManifestSchemaMajorVersion (Deprecated)

Name	GevManifestSchemaMajorVersion[GevManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the major version number of the schema file of the selected manifest entry.

12.76 GevManifestSchemaMinorVersion (Deprecated)

Name	GevManifestSchemaMinorVersion[GevManifestEntrySelector]
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the minor version number of the schema file of the selected manifest entry.

12.77 GevManifestPrimaryURL (Deprecated)

Name	GevManifestPrimaryURL[GevManifestEntrySelector]
Level	Optional
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	-

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the first URL to the XML device description file of the selected manifest entry.

12.78 GevManifestSecondaryURL (Deprecated)

Name	GevManifestSecondaryURL[GevManifestEntrySelector]
Level	Optional
Interface	IString
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	-

This feature is deprecated. See section 2.1 Device Control for an equivalent.. Indicates the second URL to the XML device description file of the selected manifest entry.

12.79ClConfiguration

Name	ClConfiguration
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Values	Base Medium Full DualBase Deca

This Camera Link specific feature describes the configuration used by the camera. It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitizationTaps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera.

Possible values are:

1 **Base**: Standard base configuration described by the Camera Link standard.

2 **Medium**: Standard medium configuration described by the Camera Link standard.

3 **Full**: Standard full configuration described by the Camera Link standard.

4 **DualBase**: The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two Camera Link base ports. It is responsibility of the application or frame grabber to reconstruct the full image. Only one of the ports (fixed) serves as the "master" for serial communication and triggering.

5 **Deca**: Standard Deca configuration with 10 taps of 8-bit, as described by the Camera Link Standard.

If the feature is omitted, one of the standard configurations (Base-Medium-Full) is expected. In that case the configuration can be unequivocally deduced from the SensorDigitizationTaps and PixelSize values.

12.80ClTimeSlotsCount

Name	ClTimeSlotsCount

Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Values	One Two Three

This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

It indicates the number of consecutive time slots required to transfer one data of each tap.

12.81DeviceTapGeometry

Name	DeviceTapGeometry
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Values	Geometry_1X_1Y Geometry_1X2_1Y Geometry_2X_1Y Geometry_2XE_1Y Geometry_2XM_1Y Geometry_1X_1Y2 Geometry_1X_2YE Geometry_1X3_1Y Geometry_3X_1Y Geometry_1X Geometry_1X2 Geometry_2X Geometry_2XE Geometry_2XM Geometry_1X3 Geometry_3X

	Geometry_1X4_1Y Geometry_4X_1Y Geometry_2X2_1Y Geometry_2X2E_1Y Geometry_2X2M_1Y Geometry_1X2_2YE Geometry_2X_2YE Geometry_2XE_2YE Geometry_2XM_2YE Geometry_1X4 Geometry_4X Geometry_2X2 Geometry_2X2E Geometry_2X2M Geometry_1X8_1Y Geometry_8X_1Y Geometry_4X2_1Y Geometry_2X2E_2YE Geometry_1X8 Geometry_8X Geometry_4X2 …

This Camera Link specific device tap geometry feature describes the geometrical properties characterizing the taps of a Camera Link camera as seen from the frame grabber or acquisition card. Note the case of RGB where even though there are 3 color components, they are considered to be one tap (for instance, Camera Link specification defines 24-bit RGB as a single tap of 24 bits). This feature is mainly applicable to Camera link cameras.

More detailed explanation, including graphical representation for every single Tap configuration is provided in the “Camera Link Tap Geometry Appendix” at the end of this document.

13 User Set Control

This section describes the features for global control of the device settings. It allows loading or saving factory or user-defined settings.

Loading the factory default User Set guarantees a state where a continuous acquisition can be started using only the mandatory features.

13.1 UserSetControl

Name	UserSetControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	-

Category that contains the User Set control features.

13.2 UserSetSelector

Name	UserSetSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Default UserSet1, UserSet2, …

Selects the feature User Set to load, save or configure.

Possible values for **UserSetSelector** are:

- **Default**: Selects the factory setting User set.
- **UserSet1**: Selects the first user set.
- **UserSet2**: Selects the second user set.
- …

When **Default** User Set is selected and loaded using **UserSetLoad**, the device must be in default factory settings state and must make sure the mandatory continuous acquisition use case works directly. Default User Set is read-only and cannot be modified.

13.3 UserSetLoad

Name	UserSetLoad[UserSetSelector]
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Beginner
Values	-

Loads the User Set specified by **UserSetSelector** to the device and makes it active.

13.4 UserSetSave

Name	UserSetSave[UserSetSelector]
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Beginner
Values	-

Save the User Set specified by **UserSetSelector** to the non-volatile memory of the device.

13.5 UserSetDefaultSelector

Name	UserSetDefaultSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner

Values	Default UserSet1, UserSet2, …

Selects the feature User Set to load and make active when the device is reset.

Possible values for **UserSetDefaultSelector** are:

- **Default**: Select the factory setting User set.
- **UserSet1**: Select the first User Set.
- **UserSet2**: Select the second User Set.
- …

If **Default** is selected, the device will boot with the default factory settings and makes sure the mandatory continuous acquisition use case works directly.

14 Chunk Data Control

Chunks are tagged blocks of data. The tags allow a chunk parser to dissect the data payload into its elements and to identify the content.

The length of a frame varies depending on the number of activated chunks, but the user can always expect a frame with the maximum size of **PayloadSize**.

With chunks disabled by setting **ChunkModeActive** to **False** the camera streams frames consisting only of the image.

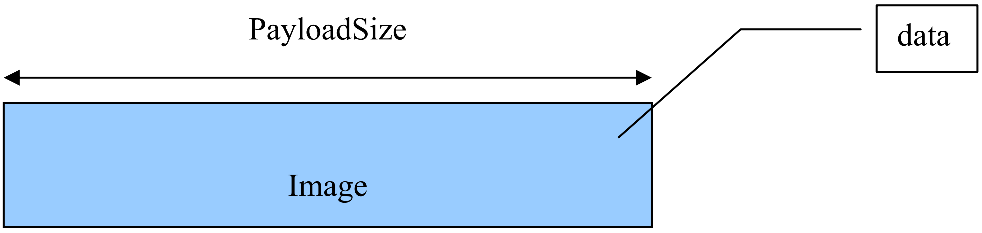

Figure 14-1: Frame with chunks disabled

With chunks enabled by setting **ChunkModeActive** to **True** the camera streams frames consisting of chunks. In this mode the image is a chunk too.

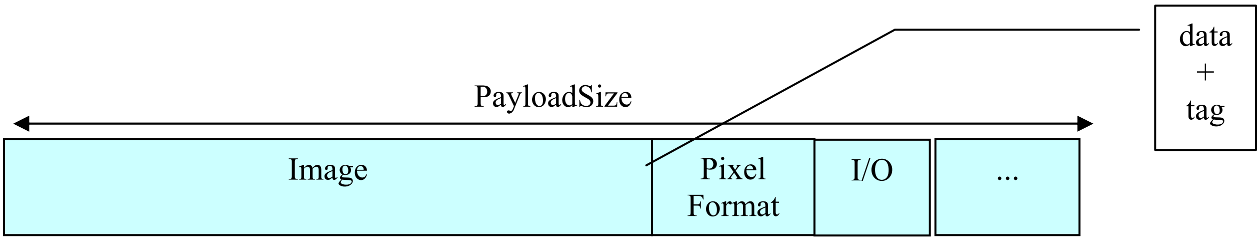

Figure 14-2: Frame with chunks enabled

Each chunk can be enabled or disabled using the **ChunkSelector** and **ChunkEnable** feature. This allows controlling the embedding of different information in the payload.

The data in the chunks is exposed via the chunk parser. The naming scheme to access the data of the chunk *name* is **Chunk***name*.

14.1 ChunkDataControl

Name	ChunkDataControl
Level	Recommended

Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains the Chunk Data control features.

14.2 ChunkModeActive

Name	ChunkModeActive
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Activates the inclusion of Chunk data in the payload of the image.

14.3 ChunkSelector

Name	ChunkSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Image OffsetX OffsetY Width Height PixelFormat DynamicRangeMax (deprecated) DynamicRangeMin (deprecated)

	PixelDynamicRangeMax PixelDynamicRangeMinTimestamp LineStatusAll CounterValue TimerValue ExposureTime Gain BlackLevel LinePitch FrameID

Selects which Chunk to enable or control.

14.4 ChunkEnable

Name	ChunkEnable[ChunkSelector]
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Enables the inclusion of the selected Chunk data in the payload of the image.

14.5 ChunkImage

Name	ChunkImage
Level	Recommended
Interface	IRegister
Access	Read
Unit	-
Recommended Visibility	Guru
Values	Device-specific

Returns the entire image data included in the payload.

14.6 ChunkOffsetX

Name	ChunkOffsetX
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Returns the **OffsetX** of the image included in the payload.

14.7 ChunkOffsetY

Name	ChunkOffsetY
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Returns the **OffsetY** of the image included in the payload.

14.8 ChunkWidth

Name	ChunkWidth
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Returns the **Width** of the image included in the payload.

14.9 ChunkHeight

Name	ChunkHeight
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	>0

Returns the **Height** of the image included in the payload.

14.10 ChunkPixelFormat

Name	ChunkPixelFormat
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Expert
Values	Mono8 Mono8Signed Mono10 Mono10Packed Mono12 Mono12Packed Mono14 Mono16 BayerGR8 BayerRG8 BayerGB8 BayerBG8 BayerGR10 BayerRG10 BayerGB10 BayerBG10 BayerGR12 BayerRG12 BayerGB12 BayerBG12

	BayerGR10Packed BayerRG10Packed BayerGB10Packed BayerBG10Packed BayerGR12Packed BayerRG12Packed BayerGB12Packed BayerBG12Packed BayerGR16 BayerRG16 BayerGB16 BayerBG16 RGB8Packed BGR8Packed RGBA8Packed BGRA8Packed RGB10Packed BGR10Packed RGB12Packed BGR12Packed RGB16Packed BGR16Packed RGB10V1Packed RGB10V2Packed BGR10V1Packed BGR10V2Packed RGB12V1Packed RGB565Packed BGR565Packed YUV411Packed YUV422Packed YUV444Packed YUYVPacked RGB8Planar RGB10Planar RGB12Planar RGB16Planar Device-specific

Returns the **PixelFormat** of the image included in the payload.

14.11 ChunkPixelDynamicRangeMin

Name	ChunkPixelDynamicRangeMin
Level	Recommended

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Returns the minimum value of dynamic range of the image included in the payload.

14.12 ChunkPixelDynamicRangeMax

Name	ChunkPixelDynamicRangeMax
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Returns the maximum value of dynamic range of the image included in the payload.

14.13 ChunkDynamicRangeMin (Deprecated)

Name	ChunkDynamicRangeMin
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. ChunkPixelDynamicRangeMin should be used instead. Returns the minimum value of dynamic range of the image included in the payload.

14.14 ChunkDynamicRangeMax (Deprecated)

Name	ChunkDynamicRangeMax
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. ChunkPixelDynamicRangeMax should be used instead. Returns the maximum value of dynamic range of the image included in the payload.

14.15 ChunkTimestamp

Name	ChunkTimestamp
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event.

See Figure 5-2.

14.16 ChunkLineStatusAll

Name	ChunkLineStatusAll
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert

Values	≥0

Returns the status of all the I/O lines at the time of the FrameStart internal event.

14.17 ChunkCounterSelector

Name	ChunkCounterSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Counter1, Counter2,…

Selects which counter to retrieve data from.

CounterSelector can take any of the following values:

- **Counter1**: Selects the first counter.
- **Counter2**: Selects the second counter.
- …

14.18 ChunkCounterValue

Name	ChunkCounterValue[ChunkCounterSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

Returns the value of the selected Chunk counter at the time of the FrameStart event.

14.19 See Figure 5-2.ChunkCounter (Deprecated)

Name	ChunkCounter[ChunkCounterSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Invisible
Values	≥0

This feature is deprecated. Returns the value of the selected Chunk counter at the time of the FrameStart internal event.

See Figure 5-2.

14.20 ChunkTimerSelector

Name	ChunkTimerSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Timer1, Timer2, …

Selects which Timer to retrieve data from.

TimerSelector can take any of the following values:

- **Timer1:** Selects the first Timer.
- **Timer2:** Selects the second Timer.
- …

14.21 ChunkTimerValue

Name	ChunkTimerValue[ChunkTimerSelector]
Level	Recommended

Interface	IFloat
Access	Read
Unit	us
Recommended Visibility	Expert
Values	>0

Returns the value of the selected Timer at the time of the FrameStart internal event.

14.22 ChunkTimer (Deprecated)

Name	ChunkTimer[ChunkTimerSelector]
Level	Recommended
Interface	IFloat
Access	Read
Unit	us
Recommended Visibility	Invisible
Values	>0

This feature is deprecated. Returns the value of the selected Timer at the time of the FrameStart internal event.

See Figure 5-2.

14.23 ChunkExposureTime

Name	ChunkExposureTime
Level	Recommended
Interface	IFloat
Access	Read
Unit	us
Recommended Visibility	Expert
Values	≥0

Returns the exposure time used to capture the image.

14.24 ChunkGainSelector

Name	ChunkGainSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	All Red Green Blue Y U V Tap1, Tap2, … AnalogAll AnalogRed AnalogGreen AnalogBlue AnalogY AnalogU AnalogV AnalogTap1, AnalogTap2, … DigitalAll DigitalRed DigitalGreen DigitalBlue DigitalY DigitalU DigitalV DigitalTap1, DigitalTap2, …

Selects which Gain to retrieve data from.

In general, there are 2 types of gain that can exist in a camera, analog or digital. Some camera will implement one or other or both. This is why there are 3 possible sets of gain.

The first one, without the **Analog** or **Digital** prefix, is to be used when only one type of gain is implemented. This permits to have an implementation independent way to set the gain.

The second and the third, with the **Analog** and **Digital** prefix, is to be used when both types of gain are implemented. This permits to have independent control over each one.

The possible values for **GainSelector** are:

- **All**: Gain will be applied to all channels or taps.
- **Red**: Gain will be applied to the red channel.
- **Green**: Gain will be applied to the green channel.
- **Blue**: Gain will be applied to the blue channel.
- **Y**: Gain will be applied to Y channel.
- **U**: Gain will be applied to U channel.
- **V**: Gain will be applied to V channel.
- **Tap1**: Gain will be applied to Tap 1.
- **Tap2**: Gain will be applied to Tap 2.
- …
- **AnalogAll**: Gain will be applied to all analog channels or taps.
- **AnalogRed**: Gain will be applied to the red analog channel.
- **AnalogGreen**: Gain will be applied to the green analog channel.
- **AnalogBlue**: Gain will be applied to the blue analog channel.
- **AnalogY**: Gain will be applied to Y analog channel.
- **AnalogU**: Gain will be applied to U analog channel.
- **AnalogV**: Gain will be applied to V analog channel.
- **AnalogTap1**: Analog gain will be applied to Tap 1.
- **AnalogTap2**: Analog gain will be applied to Tap 2.
- …
- **DigitalAll**: Gain will be applied to all digital channels or taps.
- **DigitalRed**: Gain will be applied to the red digital channel.
- **DigitalGreen**: Gain will be applied to the green digital channel.
- **DigitalBlue**: Gain will be applied to the blue digital channel.
- **DigitalY**: Gain will be applied to Y digital channel.
- **DigitalU**: Gain will be applied to U digital channel.

- **DigitalV**: Gain will be applied to V digital channel.
- **DigitalTap1**: Digital gain will be applied to Tap 1.
- **DigitalTap2**: Digital gain will be applied to Tap 2.
- …

14.25 ChunkGain

Name	ChunkGain[ChunkGainSelector]
Level	Recommended
Interface	IFloat
Access	Read
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Returns the gain used to capture the image.

14.26 ChunkBlackLevelSelector

Name	ChunkBlackLevelSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	All Red Green Blue Y U V Tap1, Tap2, …

Selects which Black Level to retrieve data from.

The possible values for **BlackLevelSelector** are:

- **All**: Black Level will be applied to all channels or taps.
- **Red**: Black Level will be applied to the red channel.
- **Green**: Black Level will be applied to the green channel.
- **Blue**: Black Level will be applied to the blue channel.
- **Y**: Black Level will be applied to Y channel.
- **U**: Black Level will be applied to U channel.
- **V**: Black Level will be applied to V channel.
- **Tap1**: Black Level will be applied to Tap 1.
- **Tap2**: Black Level will be applied to Tap 2.
- …

14.27 ChunkBlackLevel

Name	ChunkBlackLevel[ChunkBlackLevelSelector]
Level	Recommended
Interface	IFloat
Access	Read
Unit	-
Recommended Visibility	Expert
Values	Device-specific

Returns the black level used to capture the image included in the payload.

14.28 ChunkLinePitch

Name	ChunkLinePitch
Level	Recommended
Interface	IInteger
Access	Read
Unit	B
Recommended Visibility	Expert
Values	>0

Returns the **LinePitch** of the image included in the payload.

14.29 ChunkFrameID

Name	ChunkFrameID
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	

Returns the unique Identifier of the frame (or image) included in the payload.

Recommended behavior of the identifier: It should start at a certain minimum value and keep incrementing by one for each frame up to a maximum, then it wraps to the minimum again. Each streaming channel should maintain the Frame ID separately.

15 File Access Control

The File Access Controls section describes all features related to accessing files in the device.

It contains the definition of a generic file access schema for GenICam compliant devices. It is based on a set of standard features that are controlled from adapter code which resides in the GenICam reference implementation. The adapter code presents its services through an interface inherited from std::iostream.

The model, on which the controls are based, is depicted in the following diagram:

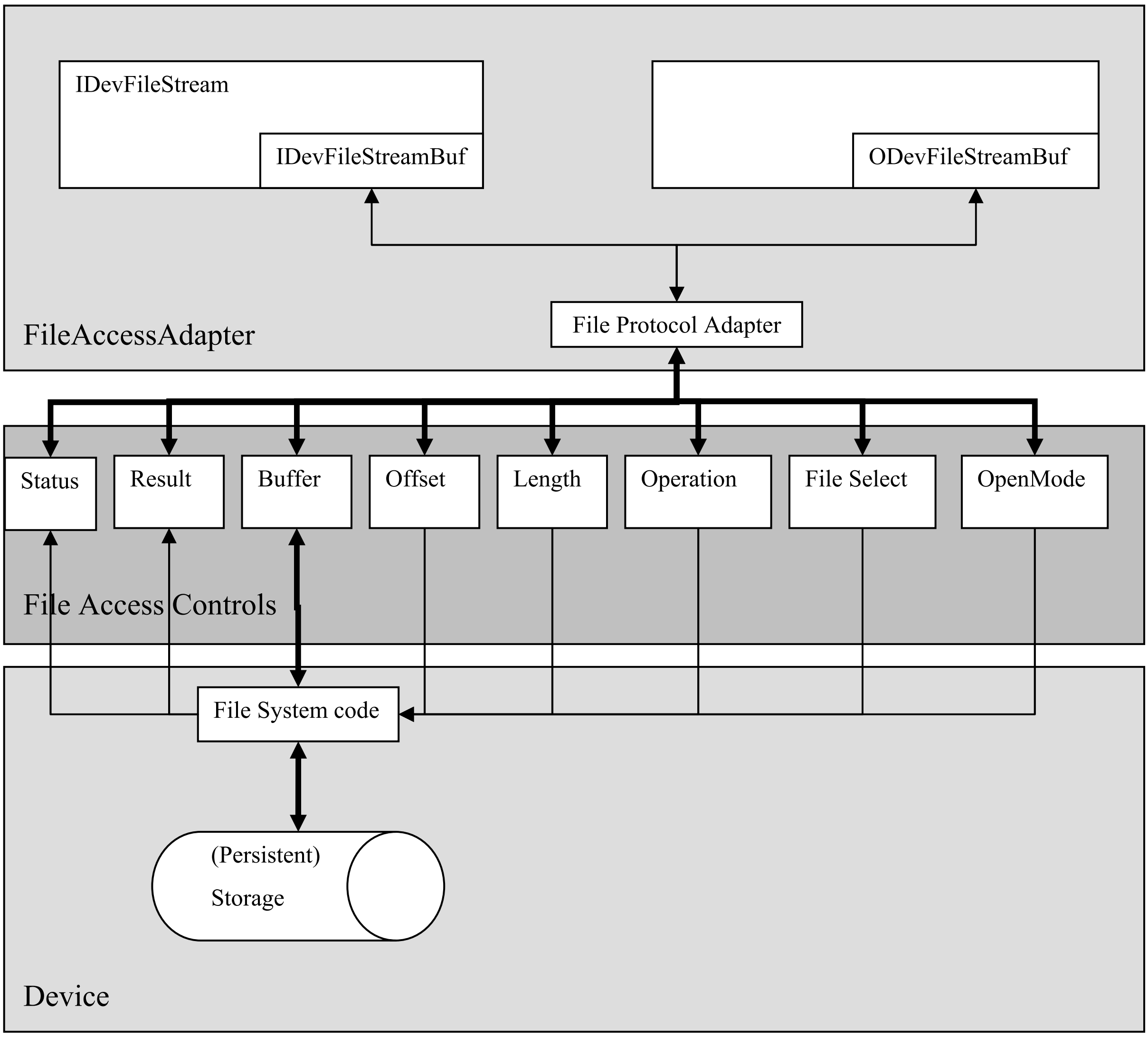

Figure 14-1: File Access Model

It assumes that all operations, which can be done on the persistent storage, could be executed by using operations with the semantic of fopen/fclose/fread/fwrite. The operations and their parameters are mapped onto the features of the list of File Access Controls.

To provide a generic API on top of the File Access Controls, a FileAccessAdapter is defined in the GenApi. T he Adapter provides two iostream interfaces to the device files:

- **IDevFileStream** Read from the device
- **ODevFileStream** Write to the device

The File Protocol Adapter is responsible for the mapping of the (I/O) DevFileStreamBuf actions Open, Close, UnderFlow, Overflow on File Access Controls

Example Code for the streaminterface:

```
//GenApi::INodeMap * pInterface
ODevFileStream usersetWrite;
usersetWrite.open(pInterface, "UserSet1");
if( ! usersetWrite.fail() ){
     usersetWrite << "Hello World\n";
}
usersetWrite.close();

IDevFileStream usersetRead;
usersetRead.open(pInterface, "UserSet1");
if( ! usersetRead.fail() ){
     cout << usersetRead.rdbuf();
}
usersetRead.close();
```

File Access Control:

The **FileSelector** feature selects the target file in the device for the Operation. The entries of this enumeration define the names of all files in the device that can be accessed via the File Access.

FileOperationSelector specifies the operation to execute on the file.

FileOperationExecute command starts the selected operation execution.

FileOpenMode is a parameter for the Open operation and controls the access mode (Read, Write, ReadWrite) in which the file is opened.

FileOperationStatus returns the status of the last operation executed on the file. This feature must return Success if the operation is executed as requested.

FileOperationResult returns the number of bytes successfully read/written bytes during the previous Read or Write operations.

FileSize returns the size of the file in bytes.

The data, that is read from or written to the device, is exchanged between the application and the device through the **FileAccessBuffer** feature. This register mapped **FileAccessBuffer** must be written with the target data before to execute the Write operation using **FileOperationExecute**. For Read operation, the data can be read from the **FileAccessBuffer** after the Read operation has been executed.

FileAccessOffset controls the starting position of the access in the file**.**

FileAccessLength controls the number of bytes to transfer to or from the **FileAccessBuffer** during the following Read or Write operation.

Altogether, the features **FileSelector**, **FileAccessOffset** and **FileAccessLength** control the mapping between the device file storage and the **FileAccessBuffer**.

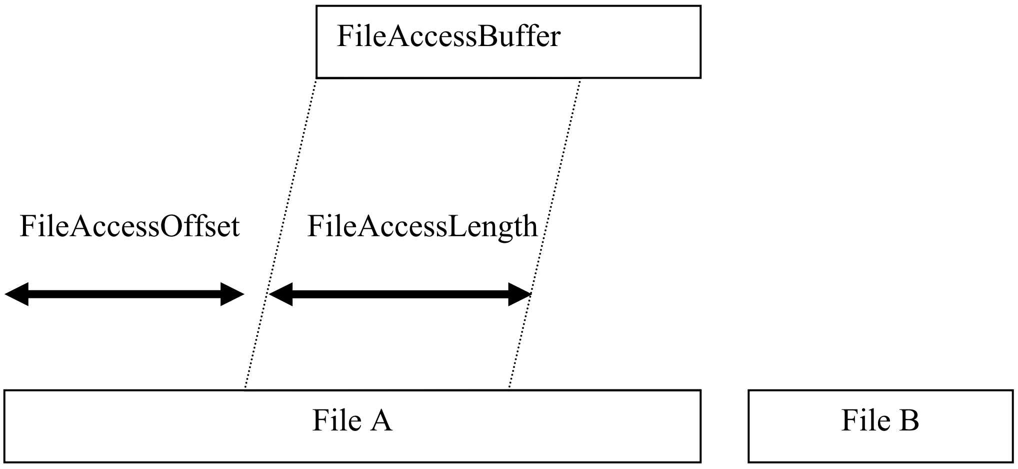

Figure 14-2: Layout of FileAccessBuffer (FileA is selected by FileSelector)

15.1 FileAccessControl

Name	FileAccessControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

Category that contains the File Access control features.

15.2 FileSelector

Name	FileSelector
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Guru
Values	UserSetDefault UserSet1 UserSet2 UserSet3 … LUTLuminance LUTRed LUTGreen LUTBlue …

Selects the target file in the device.

The entries of this enumeration define the names of all files in the device that can be accessed via the File access.

FileSelector can take any of the following values:

- **UserSetDefault**: the default user set of the device
- **UserSet1**: the first user set of the device
- **UserSet2**: the second user set of the device
- **UserSet3**: the third user set of the device
- …
- **LUTLuminance**: The Luminance LUT of the camera.
- **LUTRed**: The Red LUT of the camera.
- **LUTGreen**: The Green LUT of the camera.
- **LUTBlue**: The Blue LUT of the camera.
- …

On top of the previous standard values, a device might also provide device-specific values.

15.3 FileOperationSelector

Name	FileOperationSelector[FileSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	Open Close Read Write

Selects the target operation for the selected file in the device. This Operation is executed when the **FileOperationExecute** feature is called.

FileOperationSelector can take any of the following values:

- **Open**: Opens the file selected by **FileSelector** in the device. The access mode in which the file is opened is selected by **FileOpenMode**
- **Close**: Closes the file selected by **FileSelector** in the device.
- **Read**: Reads **FileAccessLength** bytes from the device storage at the file relative offset **FileAccessOffset** into **FileAccessBuffer.**
- **Write:** Writes **FileAccessLength** bytes taken from the **FileAccessBuffer** into the device storage at the file relative offset **FileAccessOffset.**

15.4 FileOperationExecute

Name	FileOperationExecute[FileSelector][FileOperationSelector]
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Recommended Visibility	Guru
Values	-

Executes the operation selected by **FileOperationSelector** on the selected file.

15.5 FileOpenMode

Name	FileOpenMode[FileSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Guru
Values	Read Write ReadWrite

Selects the access mode in which a file is opened in the device.

FileOpenMode can take any of the following values:

- **Read:** This mode selects read-only open mode.
- **Write:** This mode selects write-only open mode.
- **ReadWrite:** This mode selects read and write open mode.

15.6 FileAccessBuffer

Name	FileAccessBuffer
Level	Recommended

Interface	IRegister
Access	Read/(Write)
Unit	-
Recommended Visibility	Guru
Values	Device-specific

Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.

This register mapped **FileAccessBuffer** must be written with the target data before to execute a Write operation. For Read Operation, the data can be read from the **FileAccessBuffer** after the Read operation has been executed. The effective data transfer is done upon **FileOperationExecute** execution (See Figure 14-2).

15.7 FileAccessOffset

Name	FileAccessOffset[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	B
Recommended Visibility	Guru
Values	>= 0

Controls the Offset of the mapping between the device file storage and the **FileAccessBuffer**.

The **FileAccessOffset** defines the offset in bytes of the **FileAccessBuffer** relative to the beginning of the selected File (See Figure 14-2). This feature is available only when **FileOperationSelector** is set to Read or Write.

15.8 FileAccessLength

Name	FileAccessLength[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	B

Recommended Visibility	Guru
Values	>= 0

Controls the Length of the mapping between the device file storage and the **FileAccessBuffer.**

The **FileAccessLength** defines the number of bytes to transfer to or from the **FileAccessBuffer** (See Figure 14-2). This feature is available only when **FileOperationSelector** is set to Read or Write.

15.9 FileOperationStatus

Name	FileOperationStatus[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Guru
Values	Success (mandatory) Failure …

Represents the file operation execution status.

Upon execution of a successful file operation, it must return **Success**. In case of complete or partial failure of the operation, other return values can be defined to indicate the nature of the error that happened. If only one fail status is defined, it should be defined as **Failure**.

15.10 FileOperationResult

Name	FileOperationResult[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru

Values	-

Represents the file operation result. For Read or Write operations, the number of successfully read/written bytes is returned.

15.11 FileSize

Name	FileSize[FileSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	B
Recommended Visibility	Guru
Values	>=0

Represents the size of the selected file in bytes.

16 Color Transformation Control

The Color Transformation section describes all features related to color Transformations in the device.

The Color Transformation is a linear operation taking as input a triplet of Components (C0, C1, C2) for a color pixel (Typicaly: R_{in}, G_{in}, B_{in} representing a RGB color pixel). This triplet is first multiplied by a 3x3 matrix and then added to an offset triplet.

The equation is given by:

$$\begin{pmatrix} C0_{out} \\ C1_{out} \\ C2_{out} \end{pmatrix} = \begin{pmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{pmatrix} \begin{pmatrix} C0_{in} \\ C1_{in} \\ C2_{in} \end{pmatrix} + \begin{pmatrix} Offset0 \\ Offset1 \\ Offset2 \end{pmatrix}$$

$$\text{Equivalent: } \begin{pmatrix} R_{out} \\ G_{out} \\ B_{out} \end{pmatrix} = \begin{pmatrix} RR & RG & RB \\ GR & GG & GB \\ BR & BG & BB \end{pmatrix} \begin{pmatrix} R_{in} \\ G_{in} \\ B_{in} \end{pmatrix} + \begin{pmatrix} R_{offset} \\ G_{offset} \\ B_{offset} \end{pmatrix}$$

The descriptions below assume RGB to RGB transformation:

Where	$C0_{in}$ is the first component of the incoming pixel
	$C1_{in}$ is the second component of the incoming pixel
	$C2_{in}$ is the third component of the incoming pixel
	Gain00 is the red contribution to the red pixel (multiplicative factor)
	Gain01 is the green contribution to the red pixel (multiplicative factor)
	Gain02 is the blue contribution to the red pixel (multiplicative factor)
	Gain10 is the red contribution to the green pixel (multiplicative factor)
	Gain11 is the green contribution to the green pixel (multiplicative factor)
	Gain12 is the blue contribution to the blue pixel (multiplicative factor)
	Gain20 is the red contribution to the blue pixel (multiplicative factor)
	Gain21 is the green contribution to the blue pixel (multiplicative factor)
	Gain22 is the blue contribution to the blue pixel (multiplicative factor)
	Offset0 is the red offset
	Offset1 is the green offset
	Offset2 is the blue offset
	$C0_{out}$ is the first resulting component of the pixel after the transformation

	$C1_{out}$ is the second resulting component of the pixel after the transformation
	$C2_{out}$ is the third resulting component of the pixel after the transformation

Example for YUV convertion:

The Color Transformation can also be used outside of the simple scope of color correction on RGB pixels. For instance, it can be used as a color convert to convert RGB to YUV.

Here is the example of this conversion for 8-bit pixels:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R_{in} \\ G_{in} \\ B_{in} \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

16.1 ColorTransformationControl

Name	ColorTransformationControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Expert
Values	-

Category that contains the Color Transformation control features.

16.2 ColorTransformationSelector

Name	ColorTransformationSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	RGBtoRGB RGBtoYUV

	Device-specific

Selects which Color Transformation module is controlled by the various Color Transformation features.

It is typically not available when a single Color Transformation module is supported.

16.3 ColorTransformationEnable

Name	ColorTransformationEnable[ColorTransformationSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

Activates the selected Color Transformation module.

16.4 ColorTransformationValueSelector

Name	ColorTransformationValueSelector[ColorTransformationSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Gain00 Gain01 Gain02 Gain10 Gain11 Gain12 Gain20 Gain21 Gain22 Offset0

	Offset1 Offset2

Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module.

16.5 ColorTransformationValue

Name	ColorTransformationValue[ColorTransformationSelector] [ColorTransformationValueSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-Specific

Represents the value of the selected Gain factor or Offset inside the Transformation matrix.

17 Action Control

The Action section describes all features related to Action Signals in the device.

Action Signals are a method to trigger actions in multiple devices at the same time (depending on the specific transport layer). Action Signals are used in the device in the same way as e.g. digital input lines.

One possible use for action signals is to raise a FrameStart trigger in multiple devices at the same time.

On most transport layers Action Signals are implemented using broadcast protocol messages. To allow a finegrained control which devices are allowed to react on the broadcasted action protocol message, the features **ActionDeviceKey**, **ActionGroupKey** and **ActionGroupMask** define filter conditions.

Each action protocol message contains an action device key, action group key and an action group mask. If the device detects a match between this protocol information and one of the actions selected by **ActionSelector** it raises the corresponding Action Signal.

See the chapter “Typical Standard Feature Usage Examples” for typical use cases.

17.1 ActionControl

Name	ActionControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

Category that contains the Action control features.

17.2 ActionDeviceKey

Name	ActionDeviceKey
Level	Optional
Interface	IInteger
Access	Write-Only
Unit	-

Recommended Visibility	Guru
Values	>= 0

Provides the device key that allows the device to check the validity of action commands. The device internal assertion of an action signal is only authorized if the **ActionDeviceKey** and the action device key value in the protocol message are equal.

17.3 ActionSelector

Name	ActionSelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	>0

Selects to which Action Signal further Action settings apply.

17.4 ActionGroupMask

Name	ActionGroupMask[ActionSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Provides the mask that the device will use to validate the action on reception of the action protocol message.

The device asserts the selected Action signal only if :

- the selected **ActionDeviceKey** is equal to the action device key in the action protocol message,

- the logical AND-wise operation of the action group mask in the action protocol message against the selected **ActionGroupMask** is non-zero
- and the selected **ActionGroupKey** is equal to the action group key in the action protocol message.

17.5 ActionGroupKey

Name	ActionGroupKey[ActionSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Provides the key that the device will use to validate the action on reception of the action protocol message.

The device asserts the selected Action signal only if :

- the selected **ActionDeviceKey** is equal to the action device key in the action protocol message,
- the logical AND-wise operation of the action group mask in the action protocol message against the selected **ActionGroupMask** is non-zero
- and the selected **ActionGroupKey** is equal to the action group key in the action protocol message.

18 Typical Standard Feature Usage Examples

This section shows examples of typical use cases of the standard acquisition features in C/C++ pseudo-code.

For simplicity, the object name is omitted (e.g., **AcquisitionStart**() instead of **Camera.AcquisitionStart())** and the default state of the camera is assumed (e.g. Ready for a continuous acquisition start without trigger).

18.1 Acquisition and Trigger Examples

/* Continuous acquisition when the camera is in its reset state. */

```
AcquisitionMode = Continuous;
AcquisitionStart();
…
AcquisitionStop();
```

/* Single Frame acquisition in Hardware trigger mode using the external I/O Line 3. */

```
AcquisitionMode    = SingleFrame;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line3;
AcquisitionStart();
```

/* Multi-Frames acquisition started by a single Software trigger delayed by 1 millisecond. The Trigger starts the whole sequence acquisition. The Exposure time for each frame is set to 500 us.

*/

```
AcquisitionMode       = MultiFrame;
AcquisitionFrameCount = 20;
TriggerSelector       = AcquisitionStart;
TriggerMode           = On;
TriggerSource         = Software;
TriggerDelay          = 1000;
ExposureMode          = Timed;
ExposureTime          = 500;
AcquisitionStart();
TriggerSoftware();
```

/* Continuous acquisition in Hardware trigger mode. The Frame triggers are Rising Edge signals coming from the physical Line 2. The Exposure time is 500us. An exposure end event is also sent to the Host application after the exposure of each frame to signal that the inspected part can be moved. The timestamp of the event is also read.

*/

```
AcquisitionMode    = Continuous;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line2;
ExposureMode       = Timed;
ExposureTime       = 500;
Register(Camera.EventExposureEnd, CallbackDataObject, CallbackFunctionPtr)
EventSelector      = ExposureEnd;
EventNotification  = On;
AcquisitionStart();
…
// In the callback of the ExposureEnd event, get the event timestamp:
Timestamp = EventExposureEndTimestamp;
…
AcquisitionStop();
```

/* Multi-Frames acquisition with each frame triggered by a Hardware trigger on Line 1. A negative pulse of the exposure signal duration (500us) is also sent to the physical output line 2 to activate a light during the exposure time of each frame. The end of the sequence capture is signalled to the host with an acquisition end event.
*/

```
AcquisitionMode        = MultiFrame;
AcquisitionFrameCount = 20;
TriggerSelector        = FrameStart;
TriggerMode            = On;
TriggerActivation      = RisingEdge;
TriggerSource          = Line1;
ExposureMode           = Timed;
ExposureTime           = 500;
LineSelector           = Line2;
LineMode               = Output;
LineInverter           = True;
LineSource             = ExposureActive
Register(Camera.EventAcquisitionEnd,CallbackDataObject,CallbackFunctionPtr)
EventSelector          = AcquisitionEnd;
EventNotification      = On;
AcquisitionStart();
```

/* Line Scan continuous acquisition with Hardware Frame and Line trigger. */

```
AcquisitionMode     = Continuous;
TriggerSelector     = FrameStart;
TriggerMode         = On;
TriggerActivation   = RisingEdge;
TriggerSource       = Line1;
TriggerSelector     = LineStart;
TriggerMode         = On;
TriggerActivation   = RisingEdge;
TriggerSource       = Line2;
AcquisitionStart();
...
AcquisitionStop();
```

/* Frame Scan continuous acquisition with Hardware Frame trigger and the
Exposure duration controlled by the Trigger pulse width.
*/

```
AcquisitionMode     = Continuous;
TriggerSelector     = FrameStart;
TriggerMode         = On;
TriggerActivation   = RisingEdge;
TriggerSource       = Line1;
ExposureMode        = TriggerWidth;
AcquisitionStart();
...
AcquisitionStop();
```

/* Frame Scan continuous acquisition with 1 Hardware trigger controlling
the start of the acquisition and 2 others harware triggers to start and stop
the exposure of each frame.

*/

```
AcquisitionMode = Continuous;
TriggerSelector = AcquistionStart;
TriggerMode     = On;
TriggerSource   = Line1;
ExposureMode    = TriggerControlled;
TriggerSelector = ExposureStart;
TriggerMode     = On;
TriggerSource   = Line3;
TriggerSelector = ExposureStop;
TriggerMode     = On;
TriggerSource   = Line4;
AcquisitionStart();
...
AcquisitionStop();
```

18.2 Counter and Timer Examples

/* Counts the number of rising edge triggers signals received on Line1 and the number of Frame Start events in Hardware triggered Continuous acquisition to verify that no trigger were missed.

*/

```
AcquisitionMode    = Continuous;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line1;

CounterSelector    = Counter1;
CounterEventSource = Line1;
CounterEventActivation = RisingEdge;
CounterReset();

CounterSelector    = Counter2;
CounterEventSource = FrameStart;
CounterReset();

AcquisitionStart();
…
AcquisitionStop();

CounterSelector    = Counter1;
NbTrigger         = CounterValue;
CounterSelector    = Counter2;
NbFrameStart       = CounterValue;
if (NbTrigger != NbFrameStart)
    printf(“Error ! Trigger missed.”);
```

/* Use a counter to generate an event at line 200 of each captured Frame
in a continuous acquisition.
*/

```
AcquisitionMode       = Continuous;
CounterSelector       = Counter1;
CounterEventSource    = LineStart;
CounterDuration       = 200;
CounterTriggerSource = FrameStart;
CounterResetSource    = CounterTrigger;
Register(Camera.EventCounter1End,CallbackDataObject,CallbackFunctionPtr)
EventSelector         = Counter1End;
EventNotification     = On;
AcquisitionStart();
…
AcquisitionStop();
```

/* Generate a 200us Timer pulse (Strobe) delayed by 100 us on the physical output Line 2.
The Timer pulse is started using a trigger coming from physical input Line 1 .
*/

```
TimerSelector           = Timer1;
TimerDuration           = 200;
TimerDelay              = 100;
TimerTriggerSource      = Line1;
TimerTriggerActivation = RisingEdge;
LineSelector            = Line2;
LineMode                = Output;
LineSource              = Timer1Active;
```

/* Use of a Timer to measure the length in microseconds of a negative pulse on the
physical input Line1. An Event is also generated to the host application to signal
the end of the pulse.
*/

```
TimerSelector          = Timer1;
TimerTrigger           = Line1;
TimerTriggerActivation = LevelLow;
Register(Camera.EventLine1RisingEdge,CallbackDataObject,CallbackFunctionPtr)
EventSelector          = Line1RisingEdge;
EventNotifications     = On;
/* Wait for the event on the host to read the time. */
…
TimerSelector          = Timer1;
PulseDuration          = TimerValue;
```

18.3 I/O Examples

/* Read the inverted Status of the physical Line 1. */

```
LineSelector  = Line1;
LineMode      = Input;
LineInverter  = True;
CurrentState  = LineStatus;
```

/* Output of the Exposure signal of each frame on the physical Line 2. */

```
LineSelector = Line2;
LineMode     = Output;
LineSource   = ExposureActive;
```

/* User Output of a positive TTL signal on physical Line 2. */

```
LineSelector        = Line2;
LineMode            = Output;
LineFormat          = TTL;
LineSource          = UserOutput2;
UserOutputSelector  = UserOutput2;
UserOutputValue     = True;
```

18.4 Action Signal Examples

/* Triggered Single Frame acquisition using the Action Signal 1. */

```
AcquisitionMode     = SingleFrame;
TriggerSelector     = FrameStart;
TriggerMode         = On;
TriggerSource       = Action1;

ActionDeviceKey     = 0x12345678;
ActionSelector      = Action1
ActionGroupKey      = 0x1
ActionGroupMask     = 0x1

AcquisitionStart();
```

/* Generate a 200us Timer pulse (Strobe) on the physical output Line 2.
   The Timer pulse is started using a trigger coming from Action Signal 3.
*/

```
TimerSelector           = Timer1;
TimerDuration           = 200;
TimerTriggerSource      = Action3;

LineSelector            = Line2;
LineMode                = Output;
LineSource              = Timer1Active;

ActionDeviceKey         = 0x12345678;
ActionSelector          = Action3
ActionGroupKey          = 0x1
ActionGroupMask         = 0x7
```

19 Acknowledgements

The following companies have participated in the elaboration of the Standard Features Naming Convention:

Company	Represented by
SVS-Vistek	Gerd Reichle
Stemmer Imaging	Sascha Dorenbeck, Rupert Stelz, Martin Kersting
SICK IVP	Mattias Johannesson
Pleora Technologies	Francois Gobeil, Vincent Rowley
National Instruments	Chris Graf, Tony Iglesias, Johann Scholtz, Eric Gross
MVTec Software	Christoph Zierl
Matrox Imaging	Stephane Maurice
Leutron Vision	Jan Becvar, Stefan Thommen
JAI	Tue Moerk, Michael Krag, Ole Krogh Joergensen, Karsten Ingeman Christensen
Euresys	Jean-Michel Wintgens, Yves Joskin
e2v semiconductors	Frédéric Mathieu
DALSA	Peifang Zhou
DALSA Coreco	Eric Carey
Basler	Fritz Dierks, Alex Happe, Andreas Gaer, Thies Möller, Sven Seeger
AIA	Jeff Fryman

20 Camera Link Tap Geometry Appendix

TABLE OF FIGURES :

20.1 Motivations

The initial release of Camera Link® standard didn't include any information about the geometrical properties of taps.

Frame grabbers are able to reconstruct on-the-fly the image from multi-tap cameras.

Camera manufacturer should clearly specify what *geometry*(ies) is (are) required. Frame grabber manufacturer should also clearly specify what geometry(ies) is (are) supported.

The customer will easily check the compatibility, and select the appropriate geometry for the camera and the frame grabber.

Considering the limited amount of cases, a unique name is assigned to each geometry.

20.2 Identifying the geometrical properties

20.2.1 Image geometrical properties

The relevant geometrical properties required for reconstructing the image:

- **Vantage point**: an enumerated type of parameter that specifies the position of the pixel with coordinate X=1, Y=1 in the scene.
  {Top-Left, Top-Right, Bottom-Left, Bottom-Right}
  Default is Top-Left.
- **ImageWidth**: an integer type of parameter declaring the image width expressed in pixels.
- **ImageHeight**: an integer type of parameter declaring the image height expressed in pixels. This parameter is irrelevant in case of line-scan or tdi cameras.
- **TapGeometry**: an enumerated type of parameter that summarize following properties for each tap:
  - XStart : X-coordinate of the first pixel column
  - YStart : Y-coordinate if the first pixel row
  - XEnd : X-coordinate of the last pixel column
  - YEnd : Y-coordinate of the last pixel row
  - XStep : difference of X-coordinates between consecutive pixel columns; X-step is positive when X-coordinates are increasing along a row; it is negative otherwise.
  - YStep : difference of Y-coordinates between consecutive pixel rows; Y-step is positive when Y-coordinates is increasing at the end of a line; it is negative otherwise.

- Allocation of taps to ports. The camera taps are indexed using following conventional sorting rule:
  First by increasing values of YStart then by increasing value of XStart. The tap T1 is the camera tap that exhibits the smallest XStart for the smallest YStart.

20.2.1.1 Restrictions

All zones have the same size.

Zones are not overlapping.

All zones have the same number of taps.

All taps are carrying the same amount of pixels.

20.2.1.2 Naming convention

A tap configuration for **area-scan** camera is designated by:

<TapGeometryX>**-**<TapGeometryY>

A tap configuration for **line-scan or tdi-line-scan** camera is designated by:

<TapGeometryX>

TapGeometryX is designated by <ZoneX>**X**(<TapX>)(<ExtX)>)

<ZoneX>: an integer in range {1, 2, 3, 4, 8} declaring number of zones encountered across horizontal direction.

<TapX>: an integer in range {∅, 2, 3, 4, 8} declaring the number of consecutive pixels across horizontal direction that are output simultaneously from a zone. This field is omitted when all pixels are in the same column.

<ExtX>: a letter in range {∅, E, M} declaring the location of the pixels extractors across horizontal direction. Value E indicates that pixel extractors are at both ends of the line. Value M indicates that pixel extractors are at middle of the line. This field is omitted when all pixel extractors are all at the left of each zone.

TapGeometryY is designated by <ZoneY>Y(<TapY>)(<ExtY)>)

<ZoneY>: an integer in range {1, 2} declaring number of zones encountered across vertical direction.

<TapY>: an integer in range {∅, 2} declaring the number of consecutive pixels across vertical direction that are output

simultaneously from a zone. This field is omitted when all pixels are in the same line.

<ExtY>: a letter in range {∅, E} declaring the location of the pixels extractors across vertical direction. Value E indicates that pixel extractors are at both top and bottom lines. This field is omitted when all pixel extractors are at all the top line.

20.2.1.3 Tap geometrical properties

The following tables provide description of all the tap geometry configurations. For every configuration the first and last pixel belonging to that tap, as well as the pixel increment corresponding to the given tap is listed.

This table enumerates the standard tap geometries. The table is sorted by increasing number of taps. It displays the values of the 6 geometrical properties for each tap.

Table 20-1 Tap geometrical properties – One, two and three taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X-1Y	1X	Tap1	1	W	1	1	H	1
1X2-1Y	1X2	Tap1	1	W-1	2	1	H	1
		Tap2	2	W	2	1	H	1
2X-1Y	2X	Tap1	1	W/2	1	1	H	1
		Tap2	W/2+1	W	1	1	H	1
2XE-1Y	2XE	Tap1	1	W/2	1	1	H	1
		Tap2	W	W/2+1	-1	1	H	1
2XM-1Y	2XM	Tap1	W/2	1	-1	1	H	1
		Tap2	W/2+1	W	1	1	H	1
1X-1Y2		Tap1	1	W	1	1	H-1	2
		Tap2	1	W	1	2	H	2
1X-2YE		Tap1	1	W	1	1	H/2	1
		Tap2	1	W	1	H	H/2+1	-1
1X3-1Y	1X3	Tap1	1	W-2	3	1	H	1
		Tap2	2	W-1	3	1	H	1
		Tap3	3	W	3	1	H	1
3X-1Y	3X	Tap1	1	W/3	1	1	H	1
		Tap2	W/3+1	2W/3	1	1	H	1
		Tap3	2W/3+1	W	1	1	H	1

Table 20-2 Tap geometrical properties – Four taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X4-1Y	1X4	Tap1	1	W-3	4	1	H	1
		Tap2	2	W-2	4	1	H	1
		Tap3	3	W-1	4	1	H	1
		Tap4	4	W	4	1	H	1
4X-1Y	4X	Tap1	1	W/4	1	1	H	1
		Tap2	W/4+1	W/2	1	1	H	1
		Tap3	W/2+1	3W/4	1	1	H	1
		Tap4	3W/4+1	W	1	1	H	1
2X2-1Y	2X2	Tap1	1	W/2-1	2	1	H	1
		Tap2	2	W/2	2	1	H	1
		Tap3	W/2+1	W-1	2	1	H	1
		Tap4	W/2+2	W	2	1	H	1
2X2E-1Y	2X2E	Tap1	1	W/2-1	2	1	H	1
		Tap2	2	W/2	2	1	H	1
		Tap3	W-1	W/2+1	-2	1	H	1
		Tap4	W	W/2+2	-2	1	H	1
2X2M-1Y	2X2M	Tap1	W/2-1	1	-2	1	H	1
		Tap2	W/2	2	-2	1	H	1
		Tap3	W/2+1	W-1	2	1	H	1
		Tap4	W/2+2	W	2	1	H	1
1X2-2YE		Tap1	1	W-1	2	1	H/2	1
		Tap2	2	W	2	1	H/2	1
		Tap3	1	W-1	2	H	H/2+1	-1
		Tap4	2	W	2	H	H/2+1	-1
2X-2YE		Tap1	1	W/2	1	1	H/2	1
		Tap2	W/2+1	W	1	1	H/2	1
		Tap3	1	W/2	1	H	H/2+1	-1
		Tap4	W/2+1	W	1	H	H/2+1	-1
2XE-2YE		Tap1	1	W/2	1	1	H/2	1
		Tap2	W	W/2+1	-1	1	H/2	1
		Tap3	1	W/2	1	H	H/2+1	-1
		Tap4	W	W/2+1	-1	H	H/2+1	-1
2XM-2YE		Tap1	W/2	1	-1	1	H/2	1
		Tap2	W/2+1	W	1	1	H/2	1
		Tap3	W/2	1	-1	H	H/2+1	-1
		Tap4	W/2+1	W	1	H	H/2+1	-1

Table 20-3 Tap geometrical properties –Eight taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X8-1Y	1X8	Tap1	1	W-7	8	1	H	1
		Tap2	2	W-6	8	1	H	1
		Tap3	3	W-5	8	1	H	1
		Tap4	4	W-4	8	1	H	1
		Tap5	5	W-3	8	1	H	1
		Tap6	6	W-2	8	1	H	1
		Tap7	7	W-1	8	1	H	1
		Tap8	8	W	8	1	H	1
8X-1Y	8X	Tap1	1	W/8	1	1	H	1
		Tap2	W/8+1	2W/8	1	1	H	1
		Tap3	2W/8+1	3W/8	1	1	H	1
		Tap4	3W/8+1	4W/8	1	1	H	1
		Tap5	4W/8+1	5W/8	1	1	H	1
		Tap6	5W/8+1	6W/8	1	1	H	1
		Tap7	6W/8+1	7W/8	1	1	H	1
		Tap8	7W/8+1	W	1	1	H	1
4X2-1Y	4X2	Tap1	1	W/4-1	2	1	H	1
		Tap2	2	W/4	2	1	H	1
		Tap3	W/4+1	W/2-1	2	1	H	1
		Tap4	W/4+1	W/2	2	1	H	1
		Tap5	W/2+1	3W/4-1	2	1	H	1
		Tap6	W/2+2	3W/4	2	1	H	1
		Tap7	3W/4+1	W-1	2	1	H	1
		Tap8	3W/4+2	W	2	1	H	1
2X2E-2YE		Tap1	1	W/2-1	2	1	H/2	1
		Tap2	2	W/2	2	1	H/2	1
		Tap3	W-1	W/2+1	-2	1	H/2	1
		Tap4	W	W/2+2	-2	1	H/2	1
		Tap5	1	W/2-1	2	H	H/2+1	-1
		Tap6	2	W/2	2	H	H/2+1	-1
		Tap7	W-1	W/2+1	-2	H	H/2+1	-1
		Tap8	W	W/2+2	-2	H	H/2+1	-1

20.3 Tap geometry drawings

20.3.1 Single Tap geometry

1X-1Y (area-scan)

1 zone in X, 1 zone in Y.

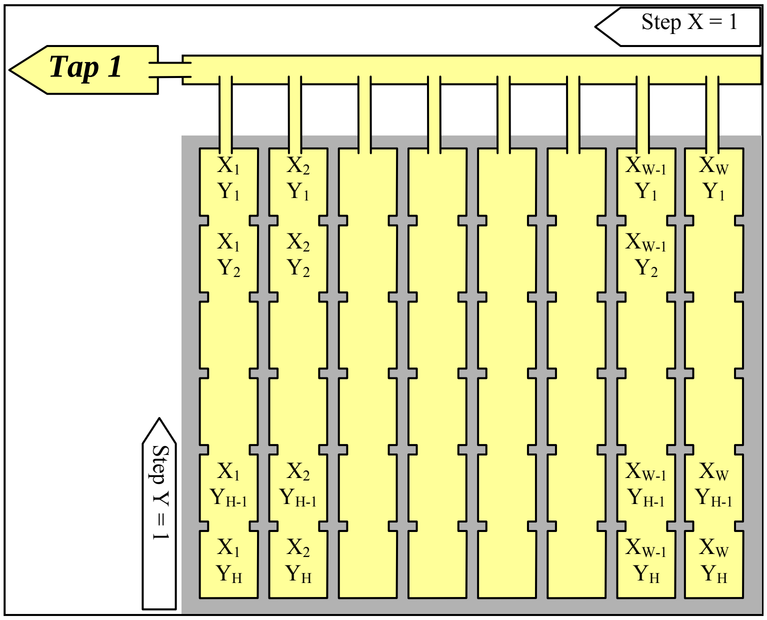

Figure 20-1 Geometry 1X-1Y (area-scan)

1X (line-scan)

1 zone in X.

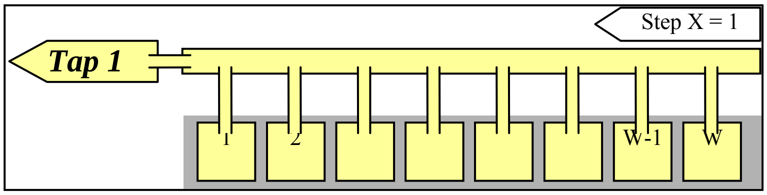

Figure 20-2 Geometry 1X (line-scan)

20.3.2 Dual Tap geometries

1X2-1Y (area-scan)

1 zone in X with 2 taps, 1 zone in Y.

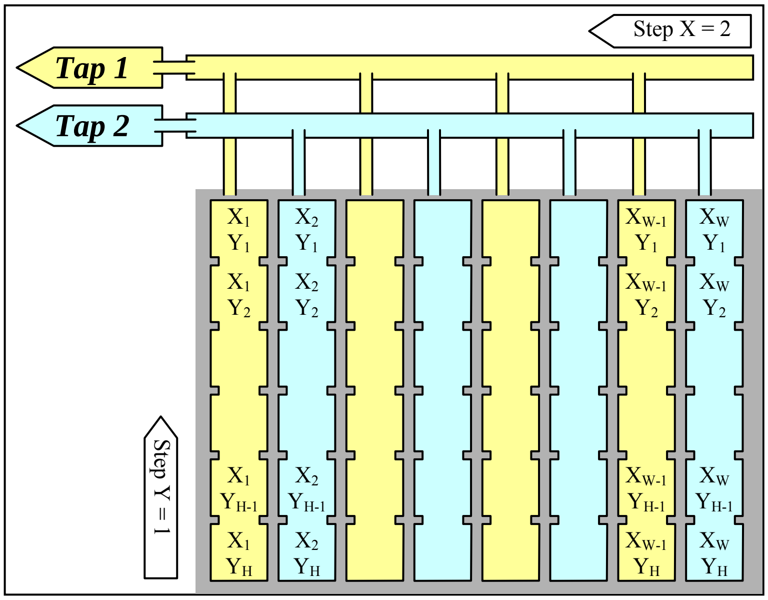

Figure 20-3 Geometry 1X2-1Y (area-scan)

1X2 (line-scan)

1 zone in X with 2 taps.

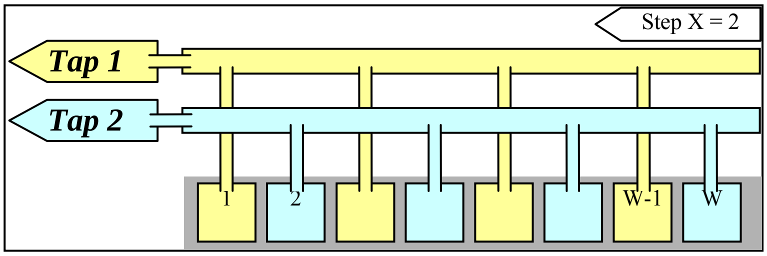

Figure 20-4 Geometry 1X2 (line-scan)

2X-1Y (area-scan)

2 zones in X, 1 zone in Y.

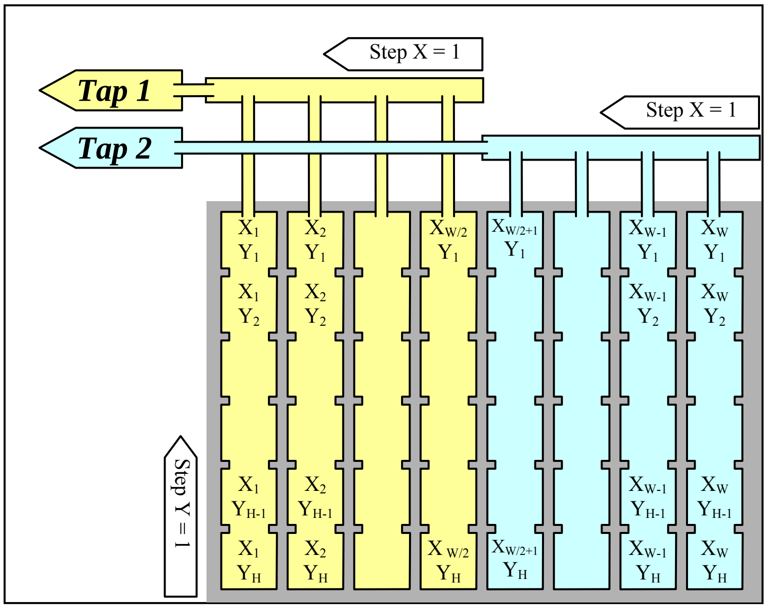

Figure 20-5 Geometry 2X-1Y (area-scan)

2X (line-scan)

2 zones in X.

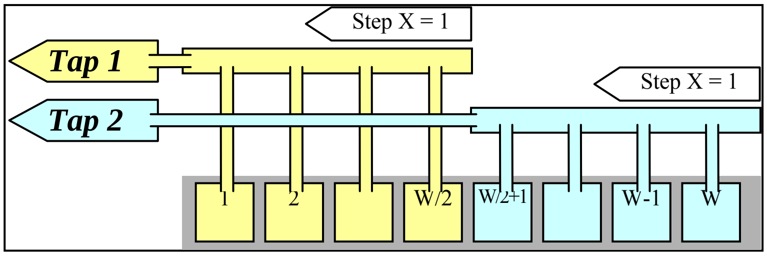

Figure 20-6 Geometry 2X (line-scan)

2XE-1Y (area-scan)

2 zones in X with end extraction, 1 zone in Y.

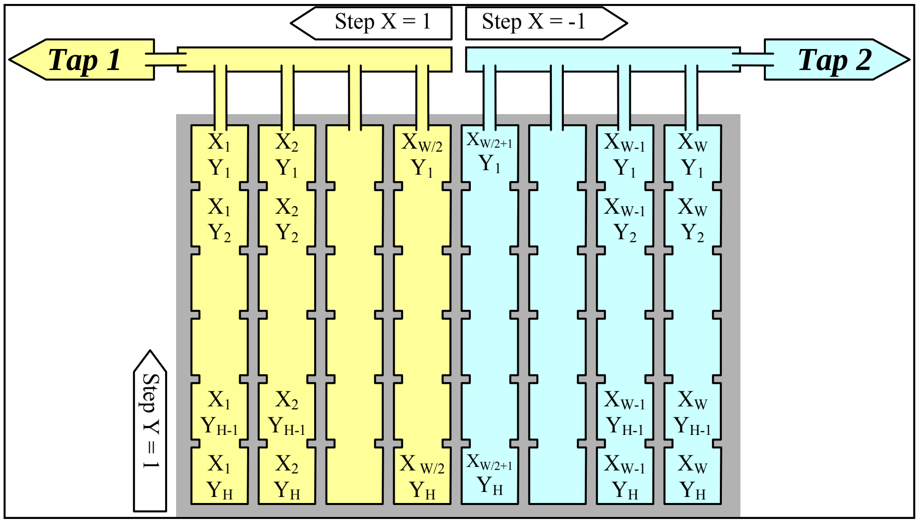

Figure 20-7 Geometry 2XE-1Y (area-scan)

2XE (line-scan)

2 zones in X with end extraction.

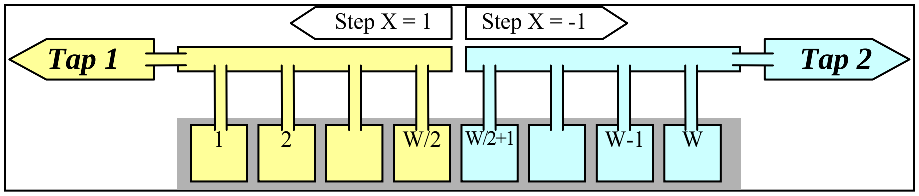

Figure 20-8 Geometry 2XE (line-scan)

2XM-1Y (area-scan)

2 zones in X with middle extraction, 1 zone in Y.

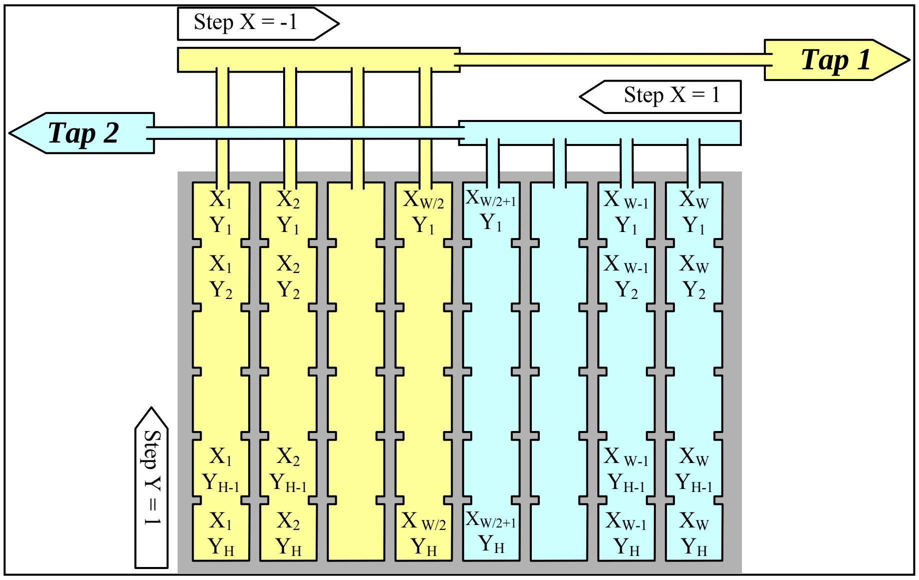

Figure 20-9 Geometry 2XM-1Y (area-scan)

2XM (line-scan)

2 zones in X with middle extraction.

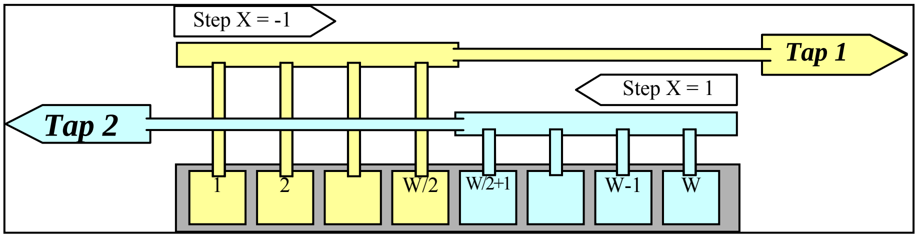

Figure 20-10 Geometry 2XM (line-scan)

1X-1Y2 (area-scan)

1 zone in X, 1 zone in Y with 2 taps.

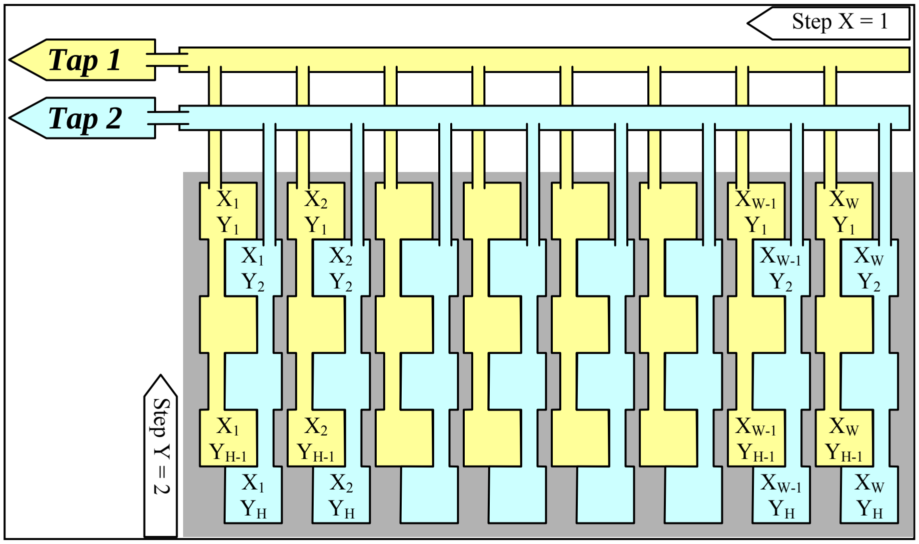

Figure 20-11 Geometry 1X-1Y2 (area-scan)

1X-2YE (area-scan)

1 zone in X, 2 zones in Y with end extraction.

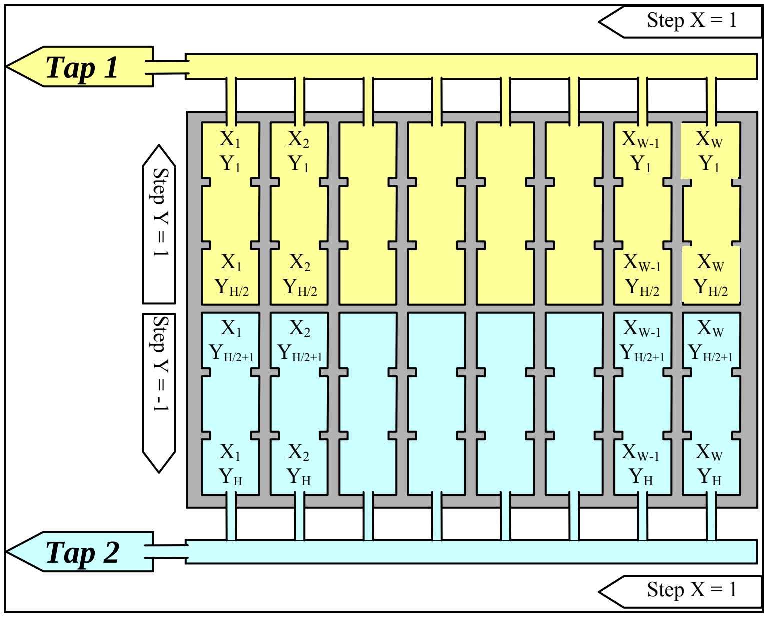

Figure 20-12 Geometry 1X-2YE (area-scan)

20.4 Triple Tap geometries

1X3-1Y (area-scan)

1 zone in X with 3 taps, 1 zone in Y.

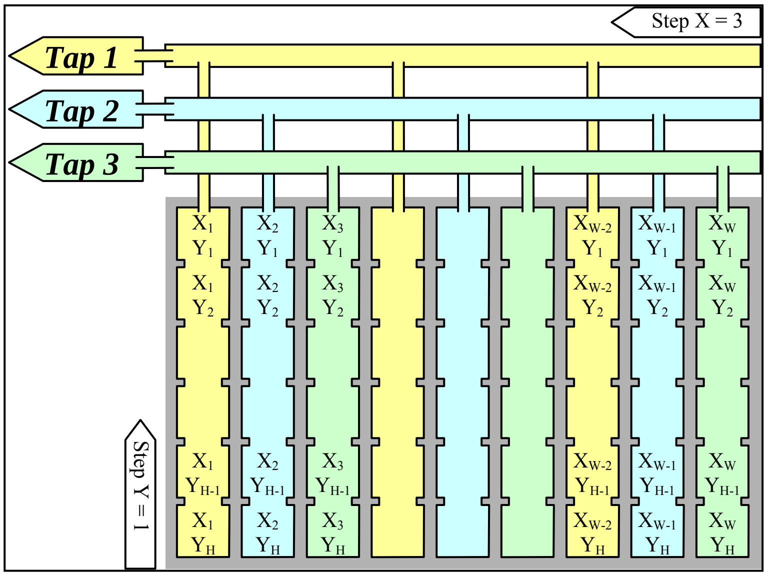

Figure 20-13 Geometry 1X3-1Y (area-scan)

1X3 (line-scan)

1 zone in X with 3 taps.

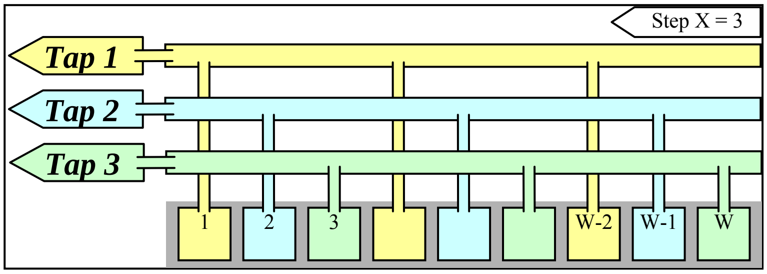

Figure 20-14 Geometry 1X3 (line-scan)

3X-1Y (area-scan)

3 zones in X, 1 zone in Y.

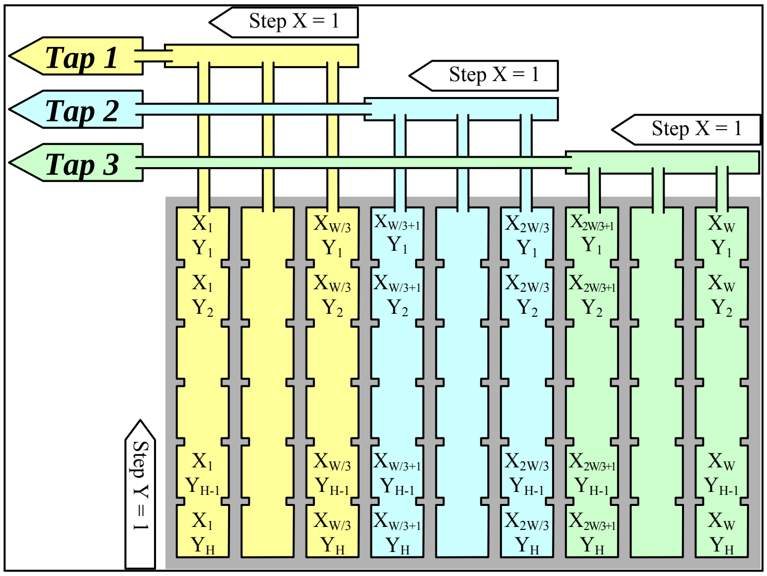

Figure 20-15 Geometry 1X3-1Y (area-scan)

3X (line-scan)

3 zones in X.

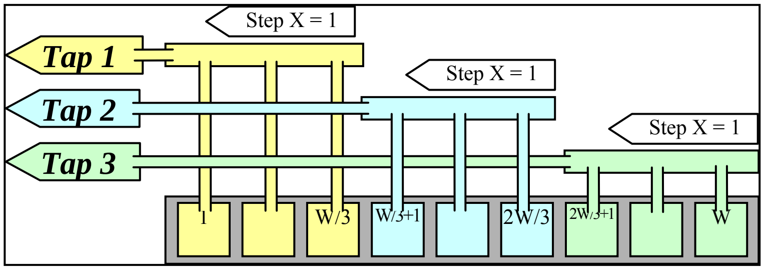

Figure 20-16 Geometry 1X3 (line-scan)

20.5 Quad Tap geometries

1X4-1Y (area-scan)

1 zone in X with 4 taps, 1 zone in Y.

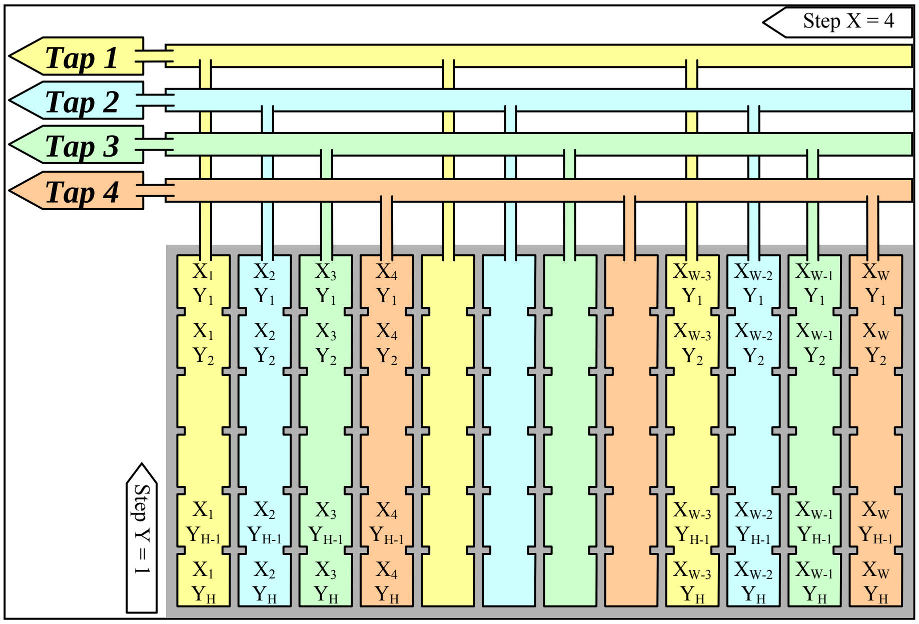

Figure 20-17 Geometry 1X4-1Y (area-scan)

1X4 (line-scan)

1 zone in X with 4 taps.

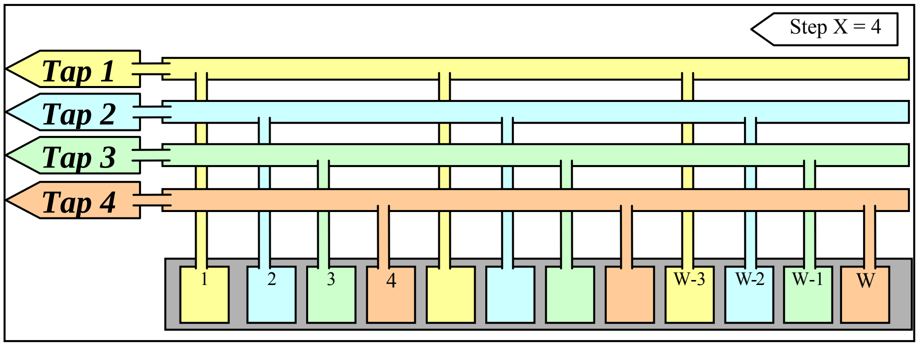

Figure 20-18 Geometry 1X4 (line-scan)

4X-1Y (area-scan)

4 zones in X, 1 zone in Y.

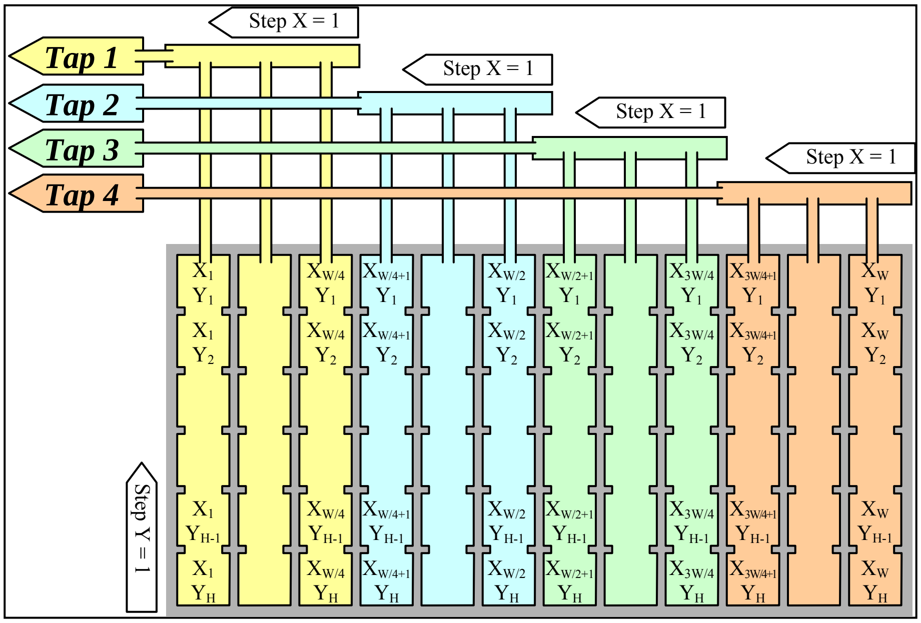

Figure 20-19 Geometry 4X-1Y (area-scan)

4X (line-scan)

4 zones in X.

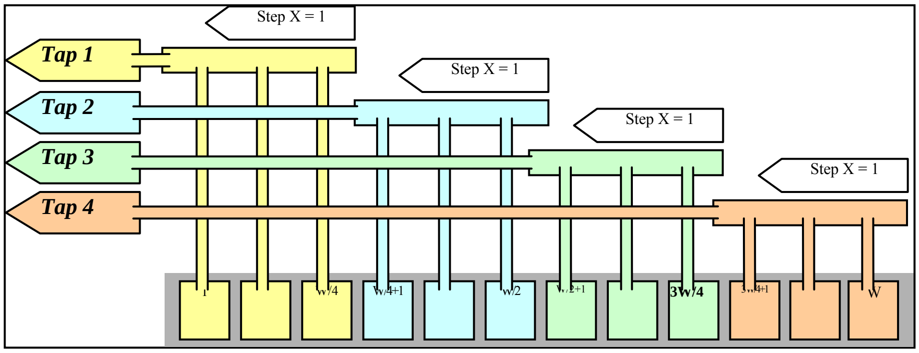

Figure 20-20 Geometry 4X (line-scan)

2X2-1Y (area-scan)

2 zones in X with 2 taps, 1 zone in Y.

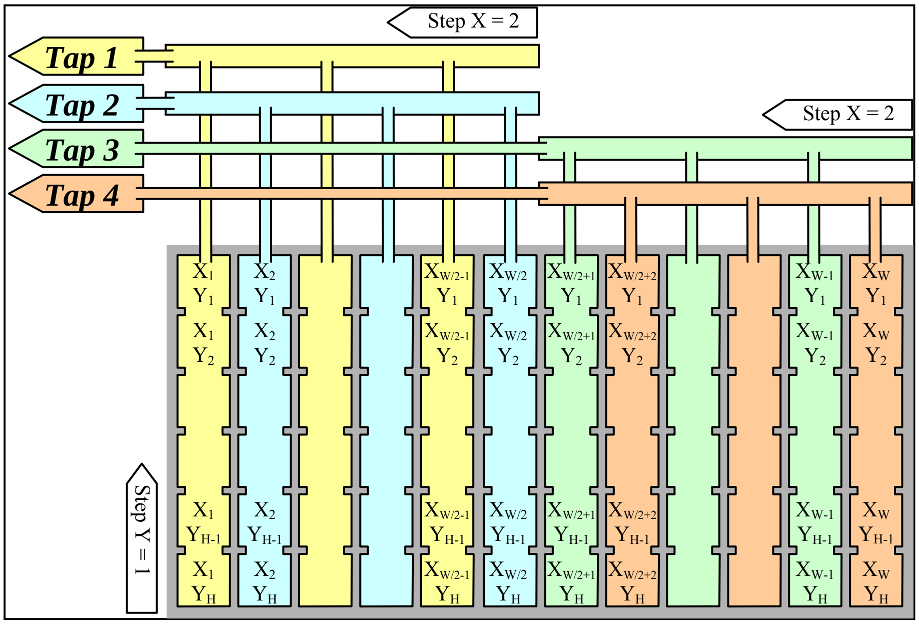

Figure 20-21 Geometry 2X2-1Y (area-scan)

2X2 (line-scan)

2 zones in X with 2 taps.

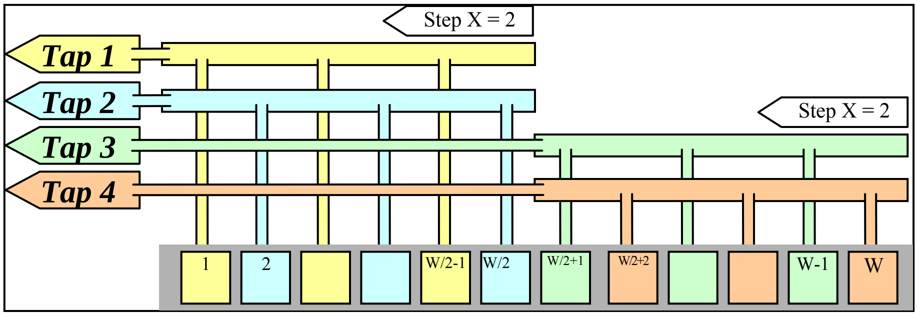

Figure 20-22 Geometry 2X2 (line-scan)

2X2E-1Y (area-scan)

2 zones in X with 2 taps and end extraction, 1 zone in Y.

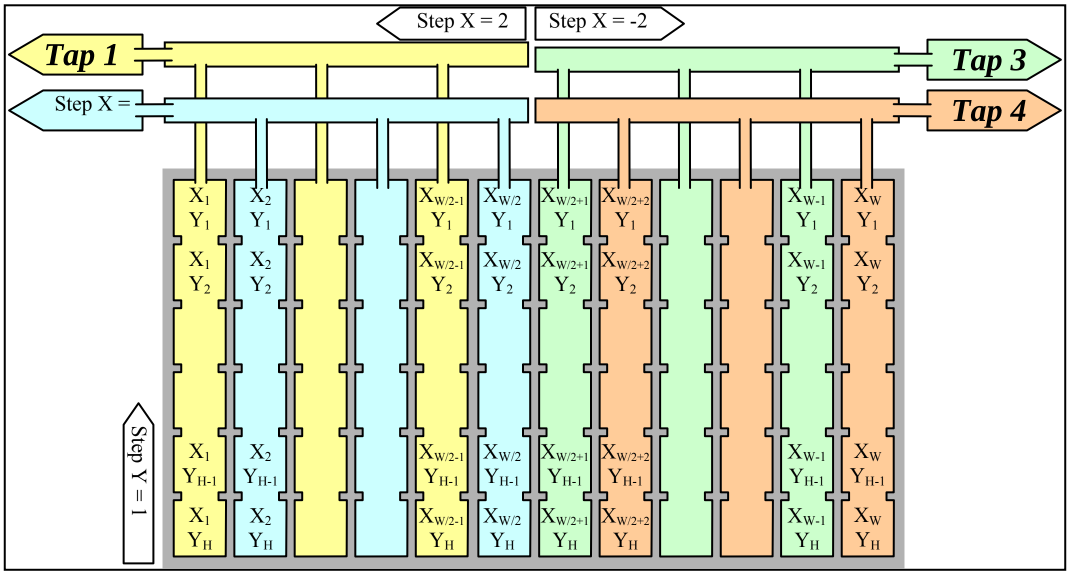

Figure 20-23 Geometry 2X2E-1Y (area-scan)

2X2E (line-scan)

2 zones in X with 2 taps and end extraction.

See similar geometries for building a drawing.

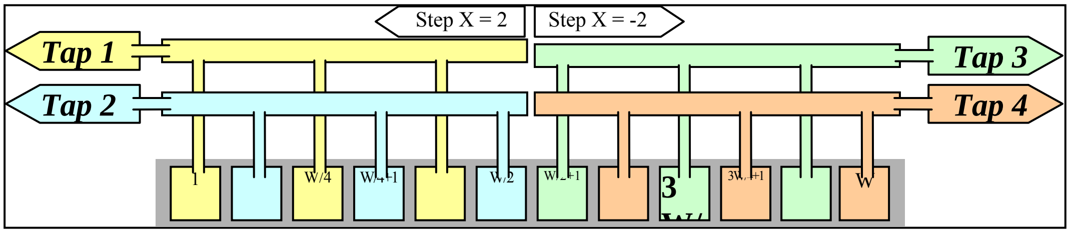

Figure 20-24 Geometry 2X2E (line-scan)

2X2M-1Y (area-scan)

2 zones in X with 2 taps and middle extraction, 1 zone in Y.

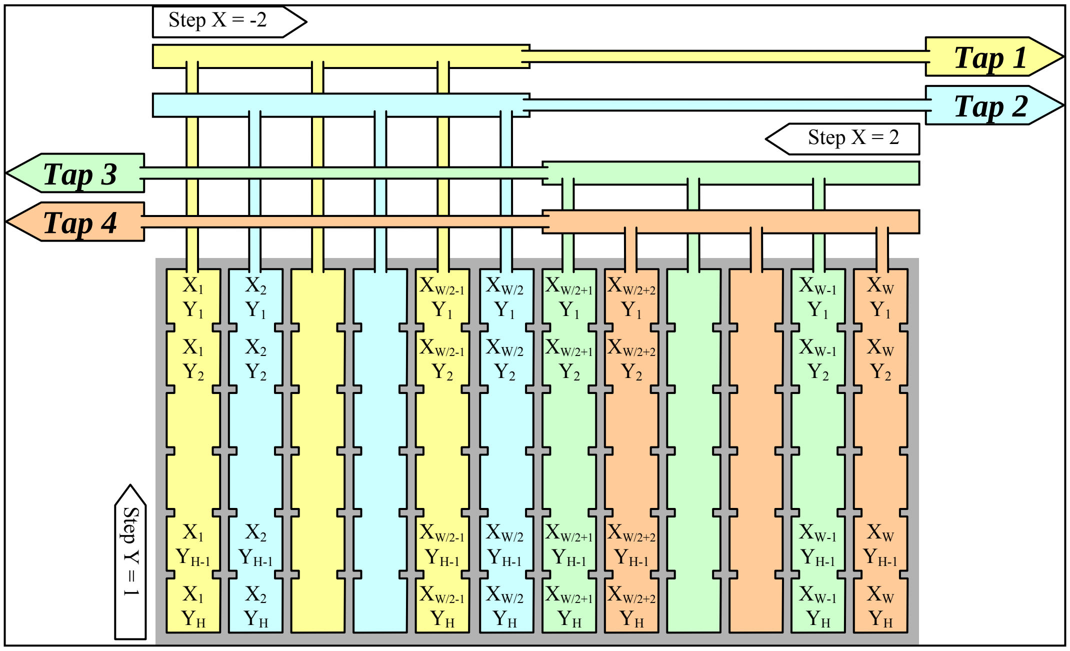

Figure 20-25 Geometry 2X2M-1Y (area-scan)

2X2M (line-scan)

2 zones in X with 2 taps and middle extraction.

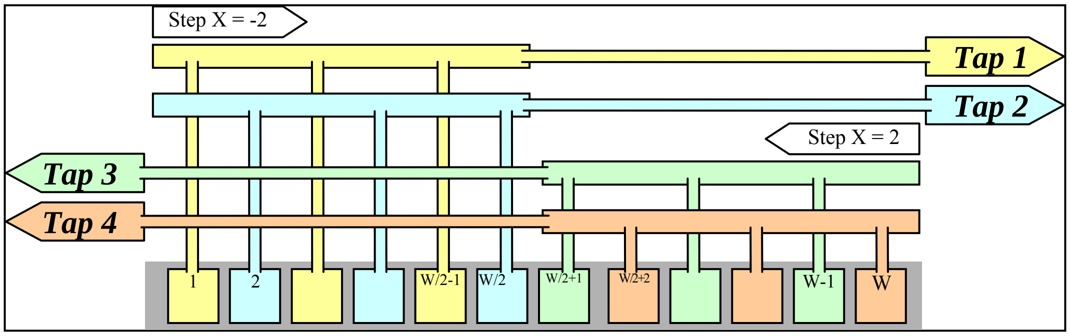

Figure 20-26 Geometry 2X2M (line-scan)

1X2-2YE (area-scan)

1 zone in X with 2 taps, 2 zones in Y with end extraction.

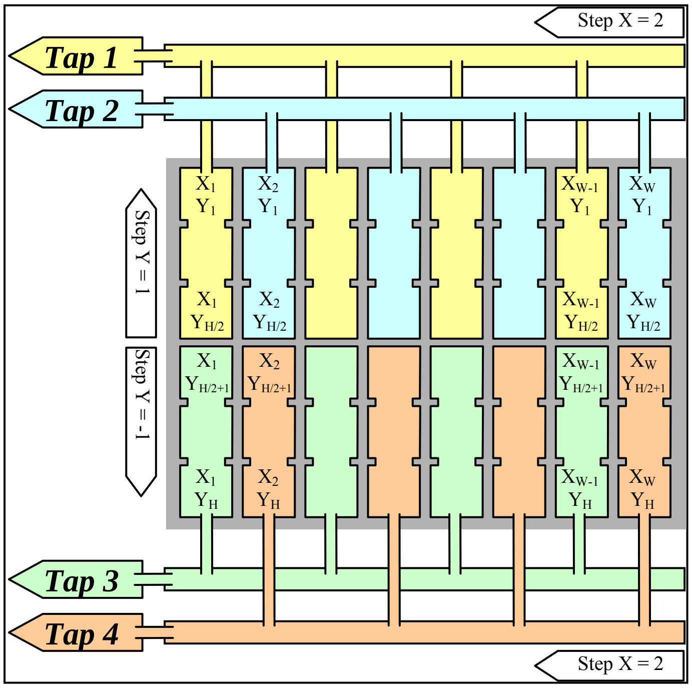

Figure 20-27 Geometry 1X2-2YE (area-scan)

2X-2YE (area-scan)

2 zones in X, 2 zones in Y with end extraction.

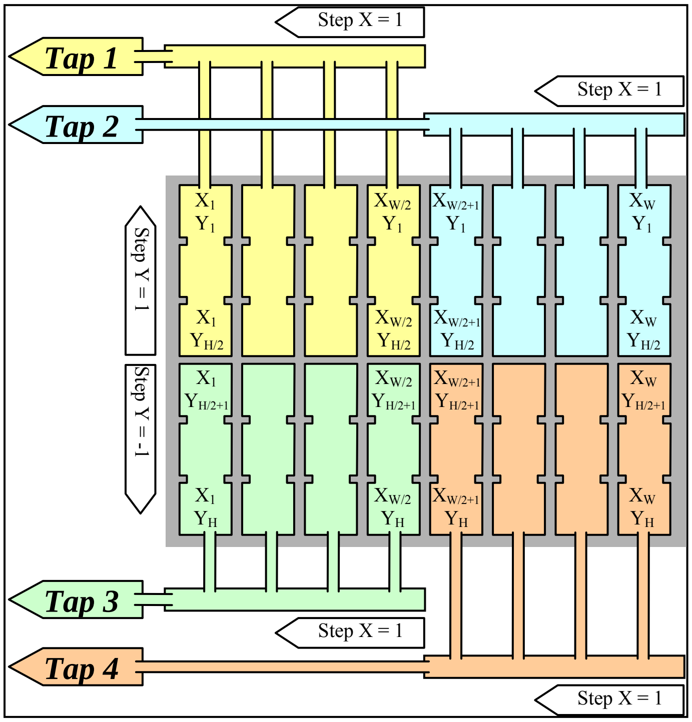

Figure 20-28 Geometry 1X2-2YE (area-scan)

2XE-2YE (area-scan)

2 zones in X with end extraction, 2 zones in Y with end extraction.

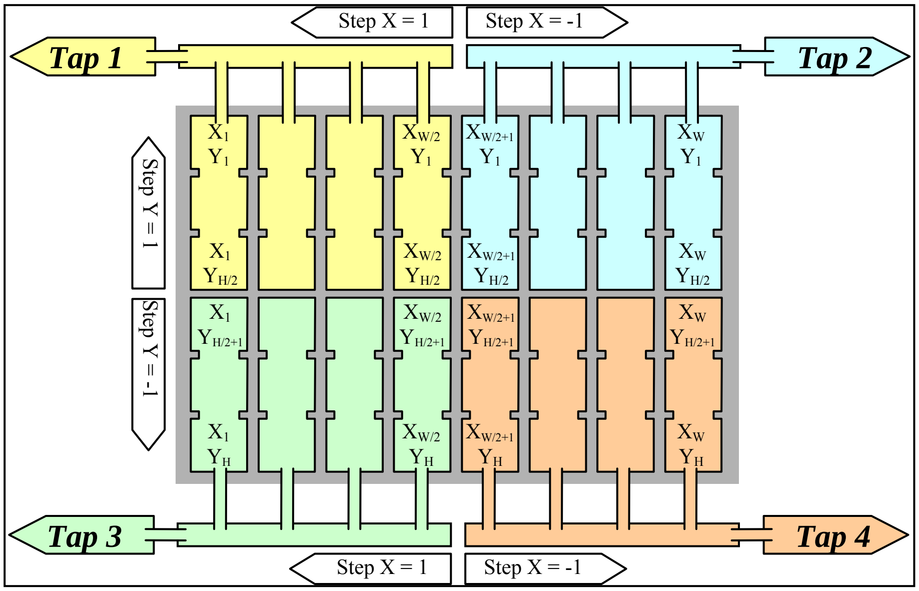

Figure 20-29 Geometry 2XE-2YE (area-scan)

2XM-2YE (area-scan)

2 zones in X with middle extraction, 2 zones in Y with end extraction.

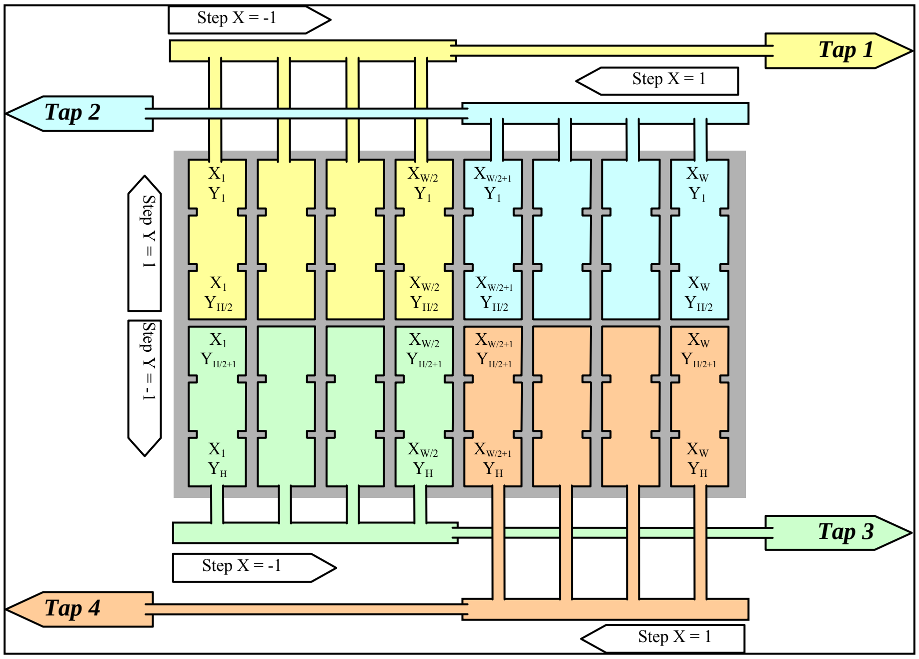

Figure 20-30 Geometry 2XM-2YE (area-scan)

20.6 Octal Tap geometries

1X8-1Y (area-scan)

1 zone in X with 8 taps, 1 zone in Y.

See similar geometries for building a drawing.

1X8 (line-scan)

1 zone in X with 8 taps.

See similar geometries for building a drawing.

8X-1Y (area-scan)

8 zones in X, 1 zone in Y.

See similar geometries for building a drawing.

8X (line-scan)

8 zones in X.

See similar geometries for building a drawing.

4X2-1Y (area-scan)

4 zones in X with 2 taps, 1 zone in Y.

See similar geometries for building a drawing.

4X2 (line-scan)

4 zones in X with 2 taps.

See similar geometries for building a drawing.

2X2E-2YE (area-scan)

2 zones in X with 2 taps, 2 zones in Y with 2 taps.

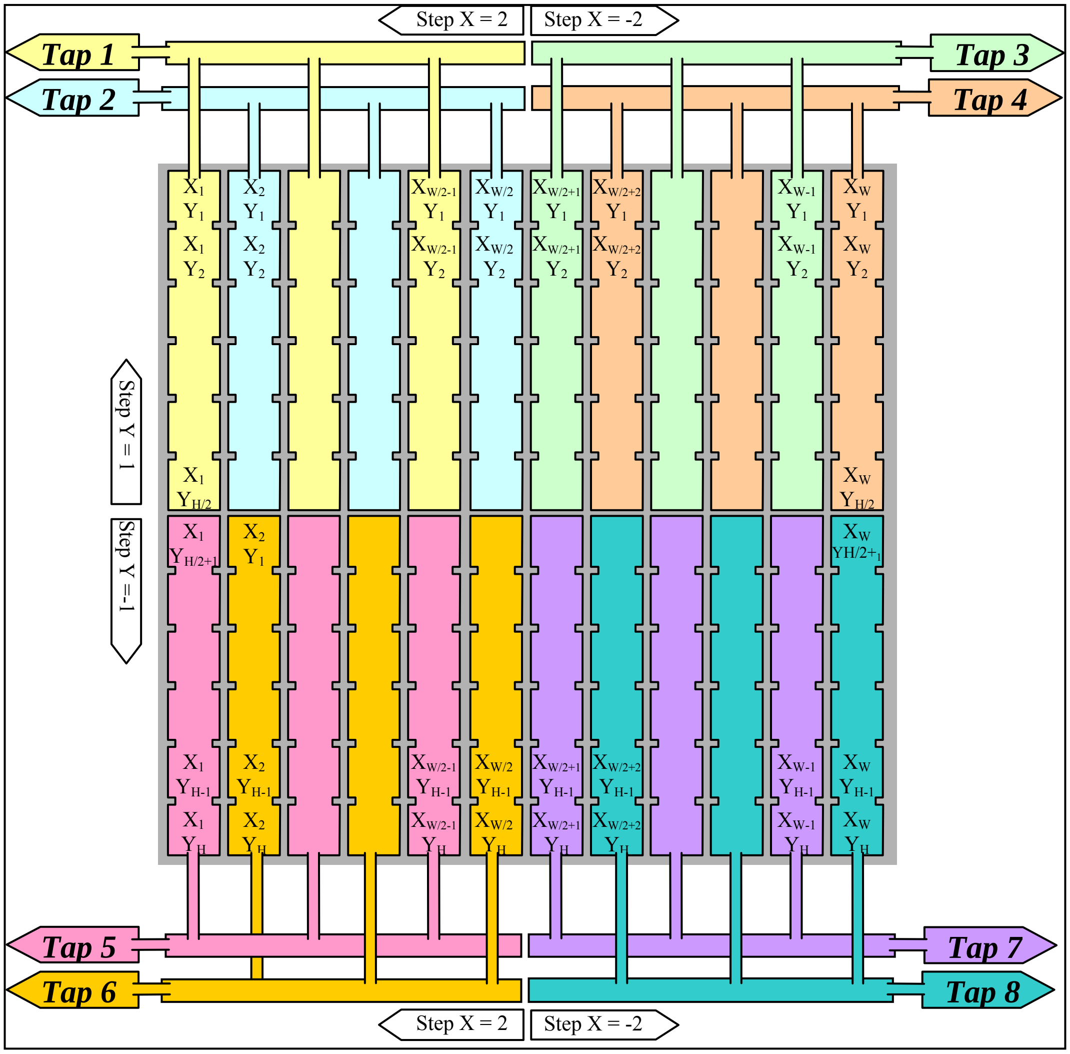

Figure 20-31 Geometry 2X2E-2YE (area-scan)