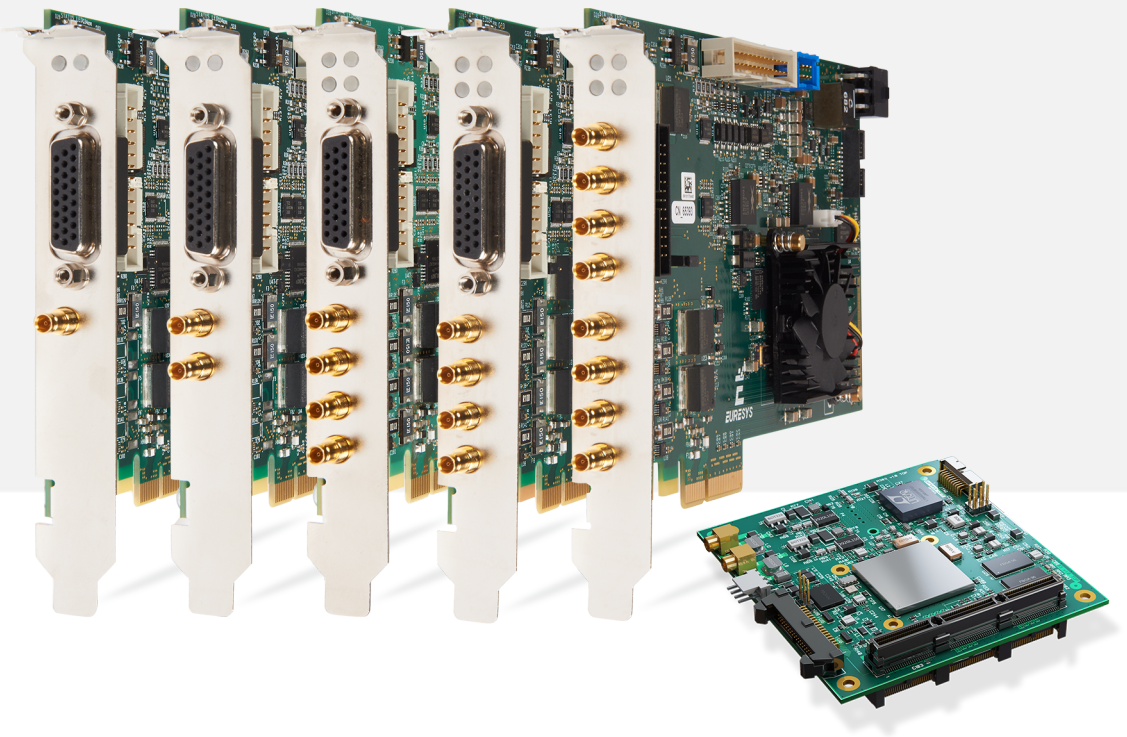
EURESYS™
Excellence in vision

FUNCTIONAL GUIDE

Coaxlink

Coaxlink Driver Version 6.1



CoaXPress

© EURESYS s.a. 2016 - Document version 6.1.513 built on 2016-09-08

EURESYS s.a. shall retain all property rights, title and interest of the documentation of the hardware and the software, and of the trademarks of EURESYS s.a. All the names of companies and products mentioned in the documentation may be the trademarks of their respective owners. The licensing, use, leasing, loaning, translation, reproduction, copying or modification of the hardware or the software, brands or documentation of EURESYS s.a. contained in this book, is not allowed without prior notice. EURESYS s.a. may modify the product specification or change the information given in this documentation at any time, at its discretion, and without prior notice. EURESYS s.a. shall not be liable for any loss of or damage to revenues, profits, goodwill, data, information systems or other special, incidental, indirect, consequential or punitive damages of any kind arising in connection with the use of the hardware or the software of EURESYS s.a. or resulting of omissions or errors in this documentation.

Contents

About This Document **8**
- Document Scope 8
- Document Revision History 8
- Document Changes 9

About Coaxlink Driver Version 6.1 ..... **11**
- Coaxlink Products 11
- Firmware Variants per Product 12
- Development Tools 13

Managing the Firmware **14**
- Firmware Manager Tool 14
- Firmware Upgrade 14
- Firmware Variant Installation 15

Product and Firmware Variant Selector **16**

Coaxlink Architecture **17**
- Coaxlink Block Diagram 17
  - Interface Module Elements 18
  - Device Module Elements 18
  - Data Stream Module Elements 18
  - Buffer Module Elements 19

CoaXPress Host Interface **20**
- CoaXPress Standard Compliance 20
- Host Interface Specification Summary 21
  - Specification Summary – Card Level 21
  - Specification Summary – Camera Level 22
  - Specification Summary – Image Data Stream Level 22
- Camera Connection Schemes 23
  - Host Connection Map Naming Convention 23
  - Host Connection Maps – Standard CoaXPress Devices 23
  - Host Connection Maps – Data Forwarding Devices 24
  - Host Connection Maps – Sub-Link Devices 25

Firmware Variant & Product vs. Host Connection Map ... 25
Link Configuration ... 26
CoaXPress 1.1.1 Discovery ... 26
CoaXPress Link GenICam Features ... 27
Power Over CoaXPress ... 30
Power Over CoaXPress GenICam Features ... 31
CoaXPress I/O Channel ... 33
CoaXPress Host To Device Trigger ... 34
Host to Device Trigger Transmitter ... 34
Camera Trigger to Exposure Latency ... 36
Host to Device Trigger Source ... 37
Host to Device Trigger GenICam Features ... 37
CoaXPress Host Connector Indicator Lamps ... 38
Connection Test ... 39
Connection Test GenICam Features ... 40

CoaXPress Data Forwarding Quad G3 DF ... **41**
Data Forwarding Principles ... 41
Data Forwarding Connection Schemes ... 42
Line-scan Triggers Synchronization ... 43
Configuration Script Example ... 43

Image Data Acquisition ... **45**
Acquisition Gating ... 45
FIFO Buffer ... 45
Image Data Stream Processing ... 46
Endianness Conversion ... 46
Pixel Component Unpacking ... 47
Pixel Component Order ... 48
Pixel Order ... 48
Processor Configurations ... 49
Pixel Processing GenICam Features ... 50
Stream Statistics GenICam Features ... 51

Line Scan Acquisition ... **53**
Line Scan Acquisition Principles ... 53
Line Scan Acquisition Control GenICam Features ... 55
Line Scan Acquisition Use cases ... 58

Camera Control Principles ... **61**
Camera Cycle ... 61
Camera Cycle Concatenation Rules ... 62

Camera Control Methods ... 63

Illumination Control Principles ... 64
Illumination Devices ... 64
Aligning Camera and illumination Cycles ... 64

Camera and Illumination Controller ... 66
Description ... 66
Cycle Timing Machine ... 66
Cycle Manager ... 67
Cycle Trigger Manager ... 68
Sequence Manager ... 68
Timing Diagrams ... 69
Single CIC Cycle ... 69
Consecutive Overlapping CIC Cycles ... 70
Cycle Sequence ... 73
Camera Trigger Signal Paths ... 73
Strobe Signal Paths ... 74
CIC Synchronization ... 74
CIC Synchronization Principle ... 74
CIC Synchronization Timing Diagram ... 76
Camera and Illumination Control GenICam Features ... 76
Camera Model GenICam Features ... 77
Cycle Timing GenICam Features ... 79
Cycle Control GenICam Features ... 79
Sequence Control GenICam Features ... 81

C2C-Link ... 83
C2C-Link Specification ... 84
Trigger Propagation Delays ... 85
C2C-Link Use Cases ... 86
C2C-Link Availability ... 87
C2C-Link Setup ... 87
C2C-Link Hardware Components ... 88
3303 C2C-Link Ribbon Cable ... 88
Custom C2C-Link Ribbon Cable Assembly ... 89
1636 InterPC C2C-Link Adapter ... 89
C2C-Link Extender ... 91
HD26F I/O Adapter ... 94

I/O Toolbox ... 95
I/O Toolbox Context ... 95

I/O Toolbox Structure....96
I/O Toolbox Composition....97
Line Input Tool....98
Line Input Tool GenICam Features....98
Quadrature Decoder Tool....100
Quadrature Decoder Tool GenICam Features....101
User Actions Tool....104
User Actions Scheduler....105
User Events....106
User Output Register....106
User Actions GenICam Features....106
Event Input Tool....109
Event Input Tool GenICam Features....110
Divider Tool....110
Divider Tool GenICam Features....111
Multiplier/Divider Tool....113
Multiplier/Divider Tool GenICam Features....115
Delay Tool....118
Delay Tool GenICam Features....119

General Purpose I/O Lines....122
I/O Lines Types....122
Available I/O Lines....122
I/O Lines Usage....123
I/O Control Blocks....123
Line Mode Control....124
I/O Lines Specifics....125
Line Polarity Control....125
Filter Control....126
Line Source Selection....126
Logical I/O Line State....127
Physical I/O Line State....127
Line Driver Physical Output States....128
Initial States....128
Digital I/O Control GenICam Features....129

Event Signaling And Counting....134
Short Description....134
About GenTL Signaling....134
Event Types....135
Custom Events Sources....136
Custom Events Counter....138
Custom Events Configuration....138
Event Specific Context Data....139
Event Control GenICam Features (Interface)....140

Event Control GenICam Features (Device)....143
Event Control GenICam Features (Data Stream)....146

OEM Safety Key....149
OEM Safety Key GenICam Features....149
Using OEMSafetyKey....150

About This Document

Document Scope

This document describes and explains how to use the functions of the Coaxlink products when they are operated with Coaxlink Driver version 6.1.

Unless specified, the functions described in this document are applicable to all the Coaxlink products and their firmware variants supported by the Coaxlink Driver.

Note: *For a description of the hardware, refer to the Coaxlink Hardware Manual.*

Document Revision History

Date	Version	Description
2014-07-24	1.0	Functional Guide of Coaxlink Driver 1.0
2014-08-07	2.0	Functional Guide of Coaxlink Driver 2.0 • Add 1630 Coaxlink Mono product • Add 1631 Coaxlink Duo product
2014-09-26	3.0	Functional Guide of Coaxlink Driver 3.0 • Add initial support of Linux
2014-11-20	3.2	Functional Guide of Coaxlink Driver 3.2 • Add 2-camera firmware variant on 1631 Coaxlink Duo product • Add external triggers and C2C-Link
2015-01-16	4.0	Functional Guide of Coaxlink Driver 4.0 • Add 1633 Coaxlink Quad G3 product • Add 1-camera and 4-camera firmware variant on 1633 Coaxlink Quad G3
2015-04-02	4.1	Functional Guide of Coaxlink Driver 4.1 • Add 1-line-scan firmware variant on 1632 Coaxlink Quad product • Add 1-line-scan firmware variant on 1633 Coaxlink Quad G3 product
2015-06-10	4.2	Functional Guide of Coaxlink Driver 4.2 • Add 2-camera firmware variant on 1632 Coaxlink Quad product • Add 2-camera firmware variant on 1633 Coaxlink Quad G3 product • Add unpacking options and red/blue swap option to the pixel processor.
2015-07-03	4.3	Functional Guide of Coaxlink Driver 4.3 • Add Events Signaling and Counting

Date	Version	Description
2015-07-17	4.3.1	Functional Guide of Coaxlink Driver 4.3.1 • Add 2-line-scan firmware variant on 1633 Coaxlink Quad G3
2016-01-07	4.5	Functional Guide of Coaxlink Driver 4.5 • Improved CoaXPress Host to Device Trigger transmission unit. • Extension of the I/O Toolbox, more LIN tools, new QDC, DIV and DEL tools • Improved CIC Timing Machine: extension of exposure/strobe duration
2016-03-04	4.6	Functional Guide of Coaxlink Driver 4.6 • New layout
2016-06-10	5.0	Functional Guide of Coaxlink Driver 5.0 • Add 1635 Coaxlink Quad G3 DF product
2016-07-11	5.1	Functional Guide of Coaxlink Driver 5.1 • Add the capability to forward a trigger signal applied on any GPIO input port to the camera through the CoaXPress Link using CoaXPress Host to Device trigger messages. See *CoaXPress Host To Device Trigger* on page 34. • Add OEM safety key functionality. See *OEM Safety Key* on page 149. • Add the capability to trigger the start of an acquisition sequence. See *Sequence Control GenICam Features* on page 81.
2016-08-05	6.0	Functional Guide of Coaxlink Driver 6.0 • Add 1634 Coaxlink Duo PCIe/104 product • Add *User Actions Tool* on page 104
2016-09-08	6.1	Functional Guide of Coaxlink Driver 6.1 • Extend IntraPC C2C-Link limit to 0.6 m • Add 1636 InterPC C2C-Link Adapter • Extend GPIO Line sources to 'Device<0:3> Stream0 Start Of Camera Readout' • Extend DIV tool range to 16-bit • Improve security level of OEM Safety Key • Preliminary support of sub-link acquisition for a particular 8-connection camera

Document Changes

Version 6.1

- *Coaxlink Products* on page 11: includes also accessories
- *Firmware Variants per Product* on page 12: add 1-slm-camera and 1-sls-camera firmware variant to 1633 Coaxlink Quad G3
- *Product and Firmware Variant Selector* on page 16: add 'Specific 8-connection area-scan camera'
- *Host Connection Maps – Sub-Link Devices* on page 25: new topic.
- *Sequence Control GenICam Features* on page 81: Add `AbortSequence` feature.
- *C2C-Link* on page 83: New section dedicated to C2C-Link.
- *Custom C2C-Link Ribbon Cable Assembly* on page 89: Change maximum cable length specification to 0.60 m.
- *1636 InterPC C2C-Link Adapter* on page 89: New accessory product.
- *Divider Tool* on page 110: Extend upper limit of DividerToolDivisionFactor and DividerToolInitialOffset to 65,535.
- *Digital I/O Control GenICam Features* on page 129: Extend `LineSource` values.

- *OEM Safety Key* on page 149: Revised section.
- *OEM Safety Key GenICam Features* on page 149: Added `OemSafetyKeyVerification` feature.
- *Using OEMSafetyKey* on page 150: New topic.

About Coaxlink Driver Version 6.1

Coaxlink Products

Coaxlink frame grabbers and related accessories

Coaxlink Frame Grabbers

Product	S/N Prefix	Icon
1630 Coaxlink Mono	KMO	Mono
1631 Coaxlink Duo	KDU	Duo
1632 Coaxlink Quad	KQU	Quad
1633 Coaxlink Quad G3	KQG	Quad G3
1634 Coaxlink Duo PCIe/104	KDR	Duo PCIe/104
1635 Coaxlink Quad G3 DF	KDF	Quad G3 DF

Related Accessories

Product	S/N Prefix	Icon
1625 DB25F I/O Adapter Cable		
1636 InterPC C2C-Link Adapter	KCC	1636
3300 HD26F I/O module for Coaxlink Duo PCIe/104		3300
3301 Thermal drain (Model 1) for Coaxlink Duo PCIe/104		
3302 DIN1.0/2.3 Coaxial cable for Coaxlink Duo PCIe/104		3302
3303 C2C-Link Ribbon Cable		
3304 HD26F I/O Adapter Cable		

Note: *The S/N prefix is a 3-letter string at the beginning of the card serial number.*

Note: *Icons are used in this document for tagging titles of card-specific content.*

Firmware Variants per Product

List of available firmware variants per product in Coaxlink Driver 6.1.

Target Product	Firmware Variant	Description	Icon
1630 Coaxlink Mono	1-camera	One 1-connection area-scan camera	Mono
1631 Coaxlink Duo	1-camera	One 1- or 2-connection area-scan camera	Duo
	2-camera	One or two 1-connection area-scan cameras	Duo
1632 Coaxlink Quad	1-camera	One 1- or 2- or 4-connection area-scan camera	Quad
	2-camera	One or two 1- or 2-connection area-scan cameras	Quad
	1-camera, line-scan	One 1- or 2- or 4-connection line-scan camera	Quad
1633 Coaxlink Quad G3	1-camera	One 1- or 2- or 4-connection area-scan camera	Quad G3
	2-camera	One or two 1- or 2-connection area-scan cameras	Quad G3
	4-camera	One or two or three or four 1-connection area-scan cameras	Quad G3
	1-camera, line-scan	One 1- or 2- or 4-connection line-scan camera	Quad G3
	2-camera, line-scan	One or two 1- or 2-connection line-scan cameras	Quad G3
	New 1-slm-camera	Master 4-connection sub-link of a 8-connection area-scan camera	Quad G3
	New 1-sls-camera	Slave 4-connection sub-link of a 8-connection area-scan camera	Quad G3
1634 Coaxlink Duo PCIe/104	1-camera	One 1- or 2-connection area-scan camera	Duo PCIe/104
	2-camera	One or two 1-connection area-scan cameras	Duo PCIe/104
	1-camera, line-scan	One 1- or 2-connection line-scan camera	Duo PCIe/104
1635 Coaxlink Quad G3 DF	1-camera	One 1- or 2- or 4-connection area-scan camera	Quad G3 DF
	1-df-camera	One 1- or 2- or 4-connection area-scan data-forwarded-camera	Quad G3 DF
	1-camera, line-scan	One 1- or 2- or 4-connection line-scan camera	Quad G3 DF

Target Product	Firmware Variant	Description	Icon
	1-df-camera, line-scan	One 1- or 2- or 4-connection line-scan data-forwarded-camera	

Notice: *Combined firmware/product icons are used in this document for tagging content that applies only to a particular firmware variant of the designated product.*

Development Tools

The Coaxlink driver is supplied as GenICam GenTL producer libraries (coaxlink.cti):

- An x86 dynamic library designed to be used with ISO-compliant C/C++ compilers for the development of x86 applications.
- An x86_64 dynamic library designed to be used with ISO-compliant C/C++ compilers for the development of x86_64 applications.
- A .NET assembly designed to be used with development environments compatible with .NET frameworks version 2.0 or higher.

The Coaxlink cards should be usable with any development tool that supports at least one of these interfaces.

Managing the Firmware

How to manage the firmware upgrades and the installation of firmware variants.

The heart of a Coaxlink card is a RAM-based FPGA (Field Programmable Gate Array) that implements several functions including the PCI Express end point.

Considering that a PCI Express end point must be ready within 150 milliseconds of the power-up time, the FPGA content, hereafter named "firmware", must be loaded quickly after having applied power to the card. Therefore, the firmware is stored into a non-volatile flash EEPROM allowing a fast start-up of the FPGA.

Note: *This situation differs from other Euresys products, such as Grablink cards, where the FPGA content is loaded by the MultiCam driver when it starts or at any time if a FPGA configuration change is requested.*

The Coaxlink Driver will never modify the content the FPGA during operation. Any modification of the FPGA content requires a two-step operation:

- **Step 1**: The new firmware is written into the Flash EEPROM of the card using a "firmware manager" tool.
- **Step 2**: The new firmware is activated by cycling the system power.

Firmware Manager Tool

The Coaxlink Driver is delivered with a command-line tool named `coaxlink-firmware.exe`.

The Coaxlink Firmware Manager Tool embeds a library of *firmware images* composed with all the *firmware variants* of all the products supported by the specific version the Coaxlink Driver. Unless specified with a `--firmware=FILE` option, the tool uses the embedded library.

The Coaxlink Firmware Manager Tool allows to:

- List the properties of the firmware installed on each Coaxlink cards present in the system including *version number*, *variant number and name*.
- Perform a firmware upgrade (without changing the variant number).
- Install a new firmware variant.

Note: *Executing `coaxlink-firmware.exe --help` displays a help message describing all the command options.*

Firmware Upgrade

To upgrade the firmware with the latest version:

- **Step 1.** Execute the following command line: `coaxlink-firmware.exe upgrade [OPTIONS]`

- **Step 2.** Reboot the PC!

IMPORTANT NOTE

The Coaxlink Firmware Manager Tool must be executed when using any Coaxlink card for the first time with a specific Coaxlink Driver version.

In particular, the firmware upgrade must be executed:

- when using a new Coaxlink card for the first time
- after installing a different version of the Coaxlink Driver

This will ensure that the Coaxlink card's firmware is suitable for the Coaxlink Driver version.

Firmware Variant Installation

To install another firmware variant:

- **Step 1.** Execute the following command line: `coaxlink-firmware.exe upgrade [OPTIONS] VARIANT`
- **Step 2.** Reboot the PC!

IMPORTANT NOTE

The Coaxlink Firmware Manager Tool must be executed when another firmware variant must be installed.

For instance:

- when another camera connection topology is required (e.g. 1-camera vs. 2-camera on Coaxlink Duo)
- when another camera type is required (e.g. 1-camera vs. 1-camera, line-scan on Coaxlink Quad)

Product and Firmware Variant Selector

Coaxlink product and firmware variant selection guide.

The following table provides help for the selection of the appropriate Coaxlink product and firmware variant regarding the following CoaXPress Host Interface requirements:

- **Device Type**: the type of camera (line- or area-scan, data-forwarded virtual camera or real camera)
- **#Dev.**: Maximum number of devices per Coaxlink card
- **#Con.**: Number of camera connections

Device Type	#Dev.	Icon	Firmware Variant	#Con.	Applicable products
Area-scan camera	1		1-camera	1, 2, or 4	Quad, Quad G3, Quad G3 DF
				1 or 2	Duo, Duo PCIe/104
				1	Mono
	1 or 2		2-camera	1 or 2	Quad, Quad G3
				1	Duo, Duo PCIe/104
	Up to 4		4-camera	1	Quad G3
Line-scan camera	1		1-camera, line-scan	1, 2, or 4	Quad, Quad G3, Quad G3 DF
				1 or 2	Duo PCIe/104
	1 or 2		2-camera, line-scan	1 or 2	Quad G3
Data-forwarded virtual area-scan camera	1		1-df-camera	1, 2, or 4	Quad G3 DF
Data-forwarded virtual line-scan camera	1		1-df-camera, line-scan	1, 2, or 4	Quad G3 DF
Specific 8-connection area-scan camera	1		1-slm-camera	4	Quad G3
			1-sls-camera	4	Quad G3

Coaxlink Architecture

Architecture of Coaxlink cards.

Coaxlink Block Diagram

The following diagram shows the main functional bocks of an image acquisition system including

- One Host PC.
- One Coaxlink frame grabber configured with a 1-camera firmware variant.
- One imaging device delivering a single image data stream over a CoaXPress Link.
- One illumination device, one encoder device and, one encoder device.

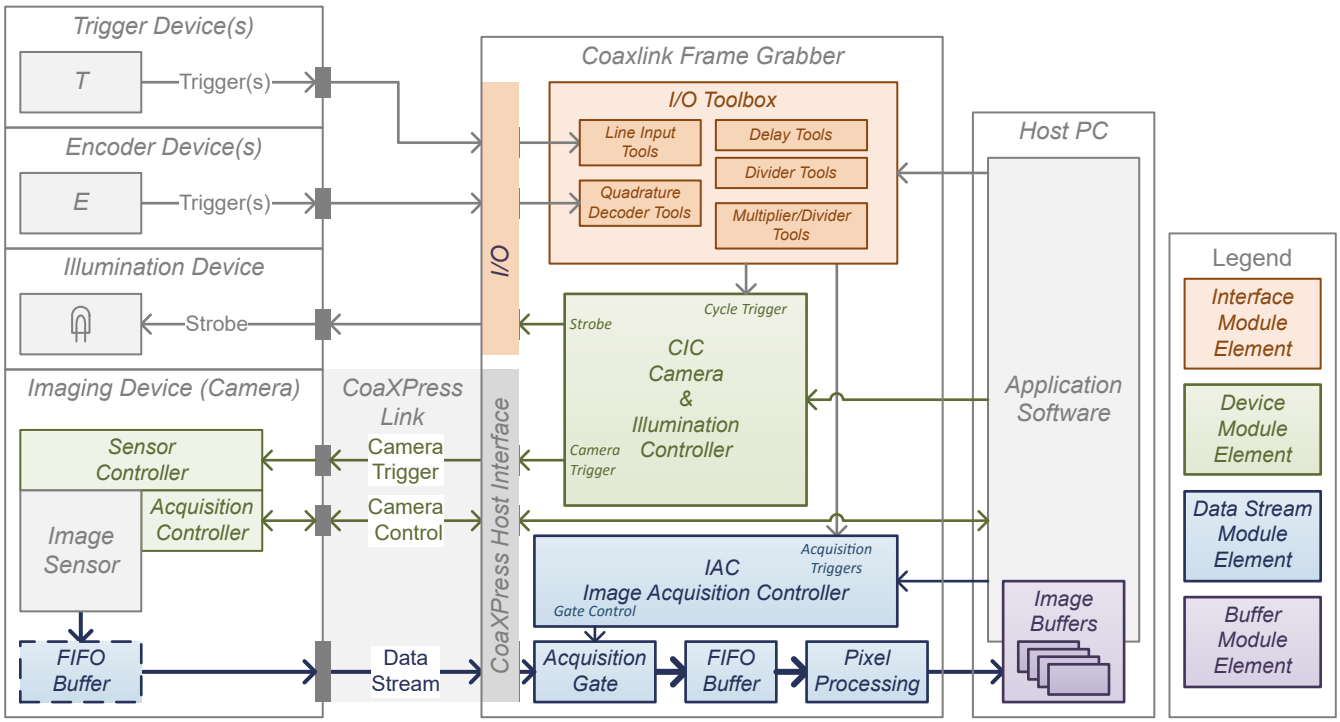


Figure 1: Single-stream single-device Coaxlink image acquisition system

The drawing uses color shading to indicate to which GenTL module the functional elements belong:

- Orange color for elements belonging to the Interface module
- Green color for elements belonging to the Device module and the associated Remote Device (camera)
- Blue color for elements belonging to the Data Stream module
- Purple color for elements belonging to the Buffer module.

Interface Module Elements

The GenTL hierarchy exposes one GenTL Interface Module for each Coaxlink card.

It gathers the elements that are common to- (or can be shared by-) all the GenTL Device Modules managed by the Coaxlink card.

The interface module contains the following elements:

- **I/O Lines:** This block gathers all the I/O ports of the card. It is used for the configuration (direction, polarity, filter strength) and reading the status of each I/O line.
- **I/O Toolbox:** This block gathers a collection of tools used to build event streams from trigger and encoder devices attached to the I/O port inputs.

Related Information

General Purpose I/O Lines on page 122
Set of general purpose digital I/O lines.

I/O Toolbox on page 95
Set of tools for building-up and processing hardware events.

Device Module Elements

The GenTL hierarchy exposes one GenTL Device Module for each for each imaging device.

It gathers the elements that are common to- (or can be shared by-) all the GENTL Data Stream Modules related to that imaging device.

The elements belonging to the imaging device (camera) itself are referred as "Remote Device". By opposition, the elements belonging to the frame grabber are also referred as "Local Device".

The device module contains the following elements:

- **Camera and Illumination Controller:** This block is used to control the camera cycle and the illumination strobe. It can be configured to receive real-time (Camera) Cycle trigger events from any I/O Toolbox output stream. It produces two real-time signals: the Camera Trigger signal, sent to the camera trigger input, and the Strobe signal, sent to the illumination device associated with the camera.

Note: *The number of Device modules per Coaxlink card is determined by the installed firmware variant.*

Related Information

Camera and Illumination Controller on page 66
Full description of the Coaxlink Camera and Illumination Controller.

Product and Firmware Variant Selector on page 16
Coaxlink product and firmware variant selection guide.

Data Stream Module Elements

The GenTL hierarchy exposes one GenTL Data Stream Module for each image data stream.

It gathers the elements involved into the image build-up and transport from the imaging device to a pool of GenTL buffers:

- **Image Acquisition Controller:** This block is used for the control of the acquisition gate. It can be configured to receive real-time start-of-scan and end-of-scan trigger events from any I/O Toolbox output stream.

- **Acquisition Gate:** This block filters the data input stream, selecting frames (area-scan imaging) or lines (line-scan imaging) to be acquired
- **FIFO Buffer:** This block provides a temporary storage for the acquired image data.
- **Pixel Processing:** This block performs pixel data formatting.

Note: *The number of Data Stream modules per Device is determined by the installed firmware variant.*

Buffer Module Elements

The GenTL hierarchy exposes one GenTL Interface Module for each image buffer.

CoaXPress Host Interface

Specification and functional description of the CoaXPress Host interface.

CoaXPress Standard Compliance

Coaxlink products together with Coaxlink Driver version 5.0 or higher complies with the following versions of the CoaXPress standard:

- CoaXPress Standard 1.0
- CoaXPress Standard 1.1
- CoaXPress Standard 1.1.1 (Partially)

The following standard deviations applies:

- **All versions** Restrictions to the camera connection schemes. Refer to *Camera Connection Schemes* on page 23
- **For 1.1.1 only** Partial implementation of the 1.1.1 discovery. Refer to: *CoaXPress 1.1.1 Discovery* on page 26

The following GenICam features can be used to report the current support for a specified CoaXPress Standard Revision

CxpRevisionSelector		
Interface	Root > CoaXPress	Enumeration (RW)

Selects the CoaXPress Standard Revision for current support.

Possible values:

- `CXP_1_0`: CoaXPress Standard Version 1.0. **Default value.**
- `CXP_1_1`: CoaXPress Standard Version 1.1.
- `CXP_1_1_1`: CoaXPress Standard Version 1.1.1.

CxpRevisionSupport		
Interface	Root > CoaXPress	Enumeration (RO)

Reports the current support of the selected CoaXPress Standard Revision.

Possible values:

- `NotSupported`: Not supported.
- `PartiallySupported`: Partially supported.
- `Supported`: Supported.

Host Interface Specification Summary

CoaXPress Host Interface specification summary in tabular format.

Specification Summary – Card Level

Specifications having the scope of one Coaxlink card (one GenTL Interface module).

Characteristic	Value	Applies to:
CoaXPress Feature Bar	CXP-6 DIN 1	Mono
	CXP-6 DIN 2	Duo Duo PCIe/104
	CXP-6 DIN 4	Quad Quad G3 Quad G3 DF
Down-connection speeds	1.250, 2.500, 3.125, 5.000, and 6.250 GT/s	All products
Maximum aggregated input data rate	625,000 Bytes/s	Mono
	1,250,000 Bytes/s	Duo Duo PCIe/104
	2,500,000 Bytes/s	Quad Quad G3 Quad G3 DF
Maximum deliverable PoCXP power	17 W per connector	All products
Devices Count	1	
	1 or 2	
	1 up to 4	

Note: *Devices Count is determined only by the installed firmware variant.*

Related Information

Firmware Variants per Product on page 12

List of available firmware variants per product in Coaxlink Driver 6.1.

Specification Summary – Camera Level

Specifications having the scope of one camera attached to the Coaxlink card (one GenTL Device module).

Characteristics	Value	Applicable Firmware Variants
Device Type	Area-scan camera	
	Line-scan camera	
	Area-scan data-forwarded virtual camera	
	Line-scan data-forwarded virtual camera	
Connections Count	1	See notes
	1 or 2	
	1, 2 or 4	
Data Streams Count	1	All firmware-variants

Note: *Device Type is determined only by the installed firmware variant.*

Note: *Connections Count is determined by the product/firmware-variant combination.*

Related Information

Firmware Variants per Product on page 12
List of available firmware variants per product in Coaxlink Driver 6.1.

Specification Summary – Image Data Stream Level

Specifications having the scope of one image data stream (one GenTL Data Stream module).

Characteristic	Values	Applicable Firmware Variants
Maximum stream packet size	16,384 Bytes	All firmware variants
Image Stream Format	Rectangular Image (YSize > 0)	
	Rectangular Image (YSize = 0)	
Image Scanning Method	Progressive	All firmware variants
Tap Geometry	1X-1Y	All firmware variants
Pixel Format	Raw	All firmware variants

Characteristic	Values	Applicable Firmware Variants
	Mono8, Mono10, Mono12, Mono14, Mono16	All firmware variants
	BayerGR8, BayerRG8, BayerGB8, BayerBG8 BayerGR10, BayerRG10, BayerGB10, BayerBG10 BayerGR12, BayerRG12, BayerGB12, BayerBG12 BayerGR14, BayerRG14, BayerGB14, BayerBG14 BayerGR16, BayerRG16, BayerGB16, BayerBG16	All firmware variants
	RGB8, RGB10, RGB12, RGB14, RGB16	All firmware variants
	RGBA8, RGBA10, RGBA12, RGBA14, RGBA16	All firmware variants

Note: *Image Stream Format is determined only by the installed firmware variant.*

Related Information

Firmware Variants per Product on page 12
List of available firmware variants per product in Coaxlink Driver 6.1.

Camera Connection Schemes

CoaXPress Host to Devices connections schemes.

The Host Interface of Coaxlink requires a specific assignment of the Device connections to the Host connectors.

Important: *This is a deviation from the CoaXPress standard rule indicating that Devices (cameras) can be connected to the Host (frame grabber) using a free connection scheme.*

Host Connection Map Naming Convention

Syntax used for naming Host Connection Maps (how the connections of the Host Interface of a card are allocated to the Devices).

A Host Connection Map is designated by an acronym using the following Euresys proprietary naming convention:

- <dev-count><dev-type><connection-count-dev$_0$>...<connection-count-dev$_{d-1}$>
- <dev-count> = {1, 2, 4} Number of Devices that can be attached to the Host Interface
- <dev-type> = {D, DF} Type of Devices: D for standard CoaXPress Devices, DF for proprietary Data Forwarding Devices

<connection-count-dev$_i$> = {1, 2, 4}Number of connections available for the Device i.

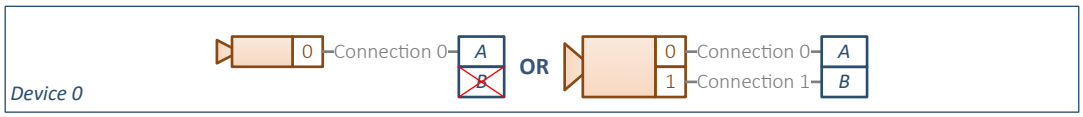
Example: **2D11** designates a Host Connection Map of a Host Interface with 2 standard CoaXPress Devices, each having 1 CoaXPress connection.

Host Connection Maps – Standard CoaXPress Devices

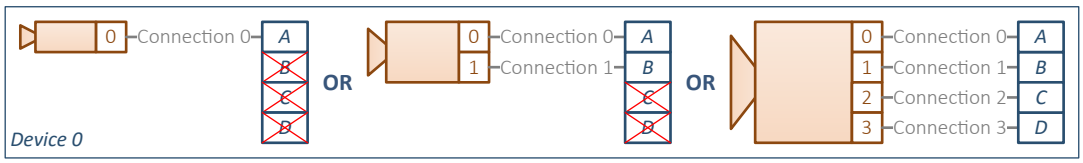
Coaxlink connection schemes for standard CoaXPress devices.

1D1 Connection Map

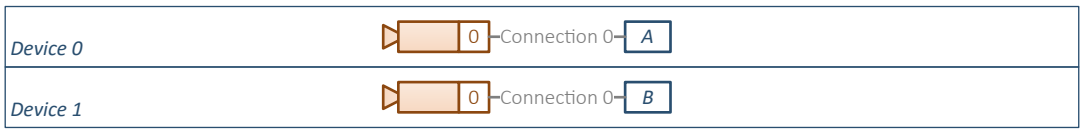
1D2 Connection Map



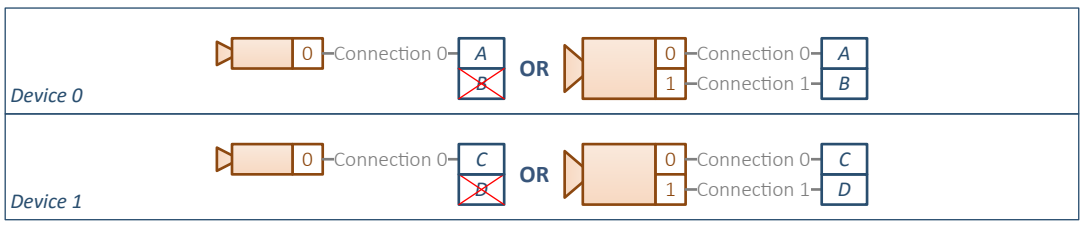
1D4 Connection Map



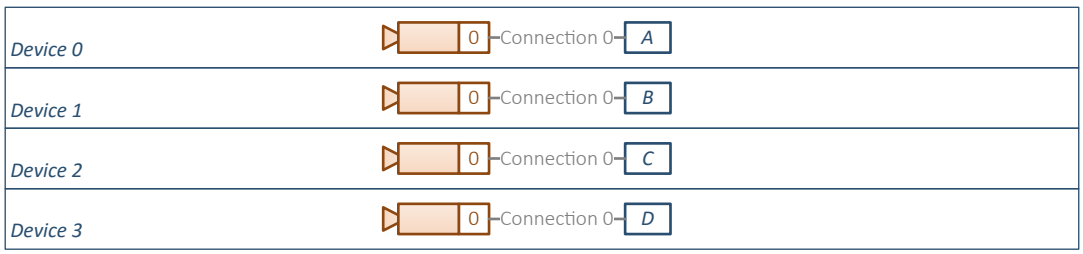
2D11 Connection Map



2D22 Connection Map



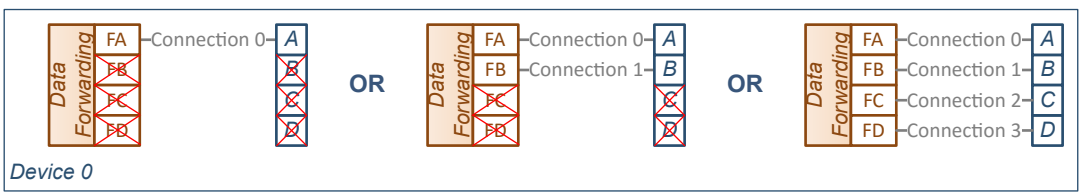
4D1111 Connection Map



Host Connection Maps – Data Forwarding Devices

Coaxlink connection schemes for proprietary Data Forwarding Devices.

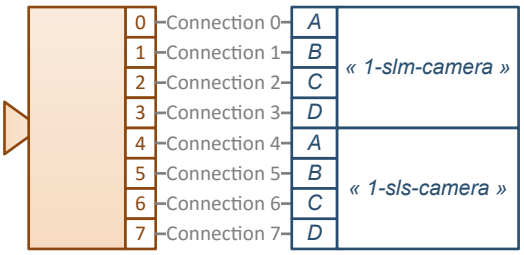
1DF4 Connection Map



Host Connection Maps – Sub-Link Devices

Coaxlink connection schemes for CoaXPress devices using sub-links.

1D8SL4 Connection Map



Firmware Variant & Product vs. Host Connection Map

The Coaxlink product and firmware variant must be selected according to the required Host Connection Map:

Host Connection Map	Firmware variant(s)	Product(s)
1D1		Mono
1D2		Duo
		Duo PCIe/104
1D4		Quad Quad G3 Quad G3 DF
1DF4		Quad G3 DF
1D8SL4 New		Quad G3
2D11		Duo Duo PCIe/104
2D22		Quad Quad G3
		Quad G3
4D1111		Quad G3

Related Information

Firmware Variants per Product on page 12

List of available firmware variants per product in Coaxlink Driver 6.1.

Link Configuration

CoaXPress Link configuration and discovery.

Automatic Link Configuration

The Coaxlink Driver provides an automatic link discovery and configuration for CoaXPress 1.0 and CoaXPress 1.1 devices.

For each connection of the CoaXPress Host interface, the discovery procedure determines:

- The presence of a CoaXPress Device
- The speed of the down-connection (Device to Host)
- The connection ID

The discovery results are reported through the `CxpConnectionState`, `CxpDownConnectionSpeed` and `CxpDeviceConnectionID` features of the Interface module.

The user is invited to check if the resulting link configuration is appropriate:

- For the application needs in terms of link bandwidth (link speed and number of connections)
- For Coaxlink in terms of camera connection schemes supported by the target Coaxlink product/firmware combination

Manual Link Configuration

If necessary, the user can manually configure the CoaXPress link of the Remote Device.

This can be achieved, regardless of the camera brand, by assigning the appropriate value to the `CxpLinkConfiguration` GenApi feature of the Coaxlink device module.

Note: *Assigning the value `Preferred` enforces the preferred link configuration of the camera:*

- *The link speed is set to the specified value.*
- *The link width is set to the specified value but, possibly limited to the number of available connections on the Coaxlink side.*

CoaXPress 1.1.1 Discovery

The **10.1.3 Discover Devices and Connection Topology** paragraph of the CoaXPress 1.1.1 standard claims:

"The Host shall read the `ConnectionConfigDefault` register to find the number of expected connections. It shall then write to the `ConnectionConfig` register to enable the number of connections read from `ConnectionConfigDefault`. However it shall not change from the discovery rate at this stage."

The **10.3.33 ConnectionConfig** paragraph of the CoaXPress 1.1.1 standard claims:

"This register shall hold a valid combination of the Device connection speed and number of active downconnections. Writing to this register shall set the connection speeds on the specified connections, and the high speed upconnection, if supported. If the new ConnectionConfig value results in a change of connection speed, the Device shall acknowledge the ConnectionConfig access at the original connection speed. Therefore it shall acknowledge the access before changing connection speed."

Note: *Not all theoretical combinations of connection speed and number of connections may be usable. One register is used to ensure that the two variables are set simultaneously. The XML file and product documentation give valid combinations for the Device. A connection reset sets the value corresponding to the selected discovery rate and one connection.*

Considering that:

- The above paragraphs disagree on the value that should be written to the `ConnectionConfig` register,

- Changing the behavior to respect CoaXPress 1.1.1 statements causes issues with some cameras,

the CoaXPress discovery procedure of the Coaxlink Driver is not modified to comply with CoaXPress 1.1.1.

At the end of the discovery procedure, the Coaxlink Driver sets the speed and the number of the connections of the CoaXPress Link according to the settings of `ConnectionConfigDefault` register of the camera.

CoaXPress Link GenICam Features

GenICam features related to the connections of CoaXPress Links.

CxpHostConnectionCount		
Interface	Root > CoaXPress	Integer (RO)

Reports the number of CoaXPress physical Host connections.

Possible values: 1, 2, 4.

CxpHostConnectionSelector		
Interface	Root > CoaXPress	Enumeration (RW)

Selects one CoaXPress physical Host connection.

Possible values:

- `A`: CoaXPress physical Host connection A. **Default value.**
- `B`: CoaXPress physical Host connection B.
- `C`: CoaXPress physical Host connection C.
- `D`: CoaXPress physical Host connection D.

Note: *refer to Host Interface Specification Summary on page 21 for connector availability.*

CxpConnectionState		
Interface	Root > CoaXPress	Enumeration (RO)

Returns the CoaXPress connection state of the CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

Possible values:

- `Undetected`: No device detected on the selected connection.
- `Detected`: Device detected on the selected connection.

CxpDownConnectionSpeed		
Interface	Root > CoaXPress	Enumeration (RO)

CxpDownConnectionSpeed

Returns the CoaXPress down-connection speed of the CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

Possible values:

- `CXP1`: 1.25 Gbps.
- `CXP2`: 2.5 Gbps.
- `CXP3`: 3.125 Gbps.
- `CXP5`: 5 Gbps.
- `CXP6`: 6.25 Gbps.

CxpDeviceConnectionID

Interface	Root > CoaXPress	Enumeration (RO)

Returns the CoaXPress connection topology information of the CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

Possible values:

- `CameraW_Master`: Master Connection of Camera W.
- `CameraW_Extension1`: Extension 1 of Camera W.
- `CameraW_Extension2`: Extension 2 of Camera W.
- `CameraW_Extension3`: Extension 3 of Camera W.
- `CameraX_Master`: Master Connection of Camera X.
- `CameraX_Extension1`: Extension 1 of Camera X.
- `CameraX_Extension2`: Extension 2 of Camera X.
- `CameraX_Extension3`: Extension 3 of Camera X.
- `CameraY_Master`: Master Connection of Camera Y.
- `CameraY_Extension1`: Extension 1 of Camera Y.
- `CameraZ_Master`: Master Connection of Camera Z.
- `NotReady`: Remote device not ready

Note: *Since Coaxlink Driver version 3.2, the discovered devices reported by `CxpDeviceConnectionID` are designated Camera[WXYZ] instead of Device[1234]*

CxpLinkConfiguration

Device	Root > CoaXPress	Enumeration (RW)

CxpLinkConfiguration

Set/report the CoaXPress Link configuration.

Possible values:

- `CXP1_X1`: 1 connection @1.250 Gbps.
- `CXP1_X2`: 2 connections @1.250 Gbps.
- `CXP1_X4`: 4 connections @1.250 Gbps.
- `CXP2_X1`: 1 connection @2.500 Gbps.
- `CXP2_X2`: 2 connections @2.500 Gbps.
- `CXP2_X4`: 4 connections @2.500 Gbps.
- `CXP3_X1`: 1 connection @3.125 Gbps.
- `CXP3_X2`: 2 connections @3.125 Gbps.
- `CXP3_X4`: 4 connections @3.125 Gbps.
- `CXP5_X1`: 1 connection @5.000 Gbps.
- `CXP5_X2`: 2 connections @5.000 Gbps.
- `CXP5_X4`: 4 connections @5.000 Gbps.
- `CXP6_X1`: 1 connection @6.250 Gbps.
- `CXP6_X2`: 2 connections @6.250 Gbps.
- `CXP6_X4`: 4 connections @6.250 Gbps.
- `Preferred`: Camera preferred configuration.

CxpHostConnectionBase

Device	Root > CoaXPress	Enumeration (RO)

Returns the base CoaXPress physical connection of this device.

Possible values:

- `A`: CoaXPress physical host connection A.
- `B`: CoaXPress physical host connection B.
- `C`: CoaXPress physical host connection C.
- `D`: CoaXPress physical host connection D.

CxpHostConnectionCount

Device	Root > CoaXPress	Integer (RO)

Returns the number of CoaXPress physical connections of this device.

Value range: 1, 2, 4.

CxpPacketArbiterReset

Device	Root > CoaXPress	Command (RO)

CoaXPress Data Packet Arbiter Reset.

Power Over CoaXPress

CoaXPress camera powering through the CoaXPress Link connections.

Each connection of the CoaXPress Host connector is capable of delivering power to the camera through the CoaXPress cable.

The Power Transmitting Unit – PTU – is responsible for a safe delivery of power. Therefore, it fulfils all the requirements of the CoaXPress standard for a CoaXPress Host, namely:

- It is capable of delivering up to 17 W of 24 V DC power to the connector
- It implements an over-current protection device – OCP
- It supports the CoaXPress PoCXP detection method

In addition, it provides the application with the capability of:

- Controlling the PoCXP automatic detection
- Disabling or interrupting the power delivery
- Resetting the OCP when tripped

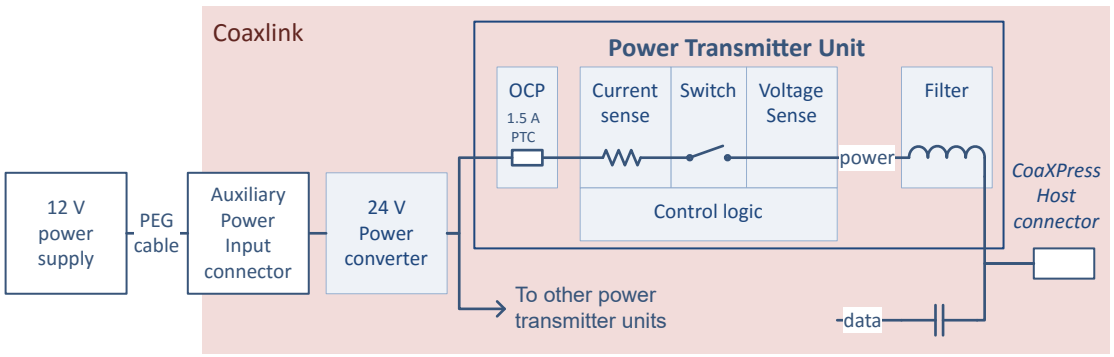


Figure 2: Power Over CoaXpress functional diagram

Power Source

All the PTU's of a Coaxlink card are powered by an external 12 V power supply through a 12 V to 24 V power converter on the Coaxlink card. The external 12 V supply is attached to the **auxiliary power input connector** through a 6-pin PEG cable.

The `AuxiliaryPowerInput` feature reports the status of the connection made by the PEG cable between the external power supply and the Coaxlink auxiliary power input connector.

The `CxpPoCxpPowerInputStatus` feature reports the status of the 24 V power converter.

Note: *The 12 V supply is typically delivered by the power supply of the Host PC.*

PTU Control Logic

On execution of the `CxpPoCxpAuto` command the PTU controller initiates a **PoCXP device detection procedure**.

If the **PoCXP device detection procedure terminates successfully**, the PTU applies power by closing the switch.

If the **PoCXP device detection procedure fails**, the controller doesn't apply power and retries a new PoCXP detection procedure. Possible causes of failure are:

- The external power is not connected (`AuxiliaryPowerInput = Unconnected`)
- The external power source is off (`CxpPoCxpPowerInputStatus = NotOK`)
- There are no camera attached
- The attached camera is not PoCXP compliant

Once the power is applied, the controller remains in that state until any of the following situations occurs:

- The application disables the power delivery by executing the `PoCxpTurnOff` command.
- The external power source is disconnected (`CxpPoCxpPowerInputStatus = NotOK`)
- The external power source is turned off (`CxpPoCxpPowerInputStatus = NotOK`)
- The CoaXPress cable is disconnected (The average output current measured over a time interval of 0.3 seconds is less than 8 mA)
- The OCP trips (The output voltage drops for a time interval greater than 20 milliseconds)

On execution of the `CxpPoCxpTurnOff` command the PTU turns off the switch and disables PoCXP powering. In that state, the PTU is not performing PoCXP detection procedures.

The `CxpPoCxpConfigurationStatus` feature reports the configuration status of the PTU: `Off` or `AUTO`

The `CxpPoCxpStatus` feature reports the status of the PTU: `Off`, `On` or, `Tripped`.

IMPORTANT NOTE

For users of early versions of the Coaxlink Driver

Since version 3.1 of the Coaxlink Driver, PoCXP powering is enabled at system power-up.

Consequently, when using PoCXP powered camera(s), the application is not anymore required to enable PoCXP powering by issuing a `CxpPoCxpAuto` command.

Over-current Protection

The OCP circuit is built with a PTC device providing two kind of protections:

- The overload protection addresses the cases when the load is excessive.
- The short-circuit protection addresses the cases of accidental short-circuits.

In case of overload, the PTC trips (= opens progressively the circuit) after several seconds or minutes depending on the current level and the ambient temperature. The higher the current, the lower the time to trip. The same applies to the ambient temperature.

In case of short-circuit, the PTC trips immediately. Consequently, the PTU controller enters the tripped state and opens the switch. The tripped PTC device returns to the conducting state after having cooled down. This may take a few seconds. However, the PTU controller remains in the tripped state until the application issues a `CxpPoCxpTripReset` command. Having left the tripped state, the PTU can initiate a new PoCXP device detection and, if successful, re-establish power.

Note: *The PTC is sized to sustain 17 W of power over the whole operating temperature range without tripping.*

Note: *Extracting more than 17 W of power and/or operating the Coaxlink card above the operating temperature range is prohibited since it may induce unexpected PTC trips.*

Power Over CoaXPress GenICam Features

GenICam features related to Power over CoaXPress.

CxpPoCxpHostConnectionSelector		
Interface	Root > CoaXPress	Enumeration (RW)

CxpPoCxpHostConnectionSelector

Selects one (or a group of) CoaXPress physical Host connection(s) for PoCXP control.

Possible values:

- `All`: All CoaXPress physical Host connections. **Default value.**
- `A`: CoaXPress physical Host connection A.
- `B`: CoaXPress physical Host connection B.
- `C`: CoaXPress physical Host connection C.
- `D`: CoaXPress physical Host connection D.

Note: *refer to Host Interface Specification Summary on page 21 for available connectors.*

CxpPoCxpAuto

Interface	Root > CoaXPress	Command (WO)

Activates automatic control of the Power over CoaXPress (PoCXP) on the CoaXPress physical Host connection(s) designated by `CxpPoCxpHostConnectionSelector`.

CxpPoCxpTurnOff

Interface	Root > CoaXPress	Command (WO)

Disables Power over CoaXPress (PoCXP) on the CoaXPress physical Host connection(s) designated by `CxpPoCxpHostConnectionSelector`.

CxpPoCxpTripReset

Interface	Root > CoaXPress	Command (WO)

Resets Power over CoaXPress (PoCXP) after an over-current trip on the CoaXPress physical Host connection(s) designated by `CxpPoCxpHostConnectionSelector`.

CxpPoCxpConfigurationStatus

Interface	Root > CoaXPress	Enumeration (RO)

Returns the Power over CoaXPress (PoCXP) configuration of the CoaXPress physical Host connection(s) designated by `CxpPoCxpHostConnectionSelector`.

Possible values:

- `Off`: PoCXP is forced off.
- `Auto`: Normal automatic PoCXP operation.
- `Unknown`: PoCXP configuration is unknown.
- `Compound`: PoCXP configuration is compound. The selected connections have different configuration states.

CxpPoCxpStatus

Interface	Root > CoaXPress	Enumeration (RO)

CxpPoCxpStatus

Returns the Power over CoaXPress (PoCXP) status on the CoaXPress physical Host connection(s) designated by `CxpPoCxpHostConnectionSelector`.

Possible values:

- `Off`: PoCXP is off.
- `On`: PoCXP is on.
- `Tripped`: PoCXP has shut down because of an over-current trip.
- `Compound`: PoCXP status is compound. The selected connections have different PoCXP statuses.

CxpPoCxpCurrent

Interface	Root > CoaXPress	Float (RO)

Returns the current delivered by the PoCXP transmitter unit of the CoaXPress physical Host connection designated by `CxpPoCxpHostConnectionSelector`.

Unit: Ampere.

Value range: from 0.0 up to 1.020 by steps of 0.004.

CxpPoCxpVoltage

Interface	Root > CoaXPress	Float (RO)

Returns the output voltage delivered by the PoCXP transmitter unit of the CoaXPress physical Host connection(s) designated by `CxpPoCxpHostConnectionSelector`.

Unit: Volt.

Value range: from 21.0 up to 29.16 by steps of 0.032.

CxpPoCxpPowerInputStatus

Interface	Root > CoaXPress	Enumeration (RO)

Returns the status of the 24 V power converter delivering power to all PoCXP transmitter units.

Possible value(s):

- `NotOK`: The 24 V power converter is not OK.
- `OK`: The 24 V power converter is OK.

CoaXPress I/O Channel

CoaXPress logical channel for real-time triggers and general purpose I/O.

According to the CoaXPress 1.0 and 1.1 standards, the CoaXPress I/O Channel:

- Is one of the three logical channels of the CoaXPress Link (I/O, Stream, Control)
- Is defined only for the master connection (connection 0) of a CoaXPress Link
- Is used for transmitting of high-priority "triggers" between the Host and the Device
- On CoaXPress 1.0 only, is used for exchanging the state of General Purpose I/Os between the Host and the Device

Coaxlink implements:

- CoaXPress Host to Device trigger

Coaxlink doesn't implement:

- CoaXPress Device to Host trigger
- CoaXPress 1.0 GPIO

CoaXPress Host To Device Trigger

Real-time Host-to-device triggering though the CoaXPress Link.

The CoaXPress Host To Device Trigger is a functionality of the CoaXPress I/O Channel that allows the Host (frame grabber) to trigger the Device (camera) through the CoaXPress Link.

The CoaXPress Host Interface of Coaxlink implements one CoaXPress Host to Device trigger transmitter for **each connected Device**.

The transmitter can be sourced by:

- The Camera Trigger output of the associated Camera and Illumination Controller
- Any input-capable General Purpose I/O

Host to Device Trigger Transmitter

The Coaxlink implementation of the CoaXPress Host to Device Trigger transmitter complies with the requirements of the CoaXPress 1.0 and 1.1 standards for a low-speed CoaXPress Host to Device Trigger.

Default Operation

This section describes the behaviour of the transmitter when it is configured with the default settings:

- `CxpTriggerMessageFormat = Pulse`
- `CxpTriggerAckTimeout = 20.0`
- `CxpTriggerMaxResendCount = 3`

The transmitter **initiates** a **trigger transaction** on both edges of the trigger source signal:

- It computes a delay value allowing the receiving device to recreate the event with a fixed latency. For more information, refer to: *Host to Device Trigger Latency Compensation* on page 36.
- It inserts a high-priority "trigger packet" on the low-speed host-to-device connection at the next character boundary.

Then, the transmitter waits for the **acknowledgment** from the Device (camera):

- If the acknowledgement is received before the expiration of the timeout, the transaction terminates normally.
- If no acknowledgement is received within the 20 µs timeout, the transmitter performs a **retry**: it resends the trigger packet and initiates a new waiting period for the acknowledgement.
- If no acknowledgement is received after 3 times, the transaction terminates abnormally.

The transmitter doesn't initiate a new transaction while the previous one is not completed.

Customization

The transmitter can be customized:

- To send trigger messages only on the rising edge of the source signal. Refer to *Host to Device Trigger Message Format* on page 35
- To configure the acknowledge timeout and the number of retries. Refer to *Host to Device Trigger Packet Acknowledgement* on page 35

Events Reporting

The transmitter reports the following events:

- `CxpTriggerAck`: Received acknowledgement for CoaXPress Host to Device trigger packet.
- `CxpTriggerResend`: Resent CoaXPress Host to Device trigger packet.

Host to Device Trigger Message Format

The Host to Device Trigger transmitter unit of Coaxlink provides a "message format" control with the `CxpTriggerMessageFormat` GenICam feature.

Default Settings

By default, `CxpTriggerMessageFormat` is set to `Pulse`.

When `CxpTriggerMessageFormat` is set to `Pulse`, the transmitter initiates a CoaXPress I/O Channel Host to Device Trigger transaction on **both edges** of the input pulse:

- The transaction initiated by the rising edge transmits a **rising edge trigger packet** from the Host to the Device.
- The transaction initiated by the falling edge transmits a **falling edge trigger packet** from the Host to the Device.

> ***Note:*** *Every trigger pulse requires two distinct CoaXPress I/O Channel transactions!*

Alternate Settings

When `CxpTriggerMessageFormat` is set to `RisingEdge`, the transmitter initiates a CoaXPress I/O Channel Host to Device Trigger transaction on **the rising edge only** of the input pulse. The transaction transmits a **rising edge trigger packet** from the Host to the Device.

> ***Note:*** *Every trigger pulse requires a single CoaXPress I/O Channel transaction.*

> ***Note:*** *This format doesn't allow the grabber to control the exposure time!*

Use Cases

The `Pulse` settings, is the only one that strictly complies with the CoaXPress standards, it is expected to operate correctly with most of the cameras.

The `Pulse` settings, is the only one that is applicable when the trigger pulse width defines the exposure; namely when `CameraControlMethod` is set to `RG` or `EXTERNAL`.

The `RisingEdge` format is a non-standard format used by some high-speed line-scan camera products to reach higher line rate (> 100 kHz).

Host to Device Trigger Packet Acknowledgement

The Host to Device Trigger transmitter unit provides a user-configurable trigger packet acknowledgement mechanism:

- The time-out value is configurable using the `CxpTriggerAckTimeout` GenICam feature.
- The number of retries is configurable using the `CxpTriggerMaxResendCount` GenICam feature.

Default Settings

By default, `CxpTriggerAckTimeout` is set to `20.0` (20 microseconds) and `CxpTriggerMaxResendCount` is set to 3. Coaxlink expects an I/O Channel Acknowledgement packet in response to every Trigger packet. If the acknowledgement packet is not received within the 20 µs time-out value, the transmitter resends the trigger packet. It performs up to 3 retries.

Alternate Settings

Setting `CxpTriggerAckTimeout` to `0` disables the acknowledgement mechanism. The trigger transaction terminates immediately after having sent the trigger packet.

Setting larger `CxpTriggerAckTimeout` values allows more time for the Device to acknowledge the trigger packet.

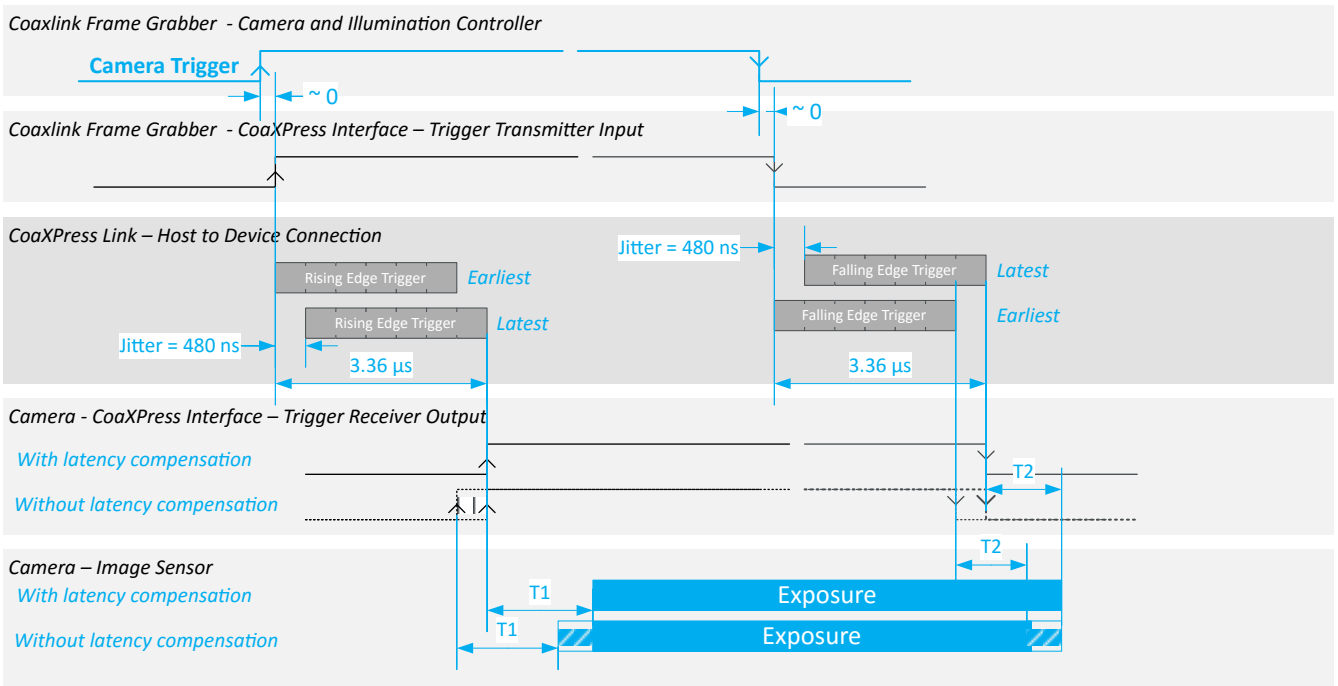
Host to Device Trigger Latency Compensation

The Host to Device trigger packets are transmitted over the low-speed (20.833 Mbps) connection of the CoaXPress Link at the next character boundary.

To minimize trigger jitter, the time between the trigger event and the trigger packet being sent is encoded into the trigger packet as a delay value expressed in units of 2 ns (1/24th of the bit period). The receiver can then use this delay to recreate the trigger event with low jitter and a fixed latency.

Assuming that the Device (the camera) performs the jitter compensation, the trigger packets are transmitted with a **fixed latency of 3.4 microseconds ± 4 nanoseconds**.

Camera Trigger to Exposure Latency



The above diagram shows the time delay required to propagate Camera Trigger events from the frame grabber up to the image sensor through the CoaXPress Link using Host to Device CoaXPress Trigger messages.

The above diagram assumes that:

- `CameraControlMethod` is set to `RG`.
- The camera properly acknowledges the trigger messages and effectively initiates a new exposure.

The delay from the rising edge of the Camera Trigger signal up the start of exposure on the image sensor is composed of 3 elements:

- Coaxlink frame grabber: a negligible delay from the CIC output up to the transmitter

- CoaXPress Link: a fixed delay of 3.36 µs if the camera compensates the jitter using the method described by the CoaXPress standard otherwise a variable delay of 2.88 ~ 3.36 µs.
- Camera: a delay T1 which corresponds to the time required by the camera to initiate a new exposure. This time is camera-dependent.

The delay from the falling edge of the Camera Trigger signal up the end of exposure on the image sensor is composed of 3 elements:

- Coaxlink frame grabber: a negligible delay from the CIC output up to the transmitter
- CoaXPress Link: a fixed delay of 3.36 µs if the camera compensates the jitter using the method described by the CoaXPress standard otherwise a variable delay of 2.88 ~ 3.36 µs.
- Camera: a delay T2 which corresponds to the time required by the camera to terminate an exposure. This time is camera-dependent and likely different than T1.

Host to Device Trigger Source

The CoaXPress Host to Device Trigger transmitter can be sourced from:

- The Camera Trigger output of the associated Camera and Illumination Controller
- Any input-capable General Purpose I/O

The trigger source is indirectly controlled through the `CameraControlMethod` GenICam feature.

When `CameraControlMethod` is set to `RG` or `RC`, the trigger source is the Camera Trigger output of the associated Camera and Illumination Controller.

When `CameraControlMethod` is set to `EXTERNAL`:

- The trigger source is the line source of a dedicated LIN tool of the I/O Toolbox: LIN1 for Device0, LIN2 for Device1, LIN3 for Device2, and LIN4 for Device3.
- Any input-capable GPIO line can be used as trigger source by configuring the line source – `LineInputToolSource` – of the dedicated LIN tool.
- The polarity of the external trigger signal can be controlled with the `LineInverter` setting of the selected I/O Control block.
- The time constant of the glitch-removal filter can be adjusted through the LineFilterStrength setting of the selected I/O Control block.

When `CameraControlMethod` is set to `NC`, the Host to Device Trigger transmitter is disabled.

Related Information

Line Input Tool on page 98
Line Polarity Control on page 125
Filter Control on page 126

Host to Device Trigger GenICam Features

GenICam features related to the real-time CoaXPress Host-to-device Trigger.

CameraControlMethod		
Device	Root > Camera and Illumination Control > Camera Model	Enumeration (RW)

CameraControlMethod

Sets/gets the camera control method.

Possible values:

- `NC` "NC" camera control method. *To be used with free-run or asynchronous reset cameras not controlled by the frame grabber*
- `RC`: "RC" camera control method. *To be used with asynchronous reset cameras having the camera cycle start controlled by the Coaxlink Camera and Illumination Controller and the exposure time controlled by the camera.*
- `RG`: "RG" camera control method. *To be used with asynchronous reset cameras having the camera cycle start and the exposure duration controlled by the Coaxlink Camera and Illumination Controller.*
- `EXTERNAL`: "External" camera control method. *To be used with asynchronous reset cameras having the camera cycle start and the exposure duration controlled by an external hardware signal applied on any GPIO input port.*

CxpTriggerMessageFormat

Device	Root > CoaXPress	Enumeration (RW)

Sets/gets the CoaXPress Host to Device Trigger Message Format.

Possible values:

- `Pulse`: Every Trigger requires two transactions on the Host to Device I/O Channel: one **rising edge trigger packet** and one **falling edge trigger packet**. **Default value.**
- `RisingEdge`: Every Trigger requires a single transaction on the Host to Device I/O Channel: one **rising edge trigger packet**

CxpTriggerAckTimeout

Device	Root > CoaXPress	Float (RW)

Sets/gets the acknowledge timeout value of the CoaXPress Host to Device Trigger Message.

Unit: microsecond.

Value range: from 0.0 up to 131.0.

Default value: 20.0. *(20 microseconds)*.

CxpTriggerMaxResendCount

Device	Root > CoaXPress	Integer (RW)

Sets/gets the maximum resend count of the CoaXPress Host to Device Trigger Message.

Unit: microsecond.

Value range: from 0 up to 7.

Default value: 3.

CoaXPress Host Connector Indicator Lamps

Lamps on the Coaxlink bracket indicating the state of each CoaXPress Link connection.

Each connector of the CoaXPress Host Interface is associated with a CoaXPress Host Indicator Lamp.

CoaXPress Host Connector Indicator Lamps States

Lamp State	Symbol	Meaning
Off		The Coaxlink card is not powered
Solid orange		System booting
Fast flash alternate green / orange		The connection detection is in progress; PoCXP is active. *This state is shown for a minimum of 1s even if the connection detection is faster*
Fast flash orange		The connection detection is in progress; PoCXP is off. *This state is shown for a minimum of 1s even if the connection detection is faster*
Solid red		The PoCXP over-current protection has tripped.
Solid green		The Device to Host connection is established, but no data being transferred
Slow pulse orange		The Device to Host connection is established, but the Host is waiting for a trigger.
Fast flash green.		The Device to Host connection is established and image data is being transferred

Flashing Lamp States Timing Definitions

Indication	Timing
Fast flash	**12.5Hz @25% duty cycle**: 20 ms on, 60 ms off
Fast flash alternate (color 1/color 2)	**12.5Hz @25% duty cycle**: 20 ms on (color 1), 60 ms off, 20 ms on (color 2), 60 ms off
Slow flash	**0.5Hz @50% duty cycle**: 1 second on, 1 second off
Slow pulse (red | orange)	**1Hz @ 20% duty cycle:** 200ms on, 800ms off

Connection Test

CoaXPress link connection test facility.

The CoaXPress Host Interface of Coaxlink provides connection test facilities to test the quality up- and down-connections of the CoaXPress link according to the procedures defined in section 8.7 of the CoaXPress 1.1 standard.

For each individual CoaXPress connector, it implements

- A test generator
- A test receiver

The test generator transmits a Test Data Packet containing a known test pattern produced by a sequence generator. It increments the packet counter for each test packet transmitted.

The test receiver compares the received test data packet content against its local sequence generator. It increments the error counter for each word that is different in the data packet, and increments the packet counter for each test packet received.

Note: *The test packet counters show how many test packets have been sent and received, so allowing a judgment to be made on the statistical meaning of the value in the error counter.*

Note: *Both Device to Host and Host to Device connection tests can be run at the same time.*

Connection Test GenICam Features

GenICam features related to the CoaXPress Link connections test facility.

CxpHostConnectionTestMode

Interface	Root > CoaXPress	Enumeration (RW)

Host to Device connection test mode for the CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

Possible values:

- `Off`: Test mode disabled. **Default value.**
- `Mode1`: Test mode 1.

CxpHostConnectionTestErrorCount

Interface	Root > CoaXPress	Integer (RW)

Returns/clears the current error count value for test packets received by CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

This feature can be read at any time.

It can be written to zero to reset the count between tests.

CxpHostConnectionTestPacketCount

Interface	Root > CoaXPress	Integer (RW)

Returns the current count of test packets received by the CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

This feature can be read at any time. It can be written to zero to reset the count between tests.

CxpHostConnectionTestInjectError

Interface	Root > CoaXPress	Command (WO)

Injects a single character error into the Host to Device test packet of the CoaXPress physical Host connection designated by `CxpHostConnectionSelector`.

CoaXPress Data Forwarding Quad G3 DF

Specification and functional description of the CoaXPress Data Forwarding interface.

The data forwarding capabilities of 1635 Coaxlink Quad G3 DF allows to forward the image data from a camera to multiple frame grabbers in different Host PC's.

Data Forwarding Principles

Data forwarding

A **DF-capable** card, such as the 1635 Coaxlink Quad G3 DF, forwards the data received on the CoaXPress Host connector to the CoaXPress Data Forwarding connector.

The serial bit stream on connections A, B, C, D of the CoaXPress Host connector are forwarded respectively to the connections FA, FB, FC, FD of the CoaXPress Data Forwarding connector.

Note: *The data are retransmitted with a negligible latency: typically, a few periods of the 32-bit character transmission time.*

Considering that link speeds can be different, idle characters are, when necessary, removed or added in the bit stream.

Note: *Addition or removal of idle characters doesn't affect the payload. Image data are preserved, including CRC's.*

Notice: *§8.2.5.1 of the Coaxlink 1.1 standard requires, for high-speed connections, the insertion of one IDLE word at least once every 100 words. Fulfilling this requirement is mandatory for proper operation of data forwarding.*

Data Forwarding Chain

A **DF-chain** is composed of 2 or more data-forwarding capable cards where the CoaXPress Data Forwarding connector of one card is connected to the CoaXPress Host connector of the next card using a set of 1, 2 or 4 coaxial cables named **DF-bridge**.

Note: *There are no specified upper limit to the number of cards in a DF-chain.*

The camera is attached to the CoaXPress Host connector of the first card of the DF-chain, this card is named **DF-master**. The other cards, of the DF-chain are named **DF-slaves**. The CoaXPress Data Forwarding connector of the last DF-slave card is left unconnected.

Note: *All cards of the DF-chain, including the last DF-slave, must be DF-capable.*

CoaXPress Link discovery and configuration

The DF-master card is responsible for the discovery and the configuration of the CoaXPress Link of the camera.

The CoaXPress Host Interface of the DF-slaves are automatically configured with the same number of connections as discovered by the DF-Master.

For instance if the camera uses two connections, only two connections are required for every DF-bridge.

Firmware variants

The firmware variant to install on the DF-master must be selected according to the camera type:

- For an area-scan camera, install the `1-camera` firmware variant.
- For a line-scan camera, install the `1-camera, line-scan` firmware variant.

The firmware variant to install on the DF-slaves must match the firmware variant installed on the DF-master:

- When the `1-camera` firmware variant is installed on the DF-master, install the `1-df-camera` firmware variant on all DF-slaves.
- When `1-camera, line-scan` firmware variant is installed on the DF-master, install the `1-df-camera, line-scan` firmware variant on all DF-slaves.

Data Acquisition Control

For correct operation of data forwarding, the application must respect the following rules:

- The data acquisition must be activated on all the DF-slaves before being activated on the DF-master.
- The data acquisition must be de-activated on the DF-master before being de-activated on the DF-slaves.

For line-scan applications only, a start-of-scan and end-of-scan synchronization mechanism is implemented to ensure that all the cards of the DF-chain can capture the same lines of image data. Refer to *Line-scan Triggers Synchronization* on page 43 for more info.

Camera Cycle Control

If required by the application, the DF-master card is responsible for the elaboration of the CoaXPress Host-to-Device trigger. This is achieved in the same way as for non-data-forwarding Coaxlink cards.

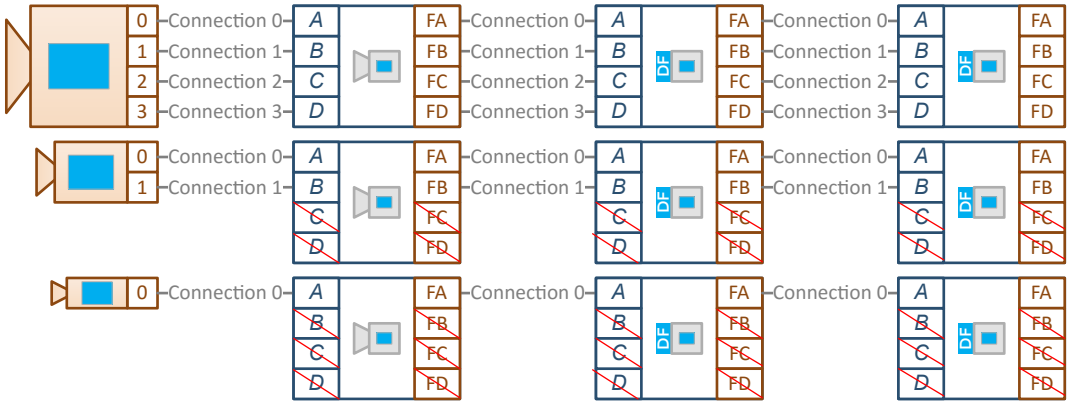
Refer to *CoaXPress Host To Device Trigger* on page 34 for more details.

Data Forwarding Connection Schemes

Area-scan Camera Data Forwarding

The following drawing illustrates 3 connection schemes where the image data of an area-scan camera is forwarded to 3 Host PCs: one for a 4-connection camera, one for a 2-connection camera, one for a single-connection camera.

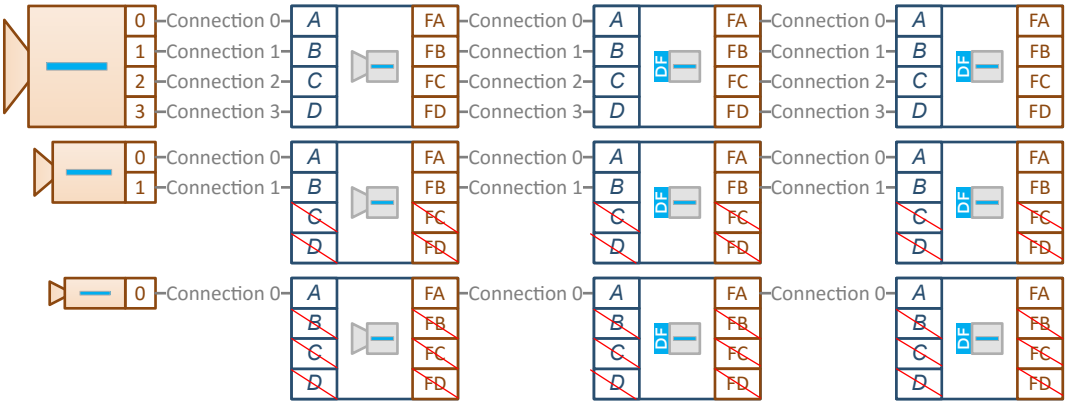
The first 1635 Coaxlink Quad G3 DF card must be configured with a `1-camera` firmware variant; the other must be configured with the `1-df-camera` firmware variant.



Line-scan Camera Data Forwarding

The following drawing illustrates 3 connection schemes where the image data of a line-scan is forwarded to 3 Host PCs: one for a 4-connection camera, one for a 2-connection camera, one for a single-connection camera.

The first 1635 Coaxlink Quad G3 DF card must be configured with a `1-camera, line-scan` firmware variant; the other must be configured with the `1-df-camera, line-scan` firmware variant.



Line-scan Triggers Synchronization

Start-of-scan and end-of-scan trigger synchronization mechanism to ensure that all the cards of the DF-chain capture the same lines of image data.

In a DF-chain, the DF-master card forwards its start-of-scan and end-of-scan events to the DF-slave cards.

The generation of the start-of-scan and end-of-scan events on the DF-master is achieved in the same way as for non-data-forwarding Coaxlink cards. Refer to *Line Scan Acquisition* on page 53 for more details.

The DF-master card:

- First, synchronizes the asynchronous scan triggers on the next start-of line image data.
- Then, share the synchronized scan triggers with all the DF-slaves.

The sharing of the scan triggers is achieved by the insertion of high-priority "custom GPIO messages" in the bit stream. These messages are forwarded by all the DF-slaves together with the image data.

On reception of such a message, the DF-slave generates a hardware event. Two kind of events are possible:

- Start-of-scan event.
- End-of-scan event

These events are available through the EIN tools of the I/O Toolbox.

For applications requiring synchronized line-scan acquisition, the I/O toolbox EIN tools of the DF-slaves must be used as local start-of-scan and end-of-scan trigger sources.

Configuration Script Example

The following script configures Data Forwarding frame grabbers for synchronized line-scan acquisition:

```
for (var grabber of grabbers) {
    if (grabber.InterfacePort.get("InterfaceID").includes('df-camera')) {
        console.log("Configuring slave card");
        // set the Width/Height/PixelFormat of the (virtual) remote device (on
        // the slave card) equal to the Width/Height/PixelFormat of the (real)
        // camera (connected to the master card)
        grabber.RemotePort.set("Width", 8192);
        grabber.RemotePort.set("Height", 1);
        grabber.RemotePort.set("PixelFormat", "Mono8");
        // configure the event input tool EIN1
        grabber.InterfacePort.set("EventInputToolSource[EIN1]", "A");
        grabber.InterfacePort.set("EventInputToolActivation[EIN1]", "StartOfScan");
        // configure the event input tool EIN2
        grabber.InterfacePort.set("EventInputToolSource[EIN2]", "A");
```

```
        grabber.InterfacePort.set("EventInputToolActivation[EIN2]", "EndOfScan");
        // configure start/end of scan triggers
        grabber.StreamPort.set("StartOfScanTriggerSource", "EIN1");
        grabber.StreamPort.set("EndOfScanTriggerSource", "ScanLength");
        grabber.StreamPort.set("ScanLength", 1000);
    } else {
        console.log("Configuring master card");
        grabber.RemotePort.set("TestPattern", "GreyDiagonalRampMoving");
        grabber.RemotePort.set("CxpLinkConfiguration", "CXP6_X4");
        grabber.RemotePort.set("CxpLinkConfigurationPreferredSwitch", "CXP6_X4");
        grabber.RemotePort.set("TriggerSource", "CXPin");
        grabber.RemotePort.set("TriggerMode", "On");
        grabber.DevicePort.set("CameraControlMethod", "RG");
        grabber.DevicePort.set("ExposureReadoutOverlap", "True");
        grabber.DevicePort.set("CxpTriggerAckTimeout", "0");
        grabber.DevicePort.set("StrobeDuration", "0");
        grabber.DevicePort.set("ExposureTime", "20");
        grabber.DevicePort.set("ExposureRecoveryTime", "0");
        grabber.DevicePort.set("CycleTargetPeriod", "50");
        // configure start/end of scan triggers
        grabber.StreamPort.set("StartOfScanTriggerSource", "Immediate");
        grabber.StreamPort.set("EndOfScanTriggerSource", "ScanLength");
        grabber.StreamPort.set("ScanLength", 1000);
    }
}
```

Note: *In this example, the start-of-scan trigger is the receipt of the start-of-scan event from the master, but the end-of-scan trigger is generated locally. One alternative would be to use `EIN2` as `EndOfScanTriggerSource`.*

Image Data Acquisition

Acquiring image data from CoaXPress data streams.

Acquisition Gating

The acquisition gate controls the data flow of a CoaXPress image data stream:

- When open, the image data are effectively acquired.
- When closed, the image data are discarded.

Area Scan Acquisition

The gate opens and closes at frame boundaries based on the application's calls of the `DSStartAcquisition` and `DSStopAcquisition` functions.

Each acquired frame is delivered into a single GenTL buffer.

Note: *The Camera and Illumination Controller indirectly controls the acquisition gating by issuing Camera Triggers using various schemes.*

Line Scan Acquisition

The gate opens and closes at line boundaries according to the application `DSStartAcquisition` and `DSStopAcquisition` function calls and, according to the settings of the Image Acquisition Controller, to the Start-of-scan and the End-of-scan triggers.

The data effectively acquired are structured as a **2-D images** composed of a fixed or a variable number of **lines**.

An image is delivered into one or several GenTL buffers.

Note: *The Camera and Illumination Controller controls the camera cycle timing, i.e. the line rate.*

FIFO Buffer

The Coaxlink cards embed a large memory that is entirely dedicated to the temporary storage of image pixel data. The total memory size is product-dependent.

Characteristic	Value	Applicable products
Total memory size	512 MB	Mono, Duo PCIe/104
	1 GB	Duo, Quad, Quad G3, Quad G3 DF

The local memory is partitioned such that each CoaXPress data stream has a dedicated portion of the memory. Each GenTL Data Stream owns one partition. The partition size is:

Characteristic	Value	Applicable products and firmware variants
Memory partition size	256 MB	Quad G3, Duo PCIe/104

Characteristic	Value	Applicable products and firmware variants
	512 MB	Mono, Duo, Quad, Quad G3, Quad G3
	1 GB	Duo, Quad, Quad, Quad G3, Quad G3, Quad G3 DF, Quad G3 DF, Quad G3 DF, Quad G3 DF

Each memory partition operates as a FIFO buffer:

- All the captured data coming from the device data stream are first stored in the FIFO.
- The data are extracted as soon as possible from the FIFO then possibly processed and finally transmitted over the PCI Express to the corresponding buffer in the Host PC memory.

Having a FIFO buffer decouples the PCI Express bandwidth requirements from the CoaXPress bandwidth:

- Temporary non-availabilities of the PCI Express are absorbed by the FIFO without having any impact on the CoaXPress link .
- Enables image burst acquisition at very high (max) CoaXPress data rates significantly exceeding the capabilities of the PCI Express interface.

Image Data Stream Processing

Every image data stream channel of the Coaxlink card is fitted with an *image data stream pixel processor*.

The image data stream pixel processor:

- Converts to little-endian the 2-byte pixel component data
- Unpacks the 10-bit, 12-bit, and 14-bit pixel component data into 16 bits using the lsb or the msb alignment
- Optionally, swaps the Red and the Blue component of an RGB packed pixel data

The above operations are executed while transferring image data from the on-board buffer to the Host-PC memory without adding any significant latency.

Endianness Conversion

Considering that:

- As specified by the CoaXPress standard, CoaXPress devices deliver image pixel data using the big-endian convention,
- The x86 processors are using the little-endian convention,

the Coaxlink image data stream pixel processor performs the endianness conversion on each individual pixel component data.

Note: *The endianness conversion is efffective only when delivering image pixel data in 16-bit containers.*

Little-endian Convention

The least-significant byte of a multiple byte data is stored at the lowest address location.

For instance, 16-bit data are stored into two consecutive byte locations as follows:

Memory Byte Location	Memory Byte Content
N	Data[7:0]
N+1	Data[15:8]

Pixel Component Unpacking

Considering that:

- As specified by the CoaXPress standard, CoaXPress devices deliver packed pixel data,
- Machine vision applications prefer 16-bit aligned pixel component data,

the Coaxlink image data stream pixel processor performs the unpacking to 16-bit of 10-bit, 12-bit, and 14-bit pixel component data.

For Coaxlink Driver versions prior to 4.2, the unpacking operation was not configurable: 10-bit, 12-bit, and 14-bit pixel component data were unpacked to 16-bit using the alignment to msb method.

Since version 4.2, the unpacking operation is user-configurable through the `UnpackingMode`GenICam feature. Two options are available:

- Unpacking to lsb.
- Unpacking to msb.

Note: *Since Coaxlink Driver 4.3, the default option is "Unpacking to lsb".*

Unpacking to lsb

The significant bits of the pixel component data are aligned to the **least significant bit** of the data container. Padding '0' bits are put as necessary in the **most significant bits** to reach the next 8-bit boundary.

Bit assignment of 2-byte pixel components after unpacking to lsb

Input bit depth	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10-bit	0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
12-bit	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
14-bit	0	0	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Note: *Unpacking to lsb doesn't modify the pixel component value.*

Unpacking to msb

The significant bits of the pixel component data are aligned to the **most significant bit** of the data container. Padding '0' bits are put as necessary in the **least significant bits** to reach the next 8-bit boundary.

Bit assignment of 2-byte pixel components after unpacking to msb

Input bit depth	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10-bit	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
12-bit	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

Input bit depth	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
14-bit	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0

Note: *Unpacking to msb 10-bit, 12-bit, and 14-bit pixel components multiplies the pixel component value by 64, 16, and 4 respectively.*

Note: *Unpacking 8-bit and 16-bit pixel components is a neutral operation:*

- *The size of the data container is unchanged: One byte for 8-bit pixel components; two bytes for 16-bit pixel components*
- *The data bits are not modified*

Note: *Unpacking 10-bit, 12-bit, and 14-bit pixel components increases the amount of data by 160%, 133%, and 114% respectively.*

Pixel Component Order

Considering that:

- As specified by the CoaXPress standard, CoaXPress devices deliver RGB pixel data using the R, G, B pixel order
- Machine vision applications prefer the B,G, R pixel component order,

since version 4.2 of the Coaxlink Driver, the Coaxlink image data stream pixel processor can be configured to swap the R and B pixel component data of RGB pixels.

The Red/Blue component swapping can be controlled through the `RedBlueSwap` boolean GenICam feature. The component swapping exclusively applies to the 3-component RGB pixel formats. It doesn't apply to other formats, e.e. RGBa.

The default settings preserve the original component order.

Pixel Component Order

Output Pixel Format [PFNC]	Component Order
RGB8 | RGB10 | RGB12| RGB14| RGB16	R (first), G, B (last)
BGR8 | BGR10 | BGR12| BGR14| BGR16	B (first), G, R (last)
RGBa8 | RGBa10 | RGBa12 | RGBa14 | RGBa16	R (first), G, B, a (last)

Note: *The order of component of the RGB8 format is not the same as the MultiCam RGB24 format. In RGB24, the component order is BGR with B at byte 0.*

Pixel Order

The Coaxlink image data stream pixel processor preserves the pixel order of the CoaXPress data stream:

- The pixels data of an image frame are stored in successive address locations starting with the first pixel of the first line at the lowest address.
- The successive lines of an image frame are concatenated in the image buffer.

Processor Configurations

Processor Configurations for 1-component Pixel Formats

Input Pixel Format [PFNC]	UnpackingMode	RedBlueSwap	Output Pixel Format[PFNC]
Mono8 | Bayer**8	lsb | msb	off | on	Mono8 | Bayer**8
Mono10pmsb | Bayer**10pmsb	lsb	off | on	Mono10 | Bayer**10
	msb	off | on	Mono16 | Bayer**16
Mono12pmsb | Bayer**12pmsb	lsb	off | on	Mono12 | Bayer**12
	msb	off | on	Mono16 | Bayer**16
Mono14pmsb | Bayer**14pmsb	lsb	off | on	Mono14 | Bayer**14
	msb	off | on	Mono16 | Bayer**16
Mono16pmsb | Bayer**16pmsb	lsb | msb	off | on	Mono16 | Bayer**16

Pixel Component Data processor Configurations for 3-component Pixel Formats

Input Pixel Format [PFNC]	UnpackingMode	RedBlueSwap	Output Pixel Format[PFNC]
RGB8	lsb | msb	off	RGB8
		on	BGR8
RGB10pmsb	lsb	off	RGB10
		on	BGR10
	msb	off	RGB16
		on	BGR16
RGB12pmsb	lsb	off	RGB12
		on	BGR12
	msb	off	RGB16
		on	BGR16
RGB14pmsb	lsb	off	RGB14
		on	BGR14
	msb	off	RGB16
		on	BGR16
RGB16pmsb	lsb | msb	off	RGB16

Input Pixel Format [PFNC]	UnpackingMode	RedBlueSwap	Output Pixel Format[PFNC]
		on	BGR16

Pixel Component Data processor Configurations for 4-component Pixel Formats

Input Pixel Format [PFNC]	UnpackingMode	RedBlueSwap	Output Pixel Format[PFNC]
RGBa8	lsb | msb	off | on	RGBa8
RGBa10pmsb	lsb	off | on	RGBa10
	msb	off | on	RGBa16
RGBa12pmsb	lsb	off | on	RGBa12
	msb	off | on	RGBa16
RGBa14pmsb	lsb	off | on	RGBa14
	msb	off | on	RGBa16
RGBa16	lsb | msb	off | on	RGBa16

Pixel Processing GenICam Features

UnpackingMode

Data Stream	Root > Pixel processing	Enumeration (RW)

On read: reports the selected unpacking mode of multi-byte pixel components.

On write: selects the unpacking mode of multi-byte pixel components.

Possible values:

- `Lsb`: **Unpacking to lsb.** Each pixel component is unpacked to the **least** significant bit. Padding '0' bits are put as necessary in the most significant bits to reach the next 8-bit boundary. 16-bit data are delivered using the little-endian convention. **Default value**.
- `Msb`: **Unpacking to msb.** Each pixel component is unpacked to the **most** significant bit. Padding '0' bits are put as necessary in the least significant bits to reach the next 8-bit boundary. 16-bit data are delivered using the little-endian convention. *Default value for Coaxlink Driver prior to 4.3.*

RedBlueSwap

Data Stream	Root > Pixel processing	Boolean (RW)

On read: reports the selected Red-Blue pixel component swap option of RGB pixels.

On write: selects the Red-Blue pixel component swap option of RGB pixels.

Possible values:

- `On`: **Swap R and B.** The first (Red) and the last (Blue) color components of an RGB packed pixel are swapped before being delivered.
- `Off`: **No swap.** The pixel component order remains unchanged. **Default value**.

Stream Statistics GenICam Features

StatisticsSamplingSelector

Data Stream	Root > Stream Statistics	Enumeration (RW)

Selects the stream statistics sampling method.

Possible values:

- `LastSecond`: During the last second. **Default value.**
- `LastTenSeconds`: During the last 10 seconds.
- `Last2Buffers`: For the last 2 buffers.
- `Last10Buffers`: For the last 10 buffers.
- `Last100Buffers`: For the last 100 buffers.
- `Last1000Buffers`: For the last 1000 buffers.
- `LastAcquisition`: During the last acquisition activity period. Namely since the last DSStartAcquisition() function call until now, if the acquisition is still active otherwise until the last DSStopAcquisition() function call.
- `Custom`: Custom sampling using `StatisticsStartSampling` and `StatisticsStopSampling` commands

StatisticsFrameRate

Data Stream	Root > Stream Statistics	Float (RW)

Average frame delivery rate using the selected sampling method.

Note: *This feature is only available with the following firmware variants:*

Note: *The statistics measures the frame rate at the level of the PCI Express interface, NOT at the level of the CoaXPress interface!*

Unit: Frames per second.

StatisticsLineRate

Data Stream	Root > Stream Statistics	Integer (RW)

Average line acquisition rate using the selected sampling method.

Note: *This feature is only available with the following firmware variants:*

Note: *The statistics measures the line rate at the level of the PCI Express interface, NOT at the level of the CoaXPress interface!*

Unit: Lines per second.

StatisticsDataRate

Data Stream	Root > Stream Statistics	Float (RW)

StatisticsDataRate

Average acquisition data delivery rate using the selected sampling method.

Note: *The statistics measures the data rate at the level of the PCI Express interface, NOT at the level of the CoaXPress interface!*

Unit: Megabytes per second. *(1,000,000 bytes per second)*

StatisticsStartSampling

Data Stream	Root > Stream Statistics	Command (WO)

Starts sampling the stream data.

Note: *Applies only when* `StatisticsSamplingSelector` = `Custom`.

StatisticsStopSampling

Data Stream	Root > Stream Statistics	Command (WO)

Stops sampling the stream data.

Note: *Applies only when* `StatisticsSamplingSelector` = `Custom`.

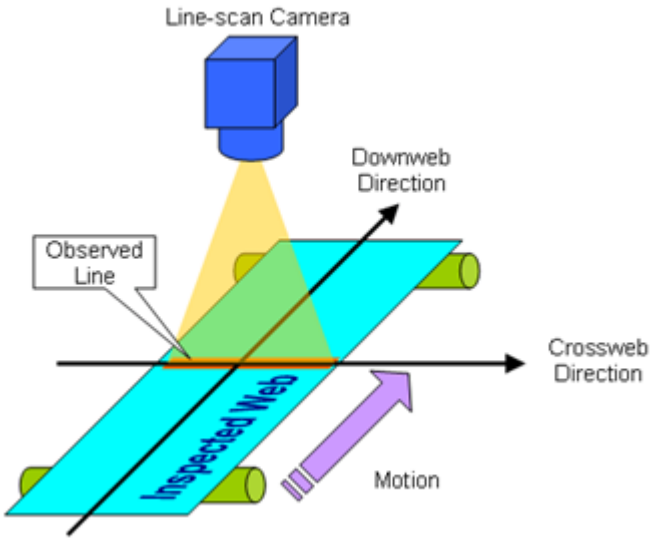
Line Scan Acquisition

Acquiring image data from line-scan cameras.

This section applies only to the line-scan firmware variants:

Line Scan Acquisition Principles

Line Scan Imaging

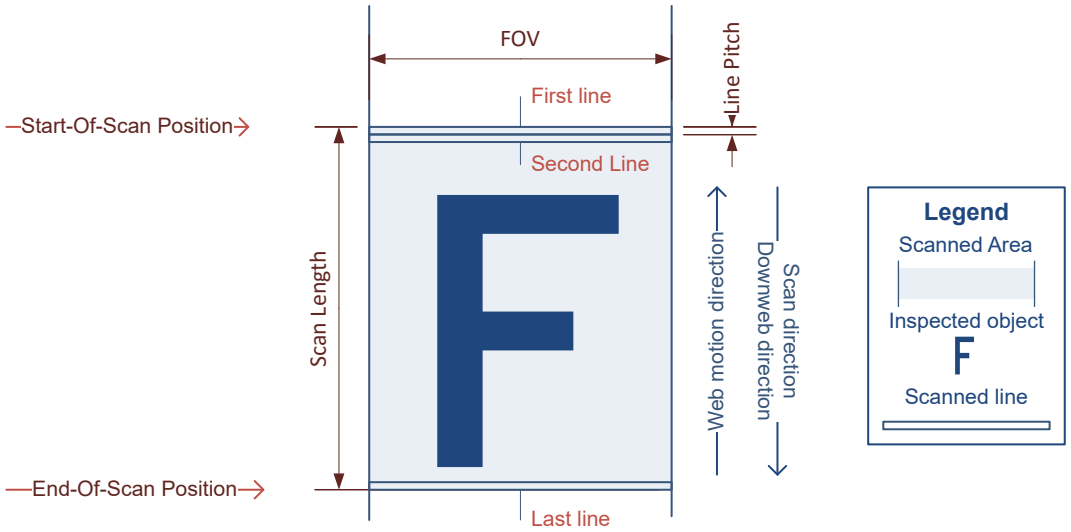


The expression “Line-scan imaging” designates machine vision applications where 2-D images are obtained by the combination of successive image lines captured from a 1-D imaging device that moves relatively to the object.

In line-scan imaging:

- The imaging device is often, but not necessarily, a line-scan camera.
- The inspected object is often a continuous web, it can also be discrete objects having fixed or variable size.
- The inspected web moves relatively to the camera. The motion speed during the acquisition can be fixed or variable.
- The **cross-web direction** or **transverse direction** is the axis on the web plane that is observed by the camera.
- The **down-web direction** or **axial direction** is the motion direction of the inspected web relatively to the camera.

Scanning Area



The scanning area is a 2-D area on the web having a width equal to FOV and a length equal to Scan Length.

In the cross-web direction (horizontal direction in the above drawing), the scanning area is delimited by the field of view – FOV – of the camera.

In the down-web direction (vertical direction in the above drawing), the scanning area is delimited by the start-of-scan and the end-of-scan positions. The line pitch is determined by the ratio between the web speed and the camera line rate.

The **field of view – FOV –** of a line-scan camera is determined only by the optical setup and the sensor geometrical properties.

The **start-of-scan position** is a position on the web corresponding to the scan-line boundary preceding the first acquired line.

The **end-of-scan position** is a position on the web corresponding to the scan-line boundary following the last acquired line.

Most of the line-scan cameras are delivering a single row of pixels every camera cycle. Consequently, **multiple camera cycles** are necessary to build-up the object image.

Pixel Aspect Ratio Control

Unlike area-scan imaging, line-scan imaging allows the application to control the image pixel aspect ratio.

In the large majority of cases, the imaging application requires a constant, and preferably a 1:1 image pixel aspect ratio.

The cross-web pitch being locked by the sensor pitch and the optical magnification factor, the image pixel aspect ratio is controllable only through the line pitch control.

The following table summarizes the methods providing a constant line pitch that are applicable with Coaxlink:

Method Name	Description
VCR	**Variable Camera cycle.** The **web speed is variable** and the **camera cycle rate is kept proportional to the web speed**. The frame grabber builds the object image by capturing all the successive lines delivered by the camera.
CCC	**Constant Camera Cycle.** The camera operates at a constant cycle rate and the frame grabber captures all the successive lines delivered by the camera.

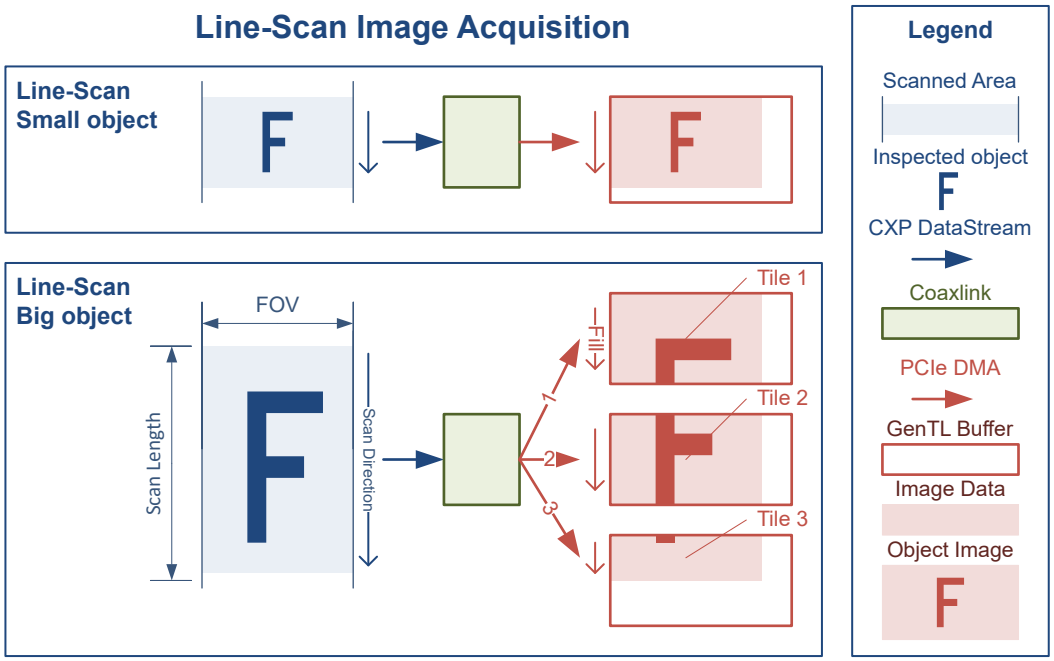
The VCR method requires:

- A **motion encoder** for measuring the web speed.
- A **real-time processing** of the motion encoder events to build a camera trigger at a rate that is **proportional** to the motion encoder events rate.

Having a proportional rate can be achieved by a **divider tool** or a **multiplier/divider tool**:

- The divider tool decimates the input rate by an integer value, it delivers 1 out of N incoming events.
- The multiplier/divider tool enables fine control of the image pixel aspect ratio by allowing any rate conversion ratio value – RCR – in the range 0.001 to 1000 with an accuracy better than 0.1% of the RCR value.

Image Acquisition with Line Scan Imaging Devices



For the transmission on the CoaXPress link, (most of) the line-scan cameras use **one CoaXPress Image Data Stream**.

Regarding the delivery methods of the image data, two cases are to be considered:

- For small objects, the object image data are delivered into a **single GenTL buffer**.
- For big objects, the object image data are delivered into **multiple GenTL buffers**.

In both cases, the image data are delivered through a single PCIe DMA channel and the transmission latency through Coaxlink is low: “one image line”.

GenTL Buffer Filling Rules

In line-scan imaging, GenTL buffers are filled according to the following rules:

- The first acquired line data of a scan is always stored at the beginning of a new buffer.
- A buffer contains an integer number of image lines data.
- When the remaining space of a buffer is not sufficient to store an image line data, the acquisition continues into a new buffer and the filled buffer is made available to the application for processing.
- When the last line data of a scan is acquired, the last buffer, possibly partially filled, is made available to the application for processing.

Line Scan Acquisition Control GenICam Features

StartOfScanTriggerSource		
Data Stream	Root > Line-scan Acquisition Control	Enumeration (RW)

StartOfScanTriggerSource

Start-of-scan trigger conditions and trigger source.

Possible values:

- `Immediate`: Immediately after `DSStartAcquisition()` is called. **Default value.**
- `StartScan`: On execution of the `StartScan` command.
- `LIN1`: When an event occurs on I/O toolbox LIN1 or on execution of the `StartScan` command.
- `LIN2`: When an event occurs on I/O toolbox LIN2 or on execution of the `StartScan` command.
- `LIN3`: When an event occurs on I/O toolbox LIN3 or on execution of the `StartScan` command.
- `LIN4`: When an event occurs on I/O toolbox LIN4 or on execution of the `StartScan` command.
- `LIN5`: When an event occurs on I/O toolbox LIN5 or on execution of the `StartScan` command.
- `LIN6`: When an event occurs on I/O toolbox LIN6 or on execution of the `StartScan` command.
- `LIN7`: When an event occurs on I/O toolbox LIN7 or on execution of the `StartScan` command.
- `LIN8`: When an event occurs on I/O toolbox LIN8 or on execution of the `StartScan` command.
- `QDC1`: When an event occurs on I/O toolbox QDC1 or on execution of the `StartScan` command.
- `QDC2`: When an event occurs on I/O toolbox QDC2 or on execution of the `StartScan` command.
- `DIV1`: When an event occurs on I/O toolbox DIV1 or on execution of the `StartScan` command.
- `DIV2`: When an event occurs on I/O toolbox DIV2 or on execution of the `StartScan` command.
- `MDV1`: When an event occurs on I/O toolbox MDV1 or on execution of the `StartScan` command.
- `MDV2`: When an event occurs on I/O toolbox MDV2 or on execution of the `StartScan` command.
- `DEL11`: When an event occurs on Output 1 of I/O toolbox DEL1 or on execution of the `StartScan` command.
- `DEL12`: When an event occurs on Output 2 of I/O toolbox DEL1 or on execution of the `StartScan` command.
- `DEL21`: When an event occurs on Output 1 of I/O toolbox DEL2 or on execution of the `StartScan` command.
- `DEL22`: When an event occurs on Output 2 of I/O toolbox DEL2 or on execution of the `StartScan` command.
- `EIN1`: When an event occurs on I/O toolbox EIN1 or on execution of the `StartScan` command.
- `EIN2`: When an event occurs on I/O toolbox EIN2 or on execution of the `StartScan` command.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

EndOfScanTriggerSource

Data Stream	Root > Line-scan Acquisition Control	Enumeration (RW)

EndOfScanTriggerSource

End-of-scan trigger conditions and trigger source.

Possible values:

- `ScanLength`: After `ScanLength` lines are captured. **Default value.**
- `StopScan`: On execution of the `StopScan` command.
- `LIN1`: When an event occurs on I/O toolbox LIN1 or on execution of the `StopScan` command.
- `LIN2`: When an event occurs on I/O toolbox LIN2 or on execution of the `StopScan` command.
- `LIN3`: When an event occurs on I/O toolbox LIN3 or on execution of the `StopScan` command.
- `LIN4`: When an event occurs on I/O toolbox LIN4 or on execution of the `StopScan` command.
- `LIN5`: When an event occurs on I/O toolbox LIN5 or on execution of the `StopScan` command.
- `LIN6`: When an event occurs on I/O toolbox LIN6 or on execution of the `StopScan` command.
- `LIN7`: When an event occurs on I/O toolbox LIN7 or on execution of the `StopScan` command.
- `LIN8`: When an event occurs on I/O toolbox LIN8 or on execution of the `StopScan` command.
- `QDC1`: When an event occurs on I/O toolbox QDC1 or on execution of the `StopScan` command.
- `QDC2`: When an event occurs on I/O toolbox QDC2 or on execution of the `StopScan` command.
- `DIV1`: When an event occurs on I/O toolbox DIV1 or on execution of the `StopScan` command.
- `DIV2`: When an event occurs on I/O toolbox DIV2 or on execution of the `StopScan` command.
- `MDV1`: When an event occurs on I/O toolbox MDV1 or on execution of the `StopScan` command.
- `MDV2`: When an event occurs on I/O toolbox MDV2 or on execution of the `StopScan` command.
- `DEL11`: When an event occurs on Output 1 of I/O toolbox DEL1 or on execution of the `StopScan` command.
- `DEL12`: When an event occurs on Output 2 of I/O toolbox DEL1 or on execution of the `StopScan` command.
- `DEL21`: When an event occurs on Output 1 of I/O toolbox DEL2 or on execution of the `StopScan` command.
- `DEL22`: When an event occurs on Output 2 of I/O toolbox DEL2 or on execution of the `StopScan` command.
- `EIN1`: When an event occurs on I/O toolbox EIN1 or on execution of the `StopScan` command.
- `EIN2`: When an event occurs on I/O toolbox EIN2 or on execution of the `StopScan` command.

ScanLength

Data Stream	Root > Line-scan Acquisition Control	Integer (RW)

Sets/gets the number of captured lines before stopping the scanning.

Note: *Applies only when `EndOfScanTriggerSource` = `ScanLength`.*

Unit: lines.

Value range: from 1 up to 16,777,215.

Default value: 512.

StartScan

Data Stream	Root > Line-scan Acquisition Control	Command (WO)

Starts a scan.

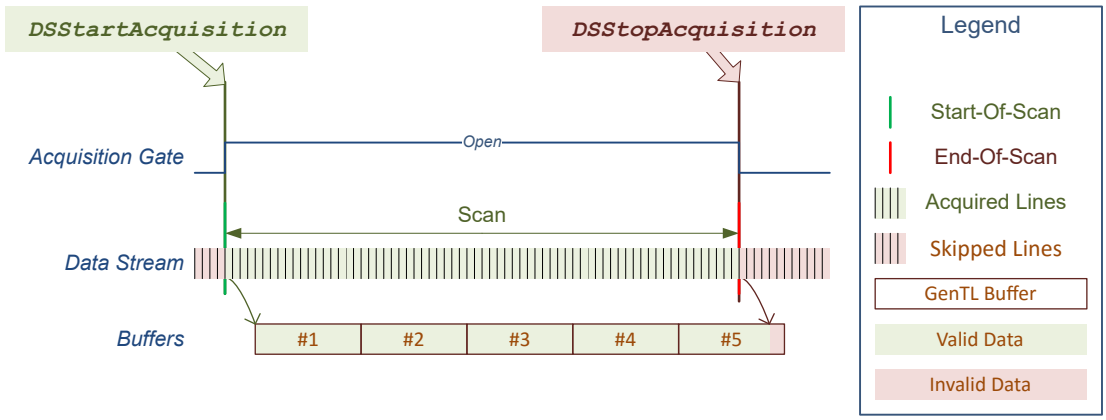
StopScan

Data Stream	Root > Line-scan Acquisition Control	Command (WO)

StopScan
Stops a scan.

Line Scan Acquisition Use cases

Scanning of continuous objects



This case applies to the image scanning of **continuous objects**.

The Coaxlink acquisition controller is configured as follows:

- `StartOfScanTriggerSource` = `Immediate`.
- `EndOfScanTriggerSource` = `StopScan`.

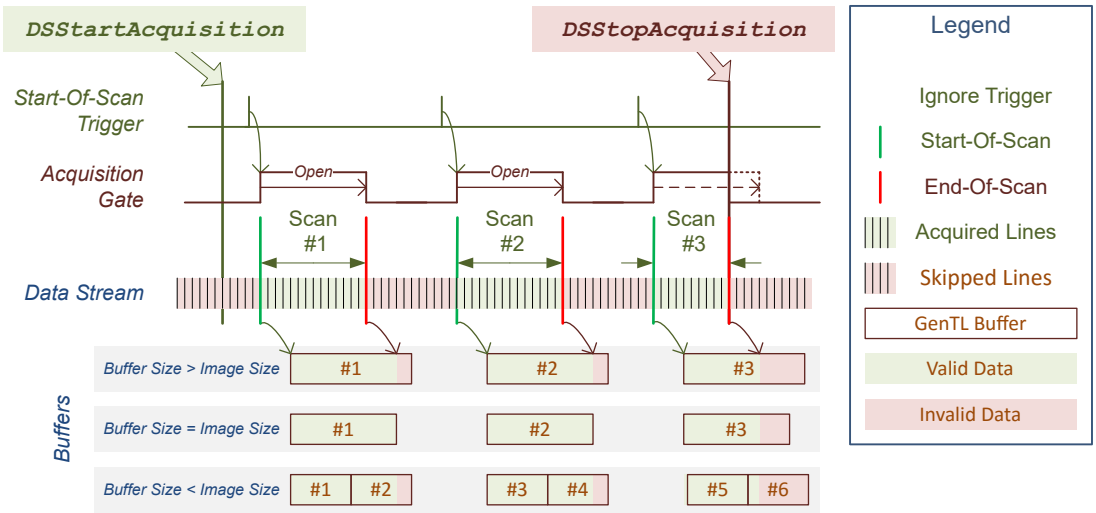
When the `DSStartAcquisition` function is called, the scanning starts at the next line boundary.

The acquisition gate closes when the application calls the `DSStopAcquisition` function.

Depending on the allocated buffer size and the scanning duration, the object image fits in a single buffer or requires multiple buffers.

Each buffer is delivered to the application as soon as it is filled. The last buffer, likely partially filled, is delivered as soon as the last image data are written.

Fixed-length scanning of discrete objects



This case applies to the scanning of **discrete objects** with a common **scan length**.

The Coaxlink acquisition controller is configured as follows:

- `StartOfScanTriggerSource` = `StartScan` or any applicable I/O Toolbox event output, for instance: `LIN1`.
- `EndOfScanTriggerSource` = `ScanLength`.
- `ScanLength` = any positive number representing the number of lines required to capture the object image entirely.

When the `DSStartAcquisition` function is called, the start-of-scan trigger of the acquisition controller is armed.

Then, the acquisition controller waits for the first occurrence of a valid start-of-scan trigger event.

A valid start-of-scan event can be generated:

- By the application using a `StartScan` command.
- By the selected hardware event source, if specified by `StartOfScanTriggerSource`.

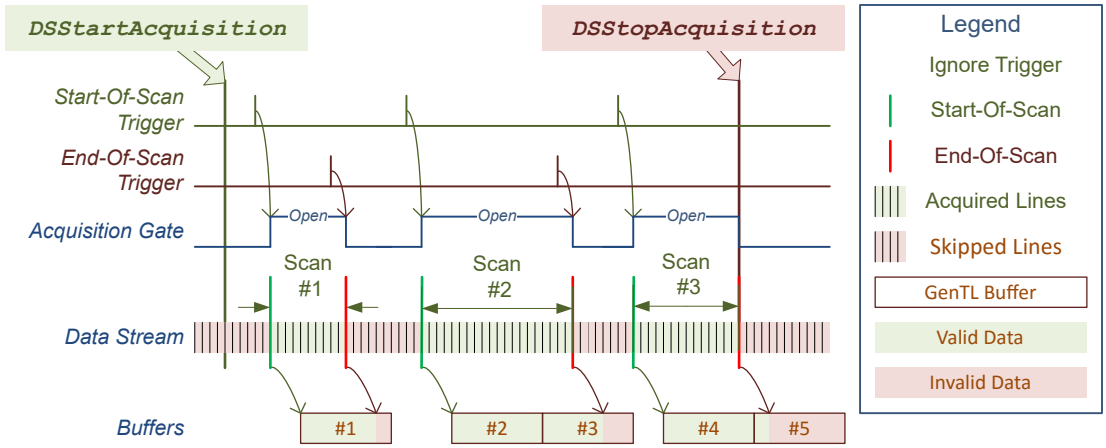
The acquisition controller ignores any Start-of-Scan trigger event while a scanning is in progress.

The acquisition gate opens at the first line boundary following a start-of-scan event.

The acquisition gate closes automatically after the specified number of lines have been acquired or anticipatively when the application calls the `DSStopAcquisition` function.

Depending on the allocated buffer size, the object image fits in a single buffer or requires multiple buffers. Each buffer is delivered to the application as soon as it is filled. At the end-of-scan, partially filled buffers are immediately delivered. The following image acquisition always begins with a new buffer.

Variable-length scanning of discrete objects



This case applies to the image scanning of **discrete objects** requiring a variable **scan length**.

The Coaxlink acquisition controller is configured as follows:

- `StartOfScanTriggerSource` = `StartScan` or any applicable I/O Toolbox event output, for instance: `LIN1`.
- `EndOfScanTriggerSource` = `StopScan` or any applicable I/O Toolbox event output, for instance: `LIN2`.

When the `DSStartAcquisition` function is called, the start-of-scan trigger of the acquisition controller is armed.

Then, the acquisition controller waits for the first occurrence of a valid start-of-scan trigger event.

A valid start-of-scan event can be generated:

- By the application using a `StartScan` command.
- By the selected hardware event source, if specified by `StartOfScanTriggerSource`.

The acquisition controller ignores any Start-of-Scan trigger event while a scanning is in progress.

The acquisition gate opens at the first line boundary following a start-of-scan event.

The acquisition gate closes at the first line boundary following a valid end-of-scan event or immediately when the application calls the `DSStopAcquisition` function.

A valid end-of-scan event can be generated:

- By the application using a `StopScan` command.
- By the selected hardware event source, if specified by `EndOfScanTriggerSource`.

The acquisition controller ignores any End-of-Scan trigger event when no scanning is in progress.

Depending on the allocated buffer size, the object image fits in a single buffer or requires multiple buffers. Each buffer is delivered to the application as soon as it is filled. At the end-of-scan, partially filled buffers are immediately delivered. The next image acquisition always begins with a new buffer.

Camera Control Principles

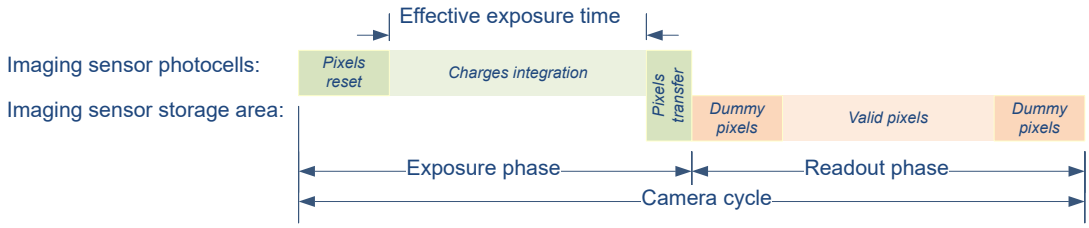
General principles used by Coaxlink for controlling a camera.

This section contains:

- *Camera Cycle* on page 61
- *Camera Cycle Concatenation Rules* on page 62
- *Camera Control Methods* on page 63
- *Camera Trigger Signal Paths* on page 73

Camera Cycle

A camera cycle is composed of two consecutive phases: the exposure phase and the readout phase.



Exposure phase

The exposure phase is the period of time during which the photocells of the imaging sensor integrate electric charges induced by the incoming photons.

For cameras having an electronic shutter, the exposure phase begins with a pixel reset action that clears all the sensor photocells. For permanent exposure cameras, i.e. cameras having no (or not using) the electronic shutter, the exposure phase begins immediately after the completion of the previous exposure phase.

For all types of cameras, the exposure phase terminates with a “pixel transfer” action. The accumulated charges in the photocell are transferred to the storage area for further readout. This action clears the photocells and new charge integration begins immediately.

Cameras having an electronic shutter have the capability to reset the pixels asynchronously and initiate a new exposure on request. These cameras are named asynchronous reset cameras.

Having the capability of controlling the time of the start of exposure (pixel reset) and the time of the end of exposure (pixel transfer) gives full control on:

- The timing of each image capture
- The sensitivity of the imaging sensor by selecting the exposure time

Readout phase

The readout phase is the period of time during which the total amount of electrical charges accumulated by each pixel is measured and delivered to the imaging sensor output.

The readout phase is not controlled by the frame grabber:

- It is automatically initiated after each pixel transfer.
- Its duration is fixed; it is determined by the amount of pixel data to be transferred and by the readout structure of the sensor (one or more taps, tap output data rate).

Some sensors provide the capability to select one or more region of interest (ROI) speeding up the readout since less data needs to be transferred.

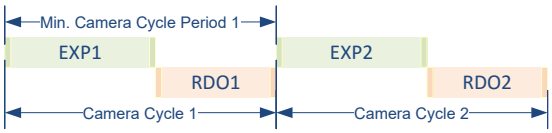
Camera Cycle Concatenation Rules

This topic explains the rules that MUST be observed by the frame grabber to avoid camera trigger overrun when requesting successive camera cycles to an asynchronous reset camera.

Rule for cameras not allowing overlapping

The next camera cycle may NOT begin before the completion of the readout phase.

The following diagram illustrates the shortest possible cycle period achievable by cameras NOT allowing the cycle overlapping:



The minimum cycle period is: $Min\ Cycle\ Period_n = EXP_n + RDO_n$

Note: *Only a minority of industrial cameras are NOT allowing the cycle overlapping!*

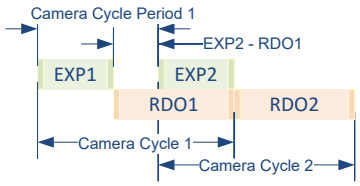
Rules for cameras allowing overlapping

The exposure phases of two consecutive camera cycles may NEVER overlap.

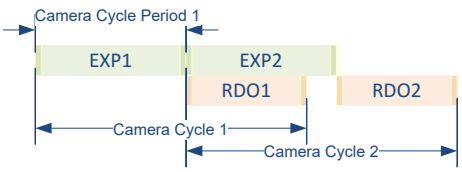
The readout phases of two consecutive camera cycles may NEVER overlap

The following diagrams illustrate the shortest possible cycle period achievable by cameras allowing cycle overlapping:

Use case 1: EXP2 < RDO1



Use case 2: EXP2 > RDO1



In the first case:

- The duration of the exposure phase of the 2nd cycle is shorter than the duration of the readout phase of the first cycle.
- The next camera cycle may start (EXP_{n+1} - RDO_n) period of time after the completion of the exposure phase.
- The minimum cycle period is $Min\ Cycle\ Period_n = EXP_n + RDO_n - EXP_{n+1}$

In the second case:

- The duration of the exposure phase of the 2nd cycle is longer than the duration of the readout phase of the first cycle.
- The next camera cycle may start immediately after the completion of the exposure phase.
- The minimum cycle period is: $Min\ Cycle\ Period_n = EXP_n$

Note: *The majority of asynchronous reset cameras used in the machine vision industry supports the overlapping of the camera cycles!*

Camera Control Methods

The Coaxlink frame grabber uses three fundamental camera control methods designated by the acronyms RG, RC, and NC:

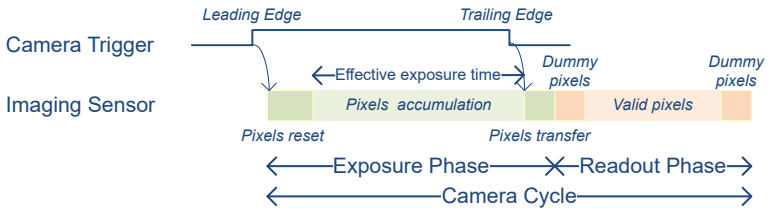
- The first letter R designates *asynchronous Reset camera control*
- The second letter C designates *Camera-controlled exposure*
- The second letter G designates *Grabber-controlled exposure*

> ***Note:*** *RC and RG are acronyms inherited from MultiCam.*

RG camera control method

The RG camera control method targets asynchronous reset cameras where both the camera cycle rate and the exposure duration are controlled by the frame grabber.

The real-time control is performed through a single upstream signal named "Camera Trigger" issued by the Camera and Illumination Controller (CIC) of Coaxlink.



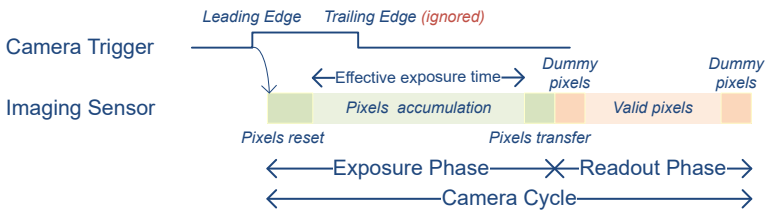
The CIC produces one single Camera Trigger pulse every camera cycle:

- The Camera Trigger leading edge triggers a new camera cycle and initiates a new exposure period
- The Camera Trigger falling edge terminates the exposure period and triggers the readout

RC camera control method

The RC camera control methods targets asynchronous reset cameras where only the camera cycle rate is controlled by the frame grabber. The exposure duration is controlled by the camera.

The real-time control is performed through a single upstream signal named "Camera Trigger" issued by the Camera and Illumination Controller (CIC) of Coaxlink.



The CIC produces one single Camera Trigger pulse every camera cycle:

- The Camera Trigger leading edge triggers a new camera cycle and initiates a new exposure period
- The Camera Trigger falling edge is ignored by the camera

NC camera control method

The NC camera control methods targets cameras that are NOT controlled by the frame grabber. This includes

- Free-run cameras not using any external trigger signal
- Asynchronous-reset cameras using an external trigger signal not delivered by the frame grabber.

There are no real-time control of the camera by the frame grabber. The Camera and Illumination Controller (CIC) of Coaxlink is not used.

Illumination Control Principles

General principles used by Coaxlink for controlling an illumination device.

This section contains:

- *Illumination Devices* on page 64
- *Aligning Camera and illumination Cycles* on page 64
- *Strobe Signal Paths* on page 74

Illumination Devices

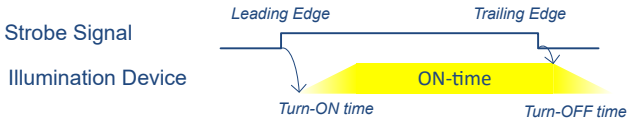
Two classes of illumination devices can be controlled by the illumination controller:

- Intermittent illumination devices
- Strobed illumination devices

Intermittent Illumination Devices

This illumination device class includes switched light sources where the turn-on and the turn-off time are controlled by the leading and the falling edges of the strobe signal.

The width of the strobe pulse determines the ON time duration of the light source.

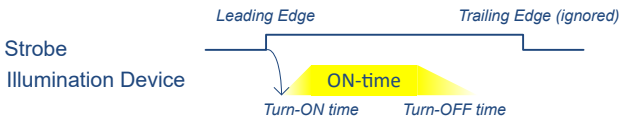


Note: *The turn-on time and the turn-off time need to be considered when configuring the illumination controller!*

Strobed Illumination Devices

This illumination device class includes switched light sources where only the turn-on time is controlled by the leading edge of the strobe signal.

The on-time duration is either uncontrolled or controlled by the illumination device itself.



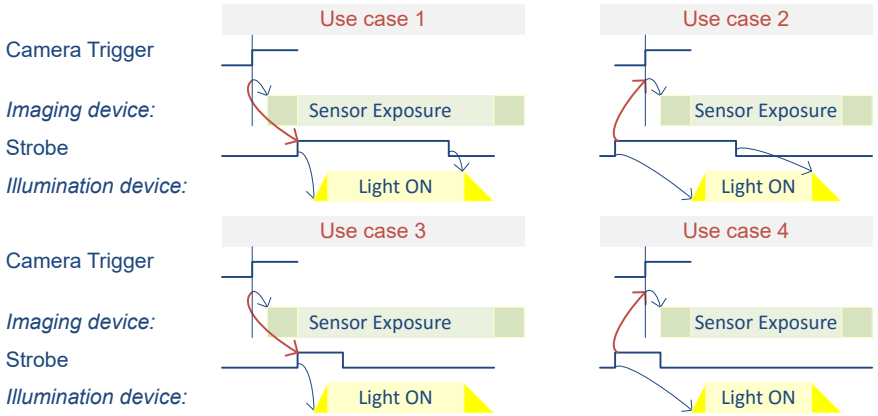
Note: *The turn-on time and the ON time duration need to be considered when configuring the illumination controller.*

Aligning Camera and illumination Cycles

Obviously, the ON time of the light source must coincide with the exposure phase of the imaging sensor.

Therefore, the time relationship between the strobe signal(s) and the camera trigger signal must be adequately controlled.

The following drawing shows 4 typical use cases:



Intermittent Light Sources (Use cases 1 & 2)

The duration of the strobe pulse must be adequately controlled in order to provide the right amount of light and get a correctly exposed image.

The sensor exposure should be adequately timed in order to terminate the sensor exposure after the light has turned off.

Strobed Light Sources (Use cases 3 & 4)

The sensor exposure should be adequately timed in order to terminate the sensor exposure after the light has turned off.

Late Strobe (Use cases 1 & 3)

The leading edge (beginning) of the Strobe signal is delayed a little to ensure that the light is not turned on too early while the imaging device is resetting its pixels.

Early Strobe (Use cases 2 & 4)

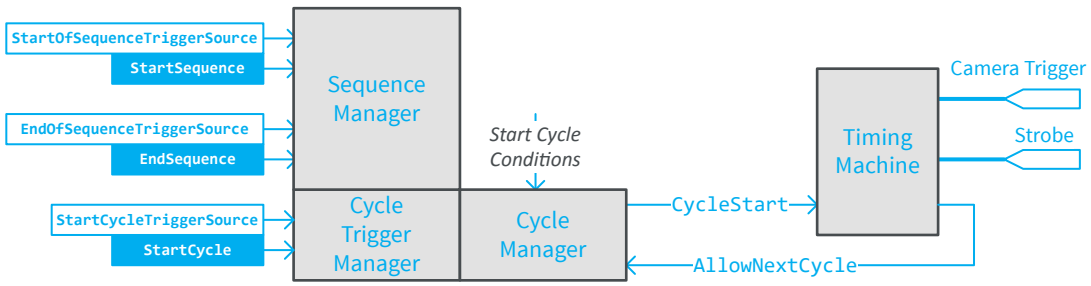
The leading edge of the Camera Trigger signal is delayed a little to ensure that the sensor exposure time is kept as short as possible and closely matches the on time.

Camera and Illumination Controller

Full description of the Coaxlink Camera and Illumination Controller.

Description

Organic and functional description of the Camera and Illumination Controller.



The Coaxlink Camera and Illumination Controller (abbreviated as CIC) provides the real-time control of one imaging device (the camera) and its associated illumination device(s).

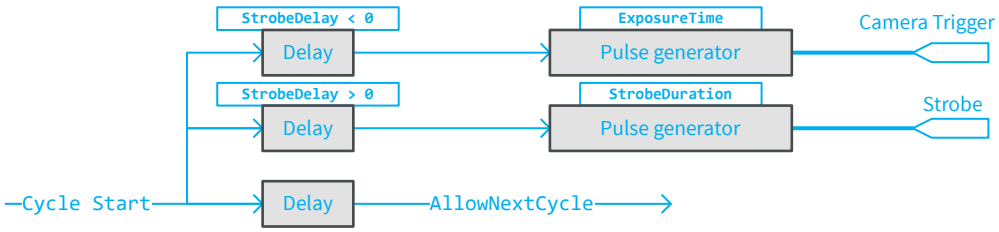
UserEvent1

It is composed of the following function blocks:

- The timing machine.
- The cycle manager.
- The cycle trigger manager.
- The sequence manager.

Cycle Timing Machine

CIC Cycle timing machine.



The CIC timing machine is responsible for the generation of accurately timed events and signals structuring one camera and illumination controller cycle (CIC Cycle).

At every occurrence of a Cycle Start event, the timing machine generates:

- One single pulse on the **Camera Trigger** signal.
- One single pulse on the **Strobe** signal.
- One **AllowNextCycle** event.

The timing of the output signals is configured by setting the following GenICam features of the Device module:

- `ExposureTime` defines the duration of the Camera Trigger pulse.
- `StobeDuration` defines the duration of the Strobe pulse.
- `StrobeDelay` defines the time offset from the leading edge of the Camera Trigger pulse up to the leading edge of the Strobe pulse.

The `AllowNextCycle` event is an used by the Cycle Manager to determine when the next Cycle may start.

The position of the `AllowNextCycle` event is not directly set by the user. Instead, it is evaluated by the Coaxlink Driver according to the following user settings:

- `ExposureReadoutOverlap` and `ExposeRecovery` features of the Camera Model category
- `CycleTargetPeriod` of the Cycle Control category

Refer to *Consecutive Overlapping CIC Cycles* on page 70 for more explanations and a timing diagrams showing various use cases.

Related Information

Single CIC Cycle on page 69
Timing diagrams of an individual CIC cycle.

Consecutive Overlapping CIC Cycles on page 70
Timing diagrams of consecutive overlapping CIC cycles.

Cycle Manager

The cycle manager is responsible for the generation of the `Cycle Start` event. It prevents initiating a new cycle while the *cycle start conditions* are not all satisfied.

Start conditions belong to two classes: the permanent conditions and the user configurable conditions.

Permanent Cycle Start Conditions Class

This class gathers the conditions that always apply; such conditions cannot be disabled by the user.

Condition	Description
Sequence Manager ACTIVE State	The sequence manager is in the ACTIVE state. *The sequence has started and the sequence stop condition has not yet been reached.*
Next Cycle Allowed	The next cycle is allowed by the timing machine. *The timing machine has already issued the `Allow Next Cycle` event of the previous cycle.*
Free Memory	There is enough free memory on board to acquire the image data of the next cycle.

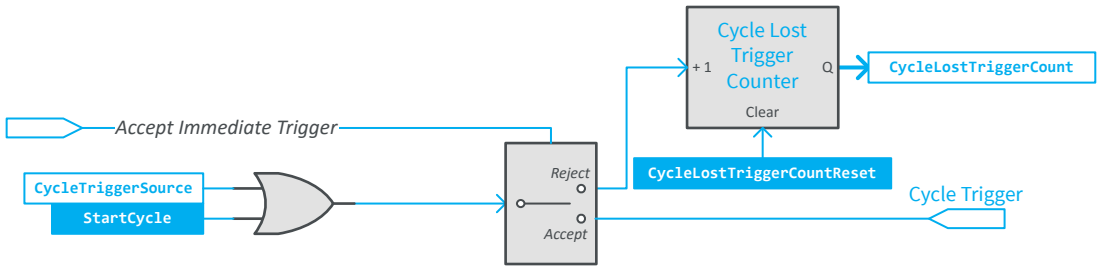
User Configurable Cycle Start Conditions Class

This class gathers the conditions that are user configurable.

Condition	Description
`CycleTriggerSource = Immediate`	A **Cycle Trigger** event is NOT required to initiate a new cycle. *The CIC cycle immediately starts when the permanent conditions are all satisfied.*
`CycleTriggerSource ≠ Immediate`	A **new Cycle Trigger event** is required to initiate a new cycle. *The CIC cycle starts on the first occurrence of a new CIC trigger event when the permanent conditions are all satisfied.*

The configurable start conditions are identical for all the cycles of a CIC cycle sequence.

Cycle Trigger Manager



The Cycle Trigger Manager is responsible, in collaboration with the Cycle Manager, to elaborate the effective Cycle Trigger event that initiates one cycle of the CIC timing machine.

Cycle Trigger Events

Cycle Trigger Events can be initiated by the application software using the `StartCycle` GenICam command or by one hardware event source specified by the `CycleTriggerSource` feature. Possible Cycle Trigger event sources are:

- Any I/O Toolbox line tool (LIN, QDC) and event tool (DIV, MDV, DEL, EIN).
- Any User Event.
- The Cycle Trigger of the Master device when `C2CLinkConfiguration = Slave`.

Cycle Trigger Filter

The Cycle Trigger Manager accepts or rejects the Cycle Triggers according to the following modalities:

Modality Name	Description
SKIP	Accept Cycle Triggers only when they can be served immediately. Triggers are accepted when **Accept Immediate Trigger** is TRUE, indicating that the Cycle Manager is currently waiting for an immediate trigger event to start a new cycle. Cycle Triggers are rejected when they occur when a permanent condition of the cycle manager is not satisfied: • Outside the sequence activity period. • When the current CIC cycle is not yet allowed by the CIC timing machine. • When there is not enough free memory in the on-board buffer.

The rejected Cycle Trigger events are discarded and a **missed cycle trigger error** is reported in Memento. They are counted by the Lost Cycle Trigger Counter an reported through the `CycleLostTriggerCount` feature.

Sequence Manager

The sequence manager is the top-level manager of the CIC. It controls the execution of sequences of camera cycles.

The Coaxlink Driver Version 6 implementation of the CIC Sequencer supports the following use cases:

Name	Description
Grab_N	On execution of the *DSStartAcquisition()* function with *iNumToAcquire=<N-value>*, the CIC Sequencer executes N consecutive CIC Cycles then stops. **Restriction:** *This mode operates exclusively when:* • `CycleTriggerSource = Immediate` • `StartOfSequenceTriggerSource = Immediate`
Grab_Infinite	On execution of the *DSStartAcquisition()* function with *iNumToAcquire=GENTL_INFINITE* the CIC Sequencer starts • Immediately when `StartOfSequenceTriggerSource = Immediate` otherwise... • at the next occurrence of a `StartSequence` command or... • at the next occurrence of an event on the specified event source. **Note:** *This mode operates with all combinations of `CycleTriggerSource` and `StartOfSequenceTriggerSource` values.* The CIC Sequencer stops at the next cycle boundary following the execution of the *DSStopAcquisition()* function. The data stream is stopped immediately.

Timing Diagrams

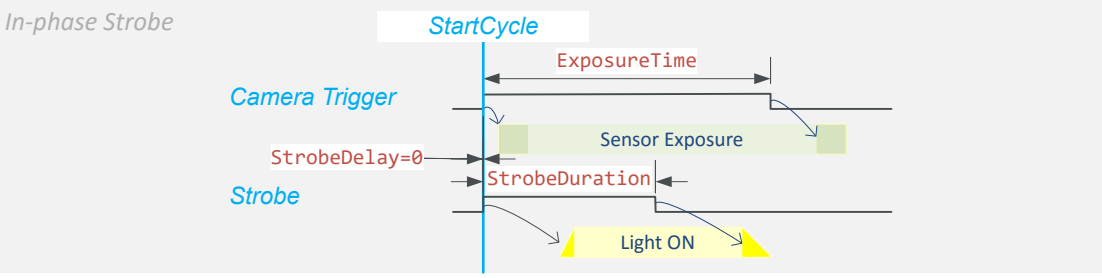
This section shows the camera cycle timing diagrams of all possible use cases of the Coaxlink Camera and Illumination Controller.

Single CIC Cycle

Timing diagrams of an individual CIC cycle.

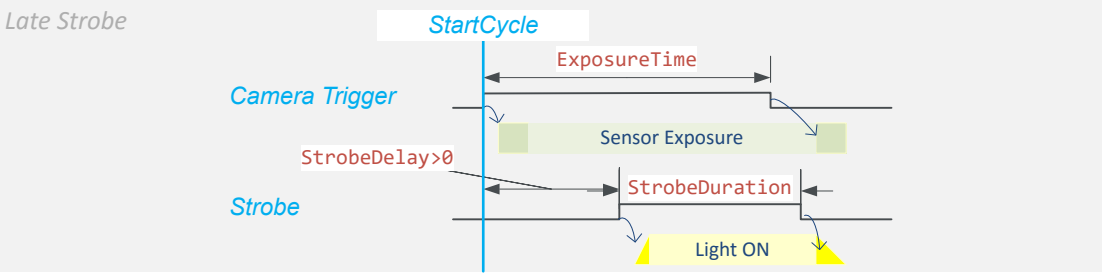
This topic illustrate the `ExposureTime`, `StrobeDuration` and `StrobeDelay` features for 3 use cases corresponding to a positive, null and negative values of `StrobeDelay`.

In-phase Strobe Case



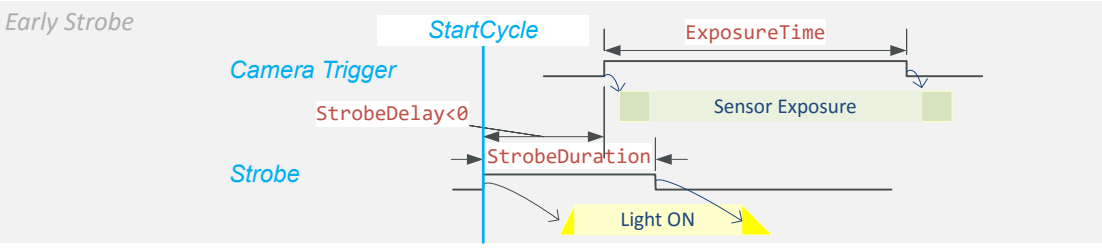
When `StrobeDelay` is set to 0, the Camera Trigger and the Strobe signals go high simultaneously.

Late Strobe Case



When `StrobeDelay` is set to a positive value, the Camera Trigger signal goes high immediately after the StartCycle event and the Strobe signal goes high after `StrobeDelay` microseconds.

Early Strobe Case



When `StrobeDelay` is set to a negative value, the Strobe signal goes high immediately after the StartCycle event and the Camera Trigger signal goes high after a time delay equal to – `StrobeDelay` microseconds.

Note: *The pulse width of the Camera Trigger and the Strobe can be adjusted independently.*

Note: *The phase between the Camera Trigger and the Strobe signal*

Consecutive Overlapping CIC Cycles

Timing diagrams of consecutive overlapping CIC cycles.

This topic illustrates how the user has to set the `ExposureRecovery` and `TargetFramePeriod` features when `ExposureReadoutOverlap = True` to properly configure the CIC to avoid trigger overruns and maximize the cycle rate.

The Coaxlink Driver calculates the duration of the Camera Cycle value from the user-defined settings `ExposureTime`, `ExposureRecovery` and `TargetFramePeriod` by searching the smallest value satisfying the following conditions:

- **Condition 1:** The time interval between consecutive Camera Trigger pulses (`r` – in the drawings) must be greater or equal to the `ExposureRecovery` settings. This ensures that the Camera Trigger properly flows through the trigger transmission link. It ensures also that a new exposure doesn't begin before the completion of the previous one.
- **Condition 2:** The CIC Cycle duration (`a` in the drawings) must be big enough to ensure that a new readout doesn't begin before the completion of the previous one.
- **Condition 3:** The CIC Cycle duration must be big enough to include both transitions of the Camera Trigger and the Strobe Signal.

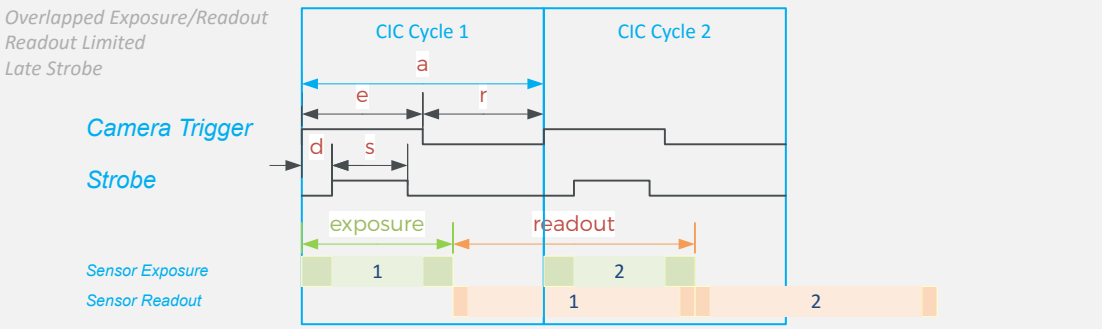
The "readout-limited" use cases illustrate situations where the cycle period is equal to the duration of the readout phase.

The "exposure-limited" use cases illustrate situations where the cycle period is equal to the duration of the exposure phase.

In the following timing diagrams:

- `e` is the Camera Trigger pulse width. This value is defined by `ExposureTime`.
- `d` is the Strobe Delay. This value is defined by `StrobeDelay`.

Case 1: Readout Limited (Late Strobe)



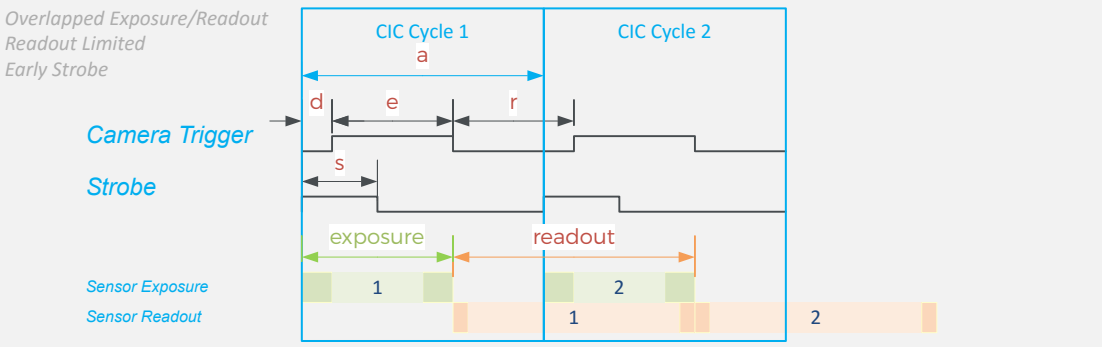
This situation occurs when `e` (= `Exposure Time`) is significantly smaller than the readout duration.
In that situation:

- `r` is likely larger that `ExposureRecovery`: **Condition 1** is fulfilled.
- The strobe pulse being "inside" the Camera Trigger pulse: **Condition 3** becomes irrelevant when **Condition 1** is fulfilled.
- The **Condition 2** is the only condition used by the Coaxlink Driver to calculate the cycle duration.

The optimal duration of the Cycle is equal to the effective duration of the sensor readout phase. This is obtained when the user sets `TargetFramePeriod` to a value corresponding to the readout duration.

Note: *The readout duration can be derived from the maximum frame rate specification of the camera data sheet or experimentally.*

Case 2: Readout Limited (Early Strobe)



This situation is similar to the case 1. It shows that despite a early strobe, it is possible to reach the maximum cycle rate of the camera.

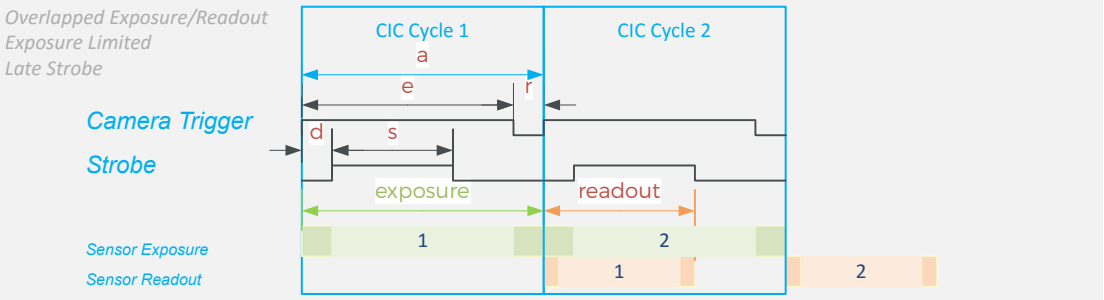
This situation occurs when `e` (= `Exposure Time`) is significantly smaller than the readout duration.
In that situation:

- `r` is likely larger that `ExposureRecovery`: **Condition 1** is fulfilled.
- The strobe pulse being terminating before the Camera Trigger pulse: **Condition 3** is fulfilled if `r` is greater than `d`. This is the case when (`d` + `e` < readout duration).
- The **Condition 2** is the only condition used by the Coaxlink Driver to calculate the cycle duration.

The optimal duration of the Cycle is equal to the effective duration of the sensor readout phase. This is obtained when the user sets `TargetFramePeriod` to a value corresponding to the readout duration.

Note: *The readout duration can be derived from the maximum frame rate specification of the camera data sheet or experimentally.*

Case 3: Exposure Limited (Late Strobe)

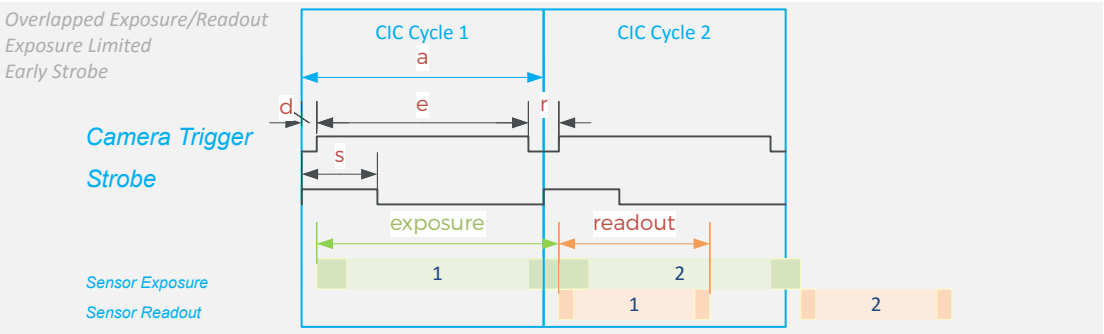


This situation occurs when `e` (= `Exposure Time`) is significantly larger than the readout duration.
In that situation:

- All cycles being identical, having the readout duration smaller than the exposure duration,implies that **Condition 2** becomes irrelevant.
- The strobe pulse being "inside" the Camera Trigger pulse: **Condition 3** becomes irrelevant when **Condition 1** is fulfilled.
- The **Condition 1** is the only condition used by the Coaxlink Driver to calculate the cycle duration .

The optimal duration of the Cycle is equal to the effective duration of the exposure phase. This is obtained when the user sets `ExposeRecovery` to a value corresponding to the minimal time interval allowed by the camera between consecutive Camera Trigger pulses.

Case 4: Exposure Limited with Early-Strobe



This situation is similar to the case 3. It shows that despite an early strobe, it is possible to reach the same cycle rate win case of small negative `StrobeDelay` values.

This situation occurs when `e` (= `Exposure Time`) is significantly larger than the readout duration.
In that situation:

- All cycles being identical, having the readout duration smaller than the exposure duration,implies that **Condition 2** becomes irrelevant.
- The strobe pulse terminating before the Camera Trigger pulse: **Condition 3** becomes irrelevant when **Condition 1** is fulfilled and `d` < `r`.
- **Condition 3** and **Condition 1** are the only condition used by the Coaxlink Driver to calculate the cycle duration.

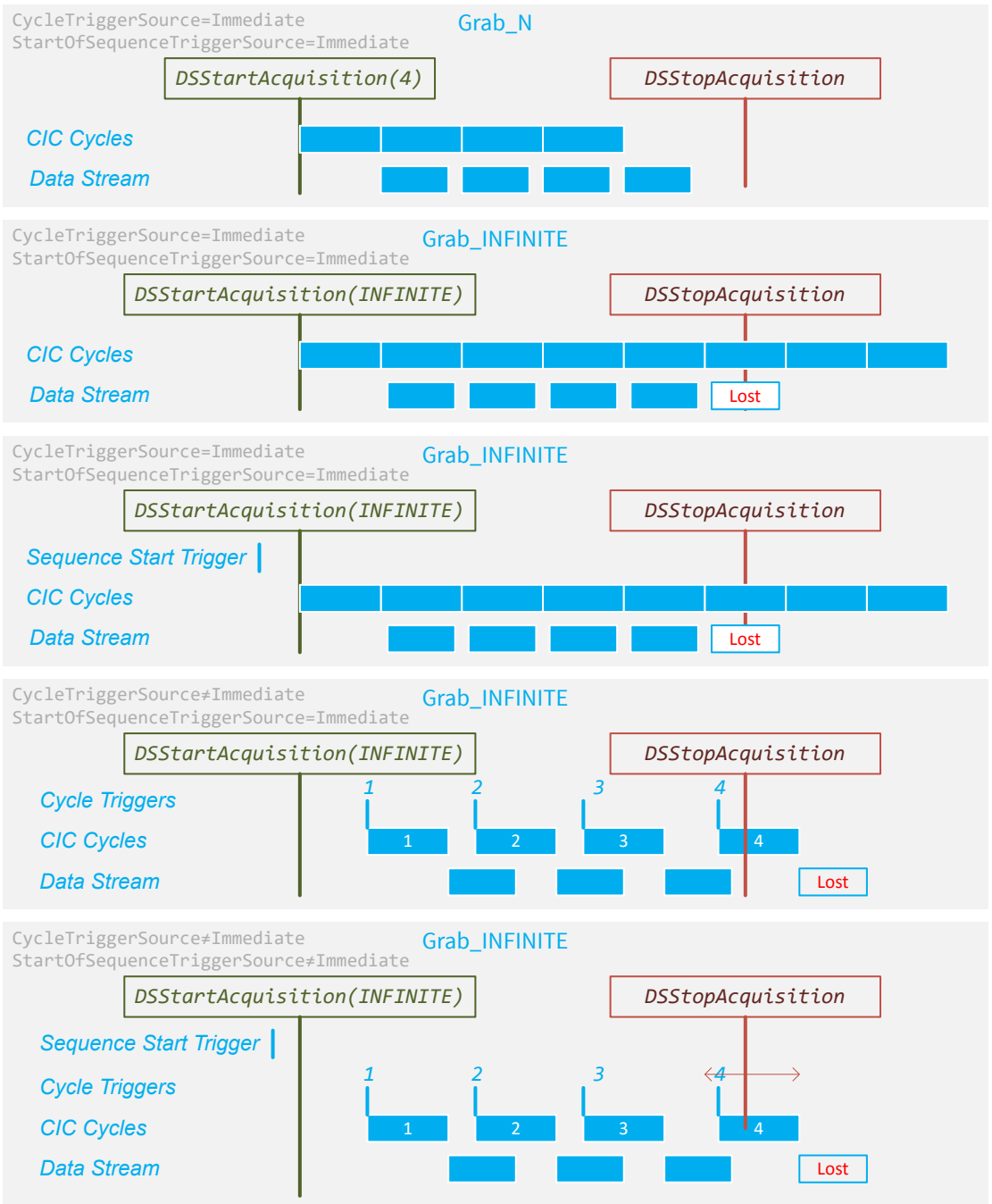
The user must set `ExposeRecovery` to a value corresponding to the largest of of the following two values:

- Minimal time interval allowed by the camera between consecutive Camera Trigger pulses.
- `-StrobeDelay`

Tip: *When `CycleTriggerSource` = `Immediate`, the cycle rate can be lowered to the desired rate by assigning a greater value to `TargetFramePeriod`.*

Cycle Sequence

Timing diagrams of CIC camera & illumination cycle sequences.



Camera Trigger Signal Paths

The frame grabber controls the camera cycle of an asynchronous reset camera by means of the Camera Trigger signal.

The signal can be transmitted from the Coaxlink card to the camera using one of the following media:

- The I/O channel of the CoaXPress link.
- A dedicated wiring driven by a TTL I/O

CoaXPress I/O channel

The Camera Trigger signal is transmitted to the camera as a high-priority Host to Device Trigger message on the CoaXPress I/O channel

For detailed information about the Host to Device Trigger transmitter, refer to *CoaXPress Host To Device Trigger* on page 34. For its configuration, refer to *Host to Device Trigger GenICam Features* on page 37.

TTL I/O Line

Any TTL I/O line can be configured as a camera trigger output. The polarity control of the I/O control block provides an individual polarity control for each I/O port. The mode control of the I/O control block of TTLIO lines provides an individual output driver configuration.

Strobe Signal Paths

I/O Lines

Every output capable I/O line can be configured as an Illumination Strobe output. The polarity control of the I/O control block provides an individual polarity control for each I/O port. The mode control of the I/O control block of TTLIO lines provides an individual output driver configuration for each I/O port used as strobe output.

CIC Synchronization

Synchronization of multiple Camera and Illumination controllers through the card-to-card link – C2C-Link.

CIC Synchronization Principle

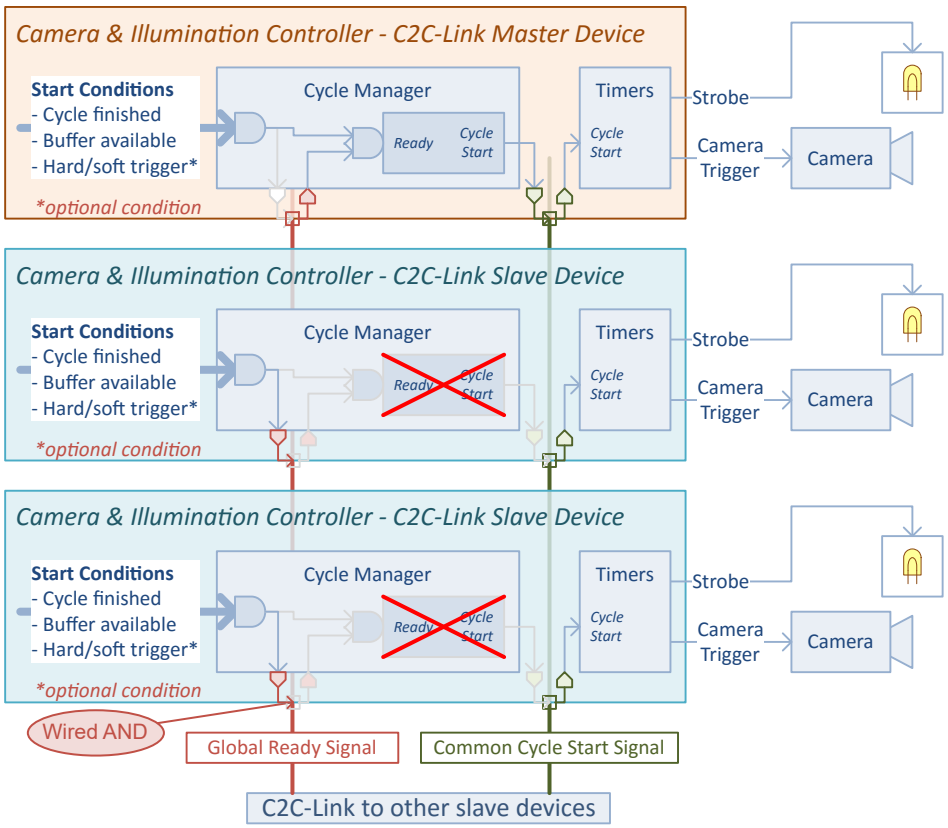
Principles of the CIC Synchronization through C2C-Link.

The synchronization is achieved by synchronizing the **CIC Cycle Trigger** of all involved CIC 's using the **C2C-Link** interconnect.

A C2C-Link interconnects two or more Camera Illumination Controllers:

- One configured as the **C2C-Link Master**.
- One (or more) configured as a **C2C-Link Slave**(s).

The following drawing shows a configuration with one C2C-Link Master (top) and two C2C-Link Slaves:



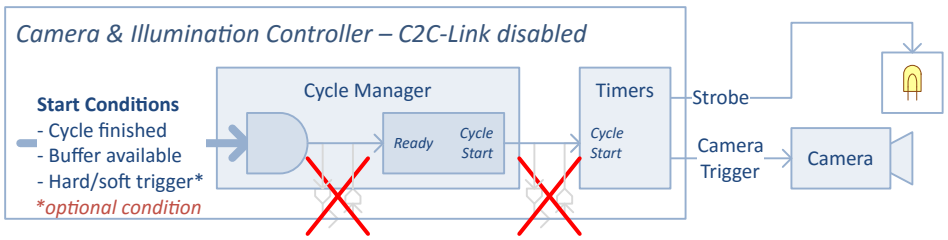
The C2C-Link Master:

- Generates a Common Cycle Start event when all the start conditions of all the synchronized CIC's, including itself, are satisfied.
- Delivers a common cycle start signal to all the synchronized CIC's.
- Upon reception of the Common Cycle Start, generates a Strobe pulse and a Camera Trigger pulse according to its timers settings.

A C2C-Link Slave:

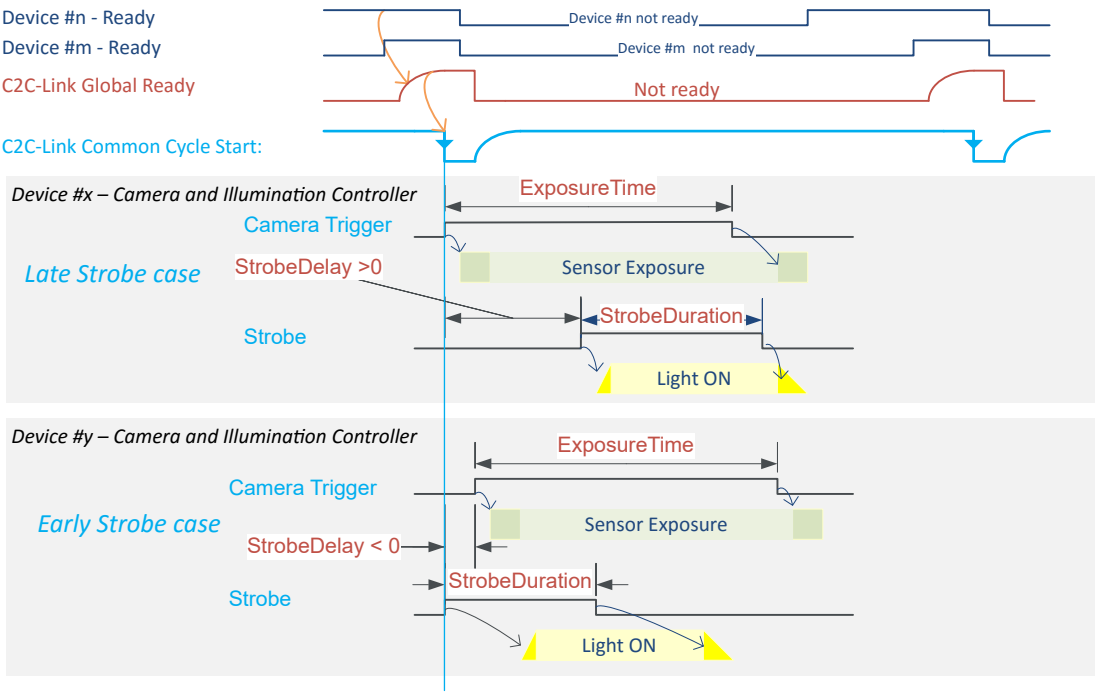
- Forces the Global Ready signal to 'false' until its start conditions are all satisfied
- Upon reception of the Common Cycle Start, generates a Strobe pulse and a Camera Trigger pulse according to its timers settings.

For comparison purpose, the following drawing shows a CIC device where the C2C-Link is disabled:



CIC Synchronization Timing Diagram

CIC Synchronization through C2C-Link timing diagram



The above diagram shows the timing diagram of two consecutive common cycle start events:

The C2C-Link Global Ready signal is held low until the Ready of all participating devices is true preventing the C2C-Link Master to issue a start event. When released by all participating devices, it ramps up rapidly with a rise time of maximum 100 ns.

As soon as the C2C-Link Global Ready signal is confirmed to be high, the master device asserts an abrupt going low transition on the Common Cycle Start signal; this edge is propagated to all the Start inputs of the timers of all participating devices.

As soon as the cycle has started, every CIC forces the ready low as long as all the local conditions to initiate the next cycle are not satisfied.

The timers of each device issue a Camera Trigger and a Strobe, with their respective delay and duration settings. Usually, the settings are identical for all participating devices; but the application is allowed to apply different ones, if needed.

The shortest cycle start period allowed by the C2C-Link is 400 ns; allowing a theoretical frequency limit of 2.5 MHz.

Camera and Illumination Control GenICam Features

CameraModel		
Device	Root > Camera and Illumination Control	Category

CameraModel		
Features describing the behavioral model of the camera.		
Note: *These features are required to check and validate the configuration settings of the Camera and Illumination Controller.*		

CycleTiming		
Device	Root > Camera and Illumination Control	Category
Features describing the timing properties of a CIC camera and illumination cycle.		
Note: *These features are required to configure the timers of the Camera and Illumination Controller.*		

CycleControl		
Device	Root > Camera and Illumination Control	Category
Features controlling a CIC camera and illumination cycle.		
Note: *These features are required to configure the triggers of the Camera and Illumination Controller.*		

SequenceControl		
Device	Root > Camera and Illumination Control	Category
Features controlling a CIC camera and illumination cycle sequence.		
Note: *These features are required to configure the sequence triggers of the Camera and Illumination Controller.*		

DeviceReset		
Device	Root > Camera and Illumination Control	Command (RW)
Device Reset.		

Camera Model GenICam Features

CameraControlMethod		
Device	Root > Camera and Illumination Control > Camera Model	Enumeration (RW)

CameraControlMethod

Sets/gets the camera control method.

Possible values:

- `NC` "NC" camera control method. *To be used with free-run or asynchronous reset cameras not controlled by the frame grabber*
- `RC`: "RC" camera control method. *To be used with asynchronous reset cameras having the camera cycle start controlled by the Coaxlink Camera and Illumination Controller and the exposure time controlled by the camera.*
- `RG`: "RG" camera control method. *To be used with asynchronous reset cameras having the camera cycle start and the exposure duration controlled by the Coaxlink Camera and Illumination Controller.*
- `EXTERNAL`: "External" camera control method. *To be used with asynchronous reset cameras having the camera cycle start and the exposure duration controlled by an external hardware signal applied on any GPIO input port.*

C2CLinkConfiguration

Device	Root > Camera and Illumination Control > Camera Model	Enumeration (RW)

Sets/gets the C2C-Link configuration.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Possible values:

- `Disconnected`: Disconnected from the C2C-Link.
- `Master`: Connected to the C2C-Link as the *C2C-Link Master Device*.
- `Slave`: Connected to the C2C-Link as a*C2C-Link Slave Device*.

ExposureReadoutOverlap

Device	Root > Camera and Illumination Control > Camera Model	Boolean (RW)

Sets/gets the exposure overlapping capability.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Possible values:

- `Off`:for cameras not allowing exposure overlapping
- `On`: for cameras allowing exposure overlapping

ExposureRecoveryTime

Device	Root > Camera and Illumination Control > Camera Model	Float (RW)

Sets/gets the exposure recovery time.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Unit: microsecond.

Value range: from 0.0 up to 17,179,869.1.

Default value: 1,000.0. *(1 millisecond)*

Cycle Timing GenICam Features

ExposureTime		
Device	Root > Camera and Illumination Control > Cycle Timing	Float (RW)

Sets/gets the exposure time.
Applies only when `CameraControlMethod` is `RG`.
Unit: microsecond.
Value range: from 3.36 up to 562,949,953,421. *(> 150 hours)*
Default value: 1,000.0. *(1 millisecond)*

StrobeDelay		
Device	Root > Camera and Illumination Control > Cycle Timing	Float (RW)

Sets/gets the strobe pulse delay.
Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).
Unit: microsecond.
Value range: from -8,589,934.5 up to 8,589,934.5.
Default value: 0

StrobeDuration		
Device	Root > Camera and Illumination Control > Cycle Timing	Float (RW)

Sets/gets the strobe pulse duration.
Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).
Unit: microsecond.
Value range: from 0 up to 562,949,953,421. *(> 150 hours)*
Default value: 1,000.0. *(1 millisecond)*

Cycle Control GenICam Features

CycleTriggerSource		
Device	Root > Camera and Illumination Control > Cycle Control	Enumeration (RW)

CycleTriggerSource

Sets/gets the start-of-camera-cycle trigger conditions and selects a hardware or software trigger source.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Possible values:

- `Immediate`: Immediately after `DSStartAcquisition()` is called and then repeatedly every `CyclePeriodTarget` period. **Default value.**
- `StartCycle`: On execution of the `StartCycle` command.
- `C2C`: Synchronized with the C2C-Link master device. *This value is enforced when* `C2CLinkConfiguration` = `Slave`
- `LIN1`: When an event occurs on I/O toolbox LIN1 or on execution of the `StartCycle` command.
- `LIN2`: When an event occurs on I/O toolbox LIN2 or on execution of the `StartCycle` command.
- `LIN3`: When an event occurs on I/O toolbox LIN3 or on execution of the `StartCycle` command.
- `LIN4`: When an event occurs on I/O toolbox LIN4 or on execution of the `StartCycle` command.
- `LIN5`: When an event occurs on I/O toolbox LIN5 or on execution of the `StartCycle` command.
- `LIN6`: When an event occurs on I/O toolbox LIN6 or on execution of the `StartCycle` command.
- `LIN7`: When an event occurs on I/O toolbox LIN7 or on execution of the `StartCycle` command.
- `LIN8`: When an event occurs on I/O toolbox LIN8 or on execution of the `StartCycle` command.
- `QDC1`: When an event occurs on I/O toolbox QDC1 or on execution of the `StartCycle` command.
- `QDC2`: When an event occurs on I/O toolbox QDC2 or on execution of the `StartCycle` command.
- `DIV1`: When an event occurs on I/O toolbox DIV1 or on execution of the `StartCycle` command.
- `DIV2`: When an event occurs on I/O toolbox DIV2 or on execution of the `StartCycle` command.
- `MDV1`: When an event occurs on I/O toolbox MDV1 or on execution of the `StartCycle` command.
- `MDV2`: When an event occurs on I/O toolbox MDV2 or on execution of the `StartCycle` command.
- `DEL11`: When an event occurs on Output 1 of I/O toolbox DEL1 or on execution of the `StartCycle` command.
- `DEL12`: When an event occurs on Output 2 of I/O toolbox DEL1 or on execution of the `StartCycle` command.
- `DEL21`: When an event occurs on Output 1 of I/O toolbox DEL2 or on execution of the `StartCycle` command.
- `DEL22`: When an event occurs on Output 2 of I/O toolbox DEL2 or on execution of the `StartCycle` command.
- `EIN1`: When an event occurs on I/O toolbox EIN1 or on execution of the `StartCycle` command.
- `EIN2`: When an event occurs on I/O toolbox EIN2 or on execution of the `StartCycle` command.
- `UserEvent1`: When a User Event 1 occurs or on execution of the `StartCycle` command.
- `UserEvent2`: When a User Event 2 occurs or on execution of the `StartCycle` command.
- `UserEvent3`: When a User Event 3 occurs or on execution of the `StartCycle` command.
- `UserEvent4`: When a User Event 4 occurs or on execution of the `StartCycle` command.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

CycleTargetPeriod

Device	Root > Camera and Illumination Control > Cycle Control	Float (RW)

Sets/gets the target cycle period.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`) and when `CycleTriggerSource` = `Immediate`.

Note: *was named `CyclePeriodTarget` in Coaxlink Driver versions prior to 4.1.*

Unit: microsecond.

Value range: from 5.0 up to 562,949,953,421. *(> 150 hours)*

Default value: 1,000.0. *(1 millisecond)*

StartCycle		
Device	Root > Camera and Illumination Control > Cycle Control	Command (WO)

Starts a camera cycle.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Note: *was named `CycleSoftwareTrigger` in Coaxlink Driver versions prior to 4.1.*

CycleLostTriggerCount		
Device	Root > Camera and Illumination Control > Cycle Control	Integer (RO)

Returns the count of lost CIC cycle trigger events.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Value range: from 0 up to 4,294,967,295.

CycleLostTriggerCountReset		
Device	Root > Camera and Illumination Control > Cycle Control	Command (WO)

Resets the count of lost CIC cycle trigger events.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Sequence Control GenICam Features

StartOfSequenceTriggerSource		
Device	Root > Camera and Illumination Control > Cycle Control	Enumeration (RW)

StartOfSequenceTriggerSource

Sets/gets the start-of-sequence trigger conditions and selects a hardware or software trigger source.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Possible values:

- `Immediate`: Immediately after `DSStartAcquisition()` is called. **Default value.**
- `StartSequence`: On execution of the `StartSequence` command.
- `LIN1`: When an event occurs on I/O toolbox LIN1 or on execution of the `StartSequence` command.
- `LIN2`: When an event occurs on I/O toolbox LIN2 or on execution of the `StartSequence` command.
- `LIN3`: When an event occurs on I/O toolbox LIN3 or on execution of the `StartSequence` command.
- `LIN4`: When an event occurs on I/O toolbox LIN4 or on execution of the `StartSequence` command.
- `LIN5`: When an event occurs on I/O toolbox LIN5 or on execution of the `StartSequence` command.
- `LIN6`: When an event occurs on I/O toolbox LIN6 or on execution of the `StartSequence` command.
- `LIN7`: When an event occurs on I/O toolbox LIN7 or on execution of the `StartSequence` command.
- `LIN8`: When an event occurs on I/O toolbox LIN8 or on execution of the `StartSequence` command.
- `QDC1`: When an event occurs on I/O toolbox QDC1 or on execution of the `StartSequence` command.
- `QDC2`: When an event occurs on I/O toolbox QDC2 or on execution of the `StartSequence` command.
- `DIV1`: When an event occurs on I/O toolbox DIV1 or on execution of the `StartSequence` command.
- `DIV2`: When an event occurs on I/O toolbox DIV2 or on execution of the `StartSequence` command.
- `MDV1`: When an event occurs on I/O toolbox MDV1 or on execution of the `StartSequence` command.
- `MDV2`: When an event occurs on I/O toolbox MDV2 or on execution of the `StartSequence` command.
- `DEL11`: When an event occurs on Output 1 of I/O toolbox DEL1 or on execution of the `StartSequence` command.
- `DEL12`: When an event occurs on Output 2 of I/O toolbox DEL1 or on execution of the `StartSequence` command.
- `DEL21`: When an event occurs on Output 1 of I/O toolbox DEL2 or on execution of the `StartSequence` command.
- `DEL22`: When an event occurs on Output 2 of I/O toolbox DEL2 or on execution of the `StartSequence` command.
- `EIN1`: When an event occurs on I/O toolbox EIN1 or on execution of the `StartSequence` command.
- `EIN2`: When an event occurs on I/O toolbox EIN2 or on execution of the `StartSequence` command.
- `UserEvent1`: When a User Event 1 occurs or on execution of the `StartSequence` command.
- `UserEvent2`: When a User Event 2 occurs or on execution of the `StartSequence` command.
- `UserEvent3`: When a User Event 3 occurs or on execution of the `StartSequence` command.
- `UserEvent4`: When a User Event 4 occurs or on execution of the `StartSequence` command.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

StartSequence

Device	Root > Camera and Illumination Control > Sequence Control	Command (WO)

Starts a CIC sequence.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`) and `StartOfSequenceTriggerSource` is not set to `Immediate`.

AbortSequence

Device	Root > Camera and Illumination Control > Sequence Control	Command (WO)

Abort a CIC sequence.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`) and `StartOfSequenceTriggerSource` is not set to `Immediate`.

C2C-Link

Card-to-card Link description.

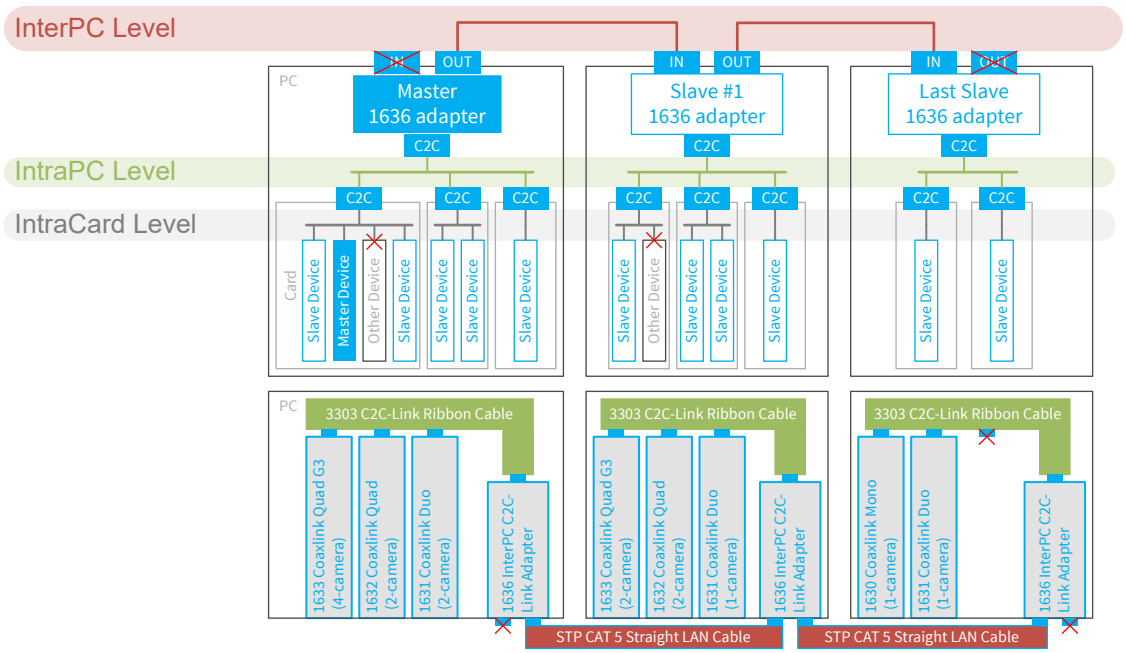
The C2C-Link is a hardware communication medium allowing a single "**C2C-Link master device**" to **reliably** share 1 or 2 trigger events with multiple "**C2C-Link slave devices**".

The C2C-Link is scalable: it may interconnect devices belonging to the same Coaxlink card, or to different cards in the same PC or to different cards in different PCs.

The C2C-Link feedback channel allows any C2C-Link device to indicate to the C2C-Link master device that it is not able to accept a trigger.

A C2C-Link interconnection may combine up to three interconnection levels:

- The **IntraCard Level** interconnects 2 or more C2C-Link devices belonging to the same card using FPGA internal resources.
- The **IntraPC Level** interconnects C2C-Link devices across two or more cards of the same PC. It requires one accessory cable such as the 3303 C2C-Link Ribbon Cable or a custom-made C2C-Link cable for each PC.
- The **InterPC Level** interconnects C2C-Link devices across two or more PCs. It requires one 1636 InterPC C2C-Link Adapter for each PC and one RJ 45 CAT 5 STP straight LAN cable for each adapter but the last one.



C2C-Link Configurations

C2C-Link Configuration	Interconnection Levels
IntraCard	IntraCard only
IntraPC	IntraPC + IntraCard (optional)
InterPC	InterPC + IntraPC+ IntraCard (optional)

C2C-Link Specification

C2C-Link operating limits per interconnection level

Definitions

Trigger delay

Propagation delay of the trigger signal from the master device to a slave device. This delay is composed of the propagation delay inside electronic devices (FPGA, adapter...) and interconnection cables.

For cables, the delay is proportional to the cable length, typically: 5ns/m.

Trigger delay skew

Dispersion of the trigger delay values across all the devices belonging to a C2C-Link.

Trigger delay jitter

Variation of the trigger delay depending on external factors such as temperature, signal noise...

Trigger rate

Rate of occurrence of repetitive trigger events. The reciprocal value (1/Trigger rate) is the minimum time interval required between consecutive triggers.

IntraCard C2C-Link Interconnection Level

Parameter	Min.	Typ.	Max.	Units
Trigger delay jitter			5	ns
Trigger rate	0		2.5	MHz

IntraPC C2C-Link Interconnection Level

Parameter	Min.	Typ.	Max.	Units
Cable connectors count (including 1636 adapter if any)	2		4	-
Cable length	–		0.6	m
Trigger delay jitter	5		10	ns
Trigger rate	0		2.5	MHz

InterPC C2C-Link Interconnection Level

The following specification targets applications where the **highest trigger rate** is required:

Parameter	Min.	Typ.	Max.	Units
1636 adapters count	2		10	-
Adapter-to-adapter LAN cable length	0.3		100	m
Cumulated adapter-to-adapter LAN cable length	0.3		100	m
Trigger delay jitter	10	20	40	ns
Trigger rate	0		2.5	MHz

The following specification targets applications where the **longest reachable distance** is required:

Parameter	Min.	Typ.	Max.	Units
1636 adapters count	2		32	-
Adapter-to-adapter cable length	0.3		1200	m
Cumulated adapter-to-adapter cable length	0.3		1200	m
Trigger delay jitter			1	µs
Trigger rate	0		200	kHz

Note: *The maximum trigger rate specification can be extrapolated for intermediate distances between 100 m and 1200 m assuming that the length x frequency product is constant: in this case 250 [m. MHz].*

Trigger Propagation Delays

Estimating the trigger propagation delays.

The propagation delay of the Trigger signals from the master device to a slave device can be roughly estimated by adding the typical delays encountered in each segment of the signal path.

Typical Delay values per C2C-Link segment

Delay Element	Min.	Typ.	Max.	Units
(1) IntraPC interconnection (Whole IntraPC C2C-Link interconnect including FPGA I/O and IntraPC cable)	0	5	10	ns
(2) 1636 InterPC C2C-Link Adapter – IntraPC-to-InterPC delay		15		ns
(3) 1636 InterPC C2C-Link Adapter – InterPC-to-IntraPC delay		20		ns
(4) 1636 InterPC C2C-Link Adapter – InterPC-to-InterPC delay		0		ns
(5) InterPC LAN cable		5		ns/m

Example 1 – IntraPC Configuration

For an IntraPC only configuration there is only one delay element to consider: (1)

Parameter	Min.	Typ.	Max.	Units
(1) IntraPC interconnection	0	5	10	ns
Total Trigger Delay	**0**	**5**	**10**	**ns**

Example 2 – 3-adapter InterPC Configuration; 20 m+20 m LAN cable

This configuration is composed of 3 Intra-PC segments. For devices belonging to the same IntraPC segment as the Master device, there is only one element to consider.

Parameter	Min.	Typ.	Max.	Units
(1) IntraPC interconnection	0	5	10	ns

Parameter	Min.	Typ.	Max.	Units
Total Trigger Delay for devices of the Master InterPC segment	**0**	**5**	**10**	**ns**

For devices belonging to the same IntraPC segment as the Slave1 adapter, there are 5 delay elements to consider:

Parameter	Min.	Typ.	Max.	Units
(1) Master IntraPC interconnection	0	5	10	ns
(2) Master 1636 InterPC C2C-Link Adapter – IntraPC-to-InterPC delay		15		ns
(5) Master to slave1 InterPC LAN cable – 20 m		100		ns
(3) Slave1 1636 InterPC C2C-Link Adapter – InterPC-to-IntraPC delay		20		ns
(1) Slave1 IntraPC interconnection	0	5	10	ns
Total Trigger Delay for devices of the Slave1 InterPC segment		**145**		**ns**

For devices belonging to the same IntraPC segment as the Slave2 adapter, there are 7 delay elements to consider:

Parameter	Min.	Typ.	Max.	Units
(1) Master IntraPC interconnection	0	5	10	ns
(2) Master 1636 InterPC C2C-Link Adapter – IntraPC-to-InterPC delay		15		ns
(5) Master to slave1 InterPC LAN cable – 20 m		100		ns
(4) Slave1 1636 InterPC C2C-Link Adapter – InterPC-to-InterPC delay		0		ns
(5) Slave1 to slave2 InterPC LAN cable – 20 m		100		ns
(3) Slave2 1636 InterPC C2C-Link Adapter – InterPC-to-IntraPC delay		20		ns
(1) Slave2 IntraPC interconnection	0	5	10	ns
Total Trigger Delay for devices of the Slave2 InterPC segment		**245**		**ns**

C2C-Link Use Cases

Possible use cases of the C2C-Link.

CIC Synchronization

The Cycle Trigger event of the Camera and Illumination Controller of the "C2C-Link Master Device" is distributed to one or more "C2C-Link Slave Devices".

The CIC Cycles of all the devices start simultaneously. However, the Cycle Timing parameters can be configured individually.

Refer to *CIC Synchronization* on page 74 for more information.

C2C-Link Availability

Availability of the C2C-Link vs. product and firmware variant.

One C2C-Link connector is available on all Coaxlink products for extension of the C2C-Link beyond the card boundary.

One **IntraCard C2C-Link** interconnect is available only in multi-device firmware variants, namely:

- 2-camera
- 2-camera, line-scan
- 4-camera

CIC Synchronization is available on all firmware variants except:

- 1-df-camera
- 1-df-camera, line-scan

C2C-Link Setup

How to setup a C2C-Link?

General Procedure

1. **C2C-Link hardware setup.** This step is specific to each C2C-Link Configuration. Refer to sections hereafter.
2. **GenAPI setup**. Assign a value to the `C2CLinkConfiguration` GenTL Device feature.

C2CLinkConfiguration		
Device	Root > Camera and Illumination Control > Camera Model	Enumeration (RW)

Sets/gets the C2C-Link configuration.

Applies only when the CIC is used (i.e., when `CameraControlMethod` is `RC` or `RG`).

Possible values:

- `Disconnected`: Disconnected from the C2C-Link.
- `Master`: Connected to the C2C-Link as the *C2C-Link Master Device*.
- `Slave`: Connected to the C2C-Link as a*C2C-Link Slave Device*.

IntraCard C2C-Link Configuration Hardware Setup

This configuration exclusively uses FPGA internal resources to build the C2C-Link interconnect; it doesn't require any additional hardware!

IntraPC C2C-Link Configuration Hardware Setup

This configuration requires one accessory cable such as the 3303 C2C-Link Ribbon Cable or a custom-made C2C-Link cable for each PC.

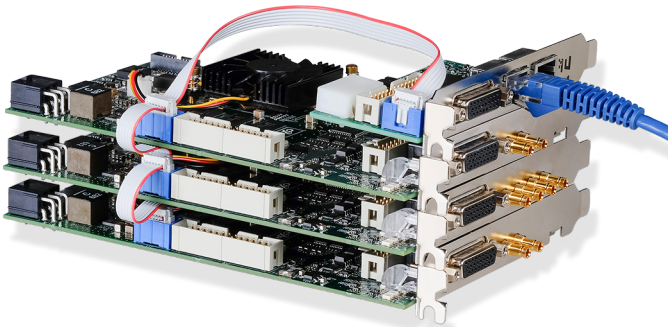
Insert a C2C-Link female connector of the C2C-Link cable into the C2C-Link pin header connector of each participating Coaxlink cards.

InterPC C2C-Link Configuration Hardware Setup

This configuration requires one 1636 InterPC C2C-Link Adapter for each PC and one RJ 45 CAT 5 STP straight LAN cable for each adapter but the last one.

In each participating PC:

1. Install 1636 InterPC C2C-Link Adapter into a free slot and secure the bracket.
2. Connect the adapter to a power source. Refer to *Adapter Powering* on page 91.
3. Using one 3303 C2C-Link Ribbon Cable, bind together the C2C-Link connectors of all the participating cards together with the C2C-Link of the adapter card.
4. Using LAN Cables, interconnect the adapters. Refer to *InterPC Interconnect* on page 92.



C2C-Link Hardware Components

Hardware components of the IntraPC and the InterPC C2C-Link interconnections.

IntraPC Interconnection Level Hardware

3303 C2C-Link Ribbon Cable on page 88

Custom C2C-Link Ribbon Cable Assembly on page 89

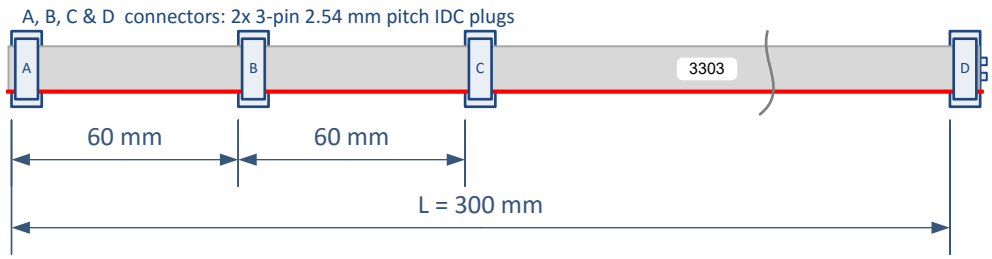
InterPC Interconnection Level Hardware

1636 InterPC C2C-Link Adapter on page 89

3303 C2C-Link Ribbon Cable

Accessory product used for Intra-PC C2C-Link interconnection.

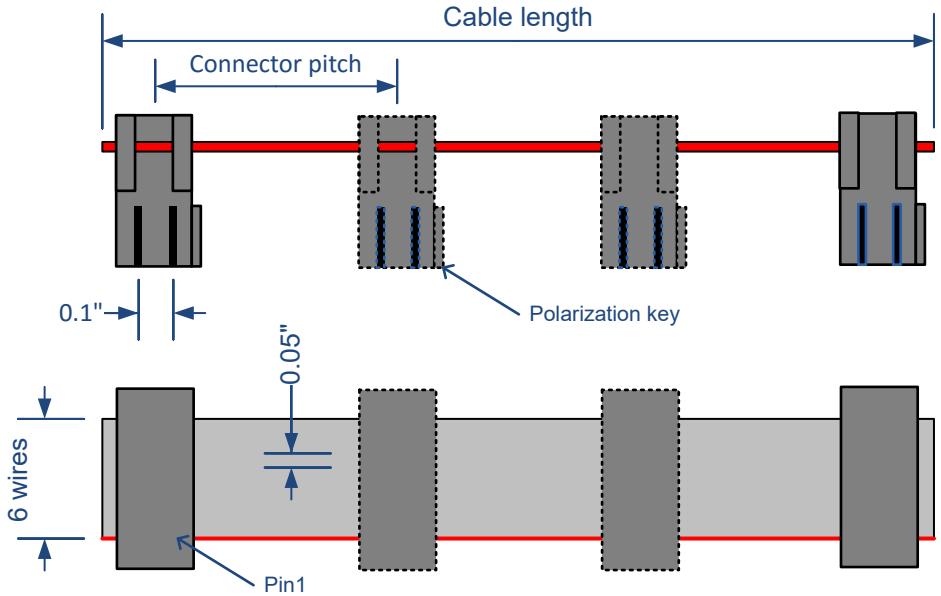
3303 C2C-Link Ribbon Cable



The 3303 C2C-Link Ribbon Cable is a 6-conductor 0.05-in pitch ribbon fitted with 4 6-pin female ribbon cable connectors. This cable is used for interconnecting the C2C-Link connectors of up to 4 cards located in the same PC.

Custom C2C-Link Ribbon Cable Assembly

Assembly instructions of a custom-made IntraPC C2C-Link interconnection.



The cable assembly is composed with:

- A piece of a 6-conductor 0.05-in pitch ribbon cable. For instance: *Belden's (9L280XX Series)*.
- Two or more pieces of a 2 x 3-pin female ribbon cable connectors. For instance: *TE connectivity 1-1658528-1*.

The cable assembly has:

- A maximum of 4 connectors allowing up to 4 cards to share the same C2C-Link.
- A maximum length of 60 cm.

The connector pitch(es) must be determined according to the actual card to card spacing in the Host PC.

1636 InterPC C2C-Link Adapter

*Accessory product for use as an **InterPC C2C-Link extender** and/or as a **HD26F I/O adapter**.*

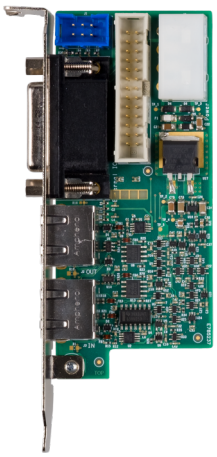
For a 1636 InterPC C2C-Link Adapter hardware description, refer to *Hardware Description* on page 90.

For a description of the C2C-Link extender usage, refer to *C2C-Link Extender* on page 91.

For a description of the HD26F I/O adapter usage, refer to *HD26F I/O Adapter* on page 94.

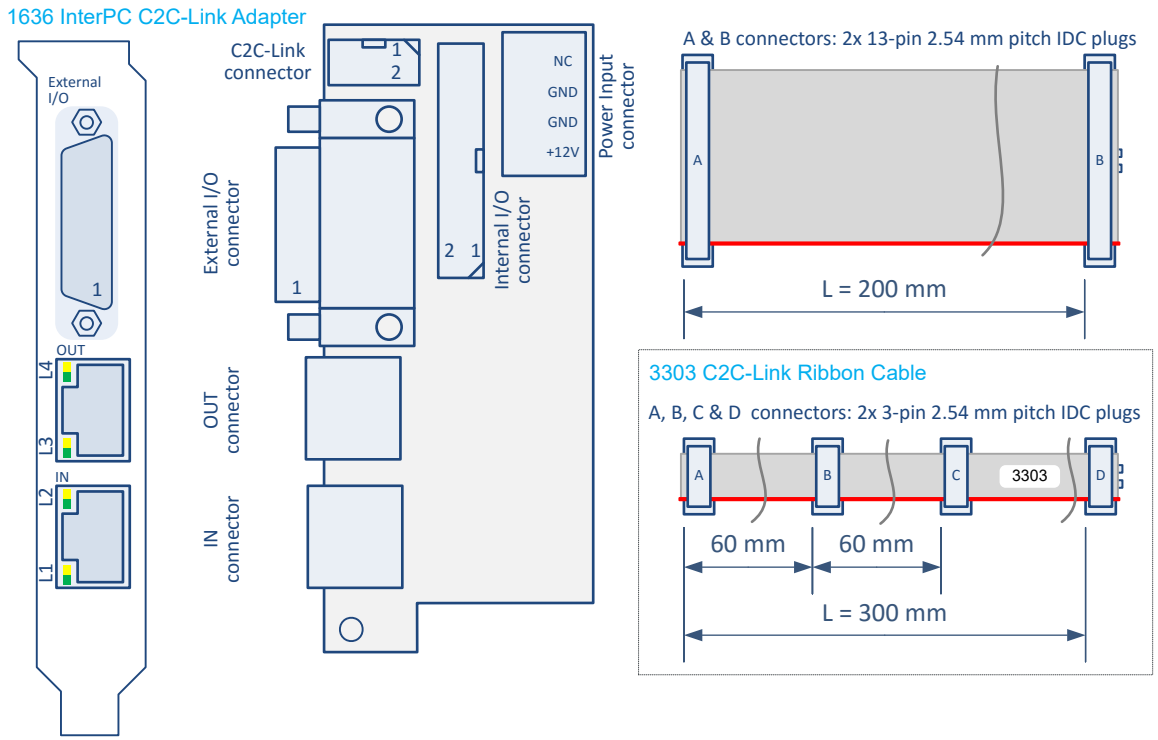
Product Pictures

Pictures of 1636 InterPC C2C-Link Adapter



Hardware Description

Detailed hardware description of 1636 InterPC C2C-Link Adapter



The 1636 InterPC C2C-Link Adapter product accessory is composed of:

- A printed circuit board assembly fitted with a standard-profile PC bracket.

- A 200-mm 26-way ribbon cable.
- A 3303 C2C-Link Ribbon Cable.

Connectors

The **External I/O connector** is a HD26F – 26-pin 3-row high-density female – Sub-D connector fitted on the bracket with UNC 4-40 screws. Refer to *HD26F I/O Adapter* on page 94 for the usage description and the pin assignments.

The **IN connector** and the **OUT connector** are RJ-45 8-pin sockets fitted on the bracket. Refer to *C2C-Link Extender* on page 91 for the usage description.

The **Internal I/O connector** is a 26-pin dual-row 0.1" pitch pin header with shrouding. Refer to *HD26F I/O Adapter* on page 94 for the usage description and the pin assignments.

The **C2C-Link connector** is a 6-pin dual-row 0.1" pitch pin header with shrouding. Refer to *C2C-Link Extender* on page 91 for the usage description.

The **Internal I/O connector** is a 26-pin dual-row 0.1" pitch pin header with shrouding. Refer to *HD26F I/O Adapter* on page 94 for the usage description.

The **Power Input connector** is a 0.2" pitch right-angled Disk Drive Power connector. Refer to *C2C-Link Extender* on page 91 for the usage description.

Lamps

The **IN connector** and the **OUT connector** are each equipped with 2 green/yellow LED lamps named respectively **L1**, **L2**, **L3** and **L4**. Refer to *C2C-Link Extender* on page 91 topic for the usage description.

C2C-Link Extender

Using 1636 InterPC C2C-Link Adapter as a C2C-Link Extender.

Adapter Powering

How to deliver power to the 1636 InterPC C2C-Link Adapter?

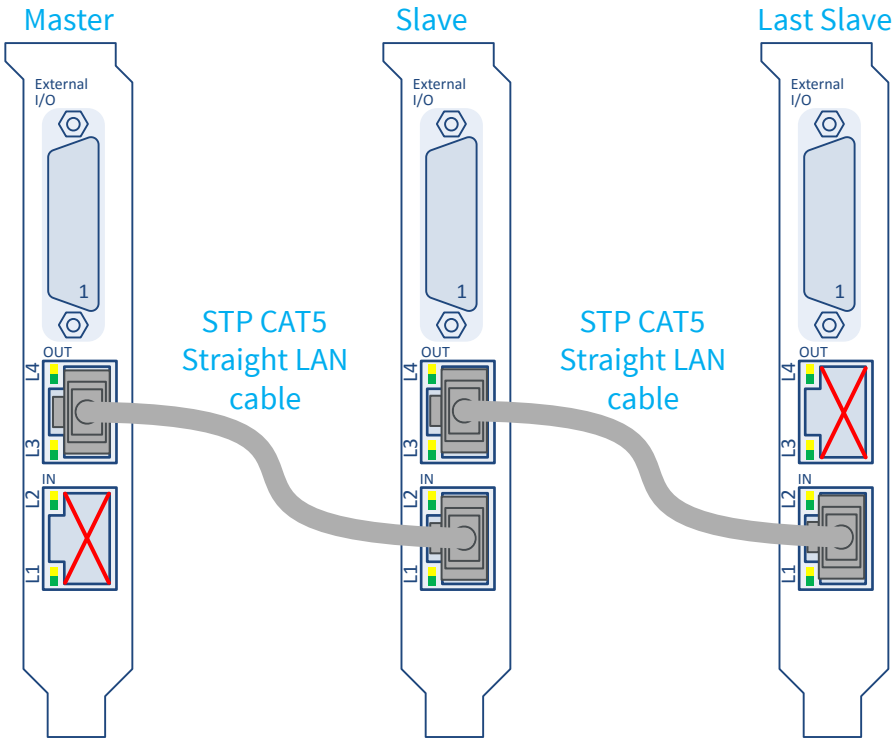
Notice: *The 1636 InterPC C2C-Link Adapter must be powered when it is used as a C2C-Link extender.*

The user has two options to supply power to the adapter:

- From the Coaxlink card +12V power output of an Internal IO connector through the 26-way ribbon cable attached to the **Internal I/O** connector.
- From the Host PC power supply through a Disk Drive Power connector cable plugged into the **Power Input connector**.

Parameter	Min.	Typ.	Max.	Units
+12 V DC Input voltage	11.0	12.0	13.0	V
+12 V Input power		1.8		W

InterPC Interconnect



The above drawing shows the external wiring of a C2C-Link across 3 adapters.
The external wiring of the C2C-Link is made with RJ 45 CAT 5 STP straight LAN cables. N-1 cables are required to interconnect N adapters in a daisy-chain scheme.

The daisy-chain begins on the OUT connector of the Master adapter and ends at the IN connector of the Last Slave adapter.

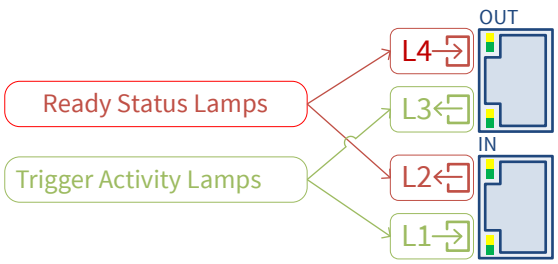
The IN connector of the Master adapter and the OUT connector of the Last Slave adapter are unused.

Note: *The adapter disables the signal drivers of the IN and OUT connectors to avoid electrical damages when it detects a bad or a missing connection.*

The InterPC cable drivers and receivers are not electrically isolated.

Important: *To avoid damages, the interconnected PCs must have a common ground reference.*

Lamps



Trigger Activity Lamps

The L1 and L3 Lamps indicate the trigger activity on the LAN cable. L1 shows the activity on the received trigger signals; L3 shows the activity on the transmitted trigger signals.

Lamp State	Indication
Off	The LAN cable is unplugged or the adapter is not powered.
Green	No trigger activity. *No trigger events in the past 10 milliseconds.*
Yellow	Trigger activity. *One or more trigger events in the past 10 milliseconds.*

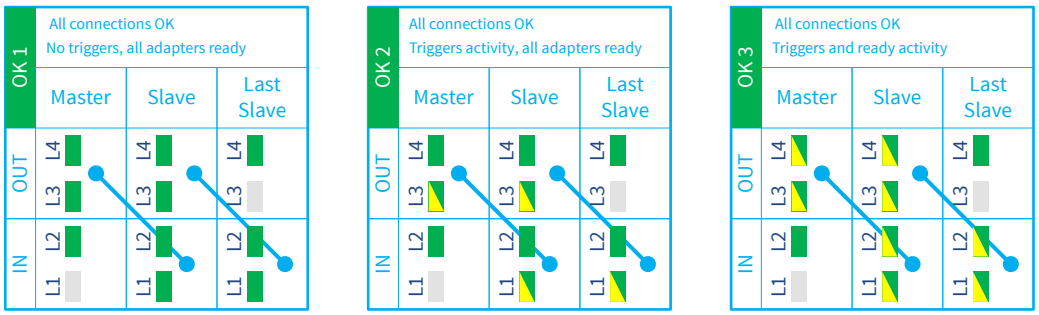
Ready Status Lamps

The L2 and L4 Lamps indicate the state of the ready signal on the LAN cable. L2 shows the state of the transmitted ready signal; L4 shows the state of the received ready signal.

Lamp State	Indication
Off	The adapter is not powered.
Green	Ready true. *For L2: all the C2C-Link devices attached to this adapter and the downwards adapters (if any) are ready.* *For L4: all the C2C-Link devices attached to the downwards adapters (if any) are ready.*
Yellow	Ready false. *For L2: one or more C2C-Link devices attached to this adapter and the downwards adapters (if any) are not ready.* *For L4: one or more C2C-Link devices attached to the downwards adapters (if any) are not ready.*

Note: *Unlike the trigger activity lamps, the ready signals are not enlarged. Short-duration not-ready states are hardly visible!*

Adapters Array Lamp States – Normal Situations



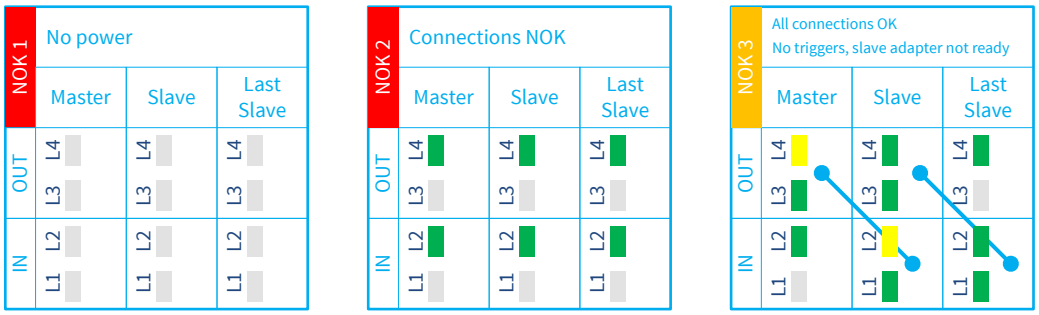
The above drawings show the lamps states of 3 daisy-chained adapters for 3 normal situations.

In the OK 1 situation, all adapters are ready to accept triggers but no triggers are sent by the master.

In the OK 2 situation, the master adapter sends triggers and the ready signal of all adapters is permanently high. The yellow/green toggling L1 and L3 lamps indicate the trigger activity. The steady green L2 and L4 lamps indicate that all adapters are permanently ready to receive triggers.

In the OK 3 situation, the master adapter sends triggers and the ready signal of all adapters is cycling. The yellow/green toggling L1 and L3 lamps indicate the trigger activity. The yellow/green toggling L2 and L4 lamps indicate that all adapters are not ready to receive triggers for a significant duration.

Adapters Array Lamp States – Abnormal Situations



The above drawings show the lamps states of 3 daisy-chained adapters for 3 abnormal situations.

In the NOK 1 situation, no adapters are powered. All lamps are Off.

In the NOK 2 situation, all adapters are powered but all connections are missing or incorrect.

In the NOK 3 situation, all adapters are powered and all connections are OK, but the second adapter is not ready preventing the master to send new triggers. This situation is considered as abnormal when it persists.

Troubleshooting Guide

Lamps state	Indication and possible causes	Action
All lamps Off	The adapter is not powered.	Apply power to the adapter
L1 Off L2 Green	The external connection to the IN connector is missing or incorrect.	For the master adapter, this is OK: nothing to do! For the other adapters: check and correct the connection to the OUT connector of the previous adapter in the daisy-chain.
L3 Off L4 Green	The external connection to the OUT connector is missing or incorrect.	For the last slave adapter of the daisy-chain, this is OK: nothing to do! For the other adapters: check and correct the connection to the IN connector of the next adapter in the daisy-chain.

HD26F I/O Adapter

Using the 1636 InterPC C2C-Link Adapter as a HD26F I/O Adapter.

To use 1636 InterPC C2C-Link Adapter as an HD26F I/O adapter:

- Plug the A-connector of the supplied 200-mm 26-way ribbon cable to the **Internal I/O** connector of the 1636 InterPC C2C-Link Adapter
- Plug the B-connector to the **Internal I/O** connector of the target card.

Note: *No power supply connection is required when using the 1636 InterPC C2C-Link Adapter as an HD26F I/O adapter only.*

I/O Toolbox

Set of tools for building-up and processing hardware events.

The **I/O Toolbox** is a configurable array of tools that builds-up streams of event pulses from external signals applied to the input-capable I/O lines.

I/O Toolbox Context

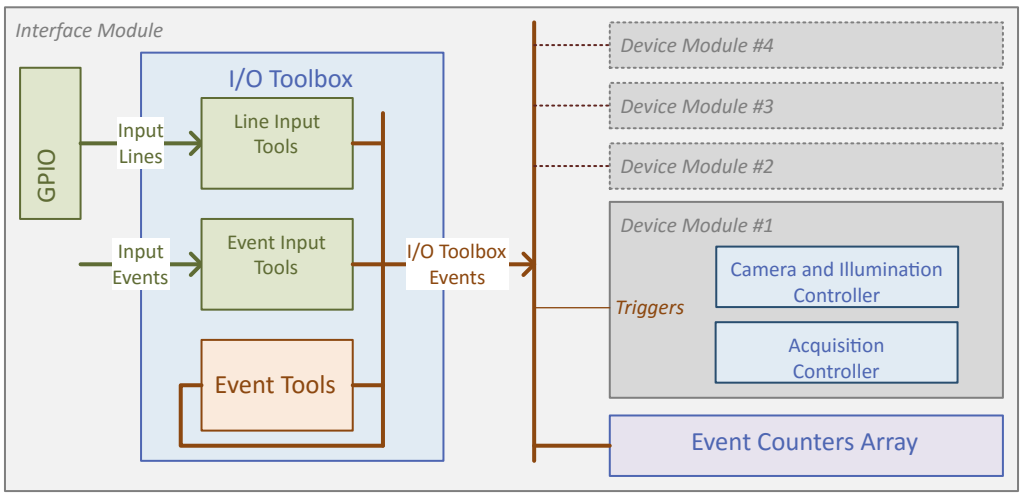


Figure 3: I/O Toolbox context

There is only one I/O Toolbox instance per Coaxlink card. It belongs to the GenTL Interface module.

The input-capable I/O lines and selected event sources are feeding the I/O Toolbox.

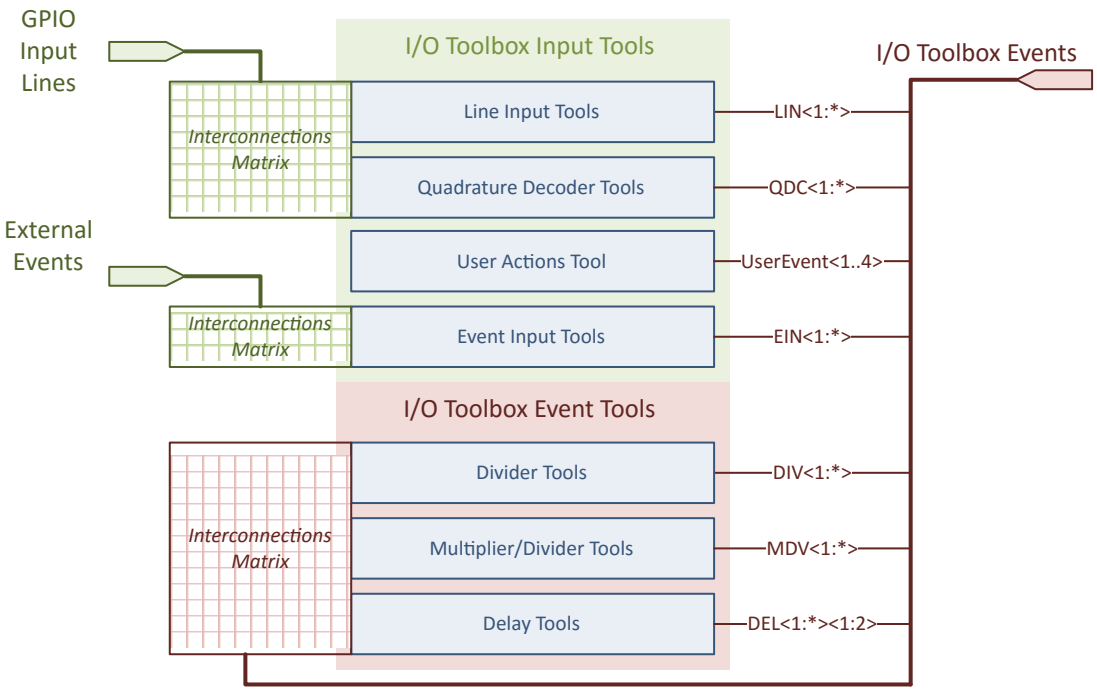
Tools of the I/O Toolbox generates one (or more) event streams. The I/O Toolbox Events streams are distributed to various consumers on the card.

The I/O Toolbox events can be used in every GenTL module of the card for:

- The hardware Cycle Trigger source.
- The hardware Start-of-scan Trigger source.
- The hardware End-of-scan Trigger source.

Every I/O Toolbox event stream is associated with a 32-bit event counter.

I/O Toolbox Structure



I/O Toolbox Input Tools

The I/O Toolbox input tools are fed by the input-capable GPIO lines and a selected set of hardware event sources from outside the I/O Toolbox.

They deliver one I/O Toolbox event stream.

There are four types of input tools:

- The **Line Input Tools**, for use with sensors/detectors using a single GPIO input line.
- The **Quadrature Decoder Tools** for use with quadrature motion encoders using two GPIO input lines.
- The **User Actions Tool** for use by the application software to generate user events
- The **Event Input Tools** for use in line-scan data-forwarding applications.

The **GPIO Input Lines Interconnections Matrix** allows any Line Tool to be fed by any GPIO input line.

The **External Events Interconnections Matrix** allows any Event Input Tool to be fed by the external event source.

Note: *Currently, only the CoaXPress Host interface GPIO message receiver of connector A is available.*

I/O Toolbox Event Tools

The I/O Toolbox Event Tools are fed by one (or more) I/O Toolbox event stream(s) and deliver one (or more) I/O Toolbox event stream (s).

There are three types of events tools:

- The **Divider Tool** generates an event stream by keeping 1 event out of D events of the input stream.
- The **Multiplier/Divider Tool** generates an event stream having M events every D events of the input stream.
- The **Delay Tool** delays the events of one (or two) streams, either by a configurable period of time or by a configurable number of motion encoder ticks.

The **Internal Events Interconnection Matrix** allows any Event Tool to be fed by any I/O Toolbox event stream.

Tools can be cascaded to form a tool chain:

- A tool chain always begins with a Line Tool.
- A (Line or Event) Tool may drive 0, 1 or several Event Tools.

Related Information

Line Input Tool on page 98
Quadrature Decoder Tool on page 100
User Actions Tool on page 104
Event Input Tool on page 109
Divider Tool on page 110
Multiplier/Divider Tool on page 113
Delay Tool on page 118

I/O Toolbox Composition

Composition of the I/O Toolbox vs. product and firmware variant.

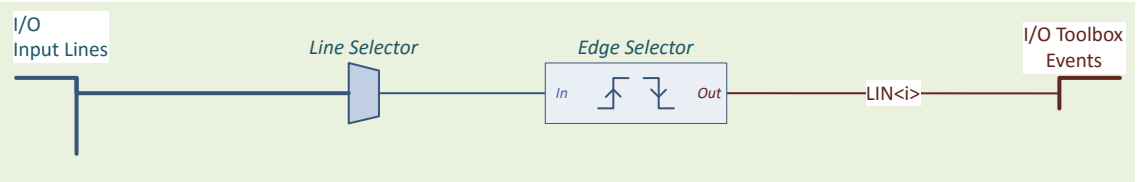
Target Product	Firmware Variant	#LIN	#QDC	#DIV	#MDV	#DEL	#UAS	#EIN
Mono		8	1	1	1	2	1	-
Duo		8	1	1	1	2	1	-
		8	1	1	-	-	1	-
Quad		8	1	1	1	2	1	-
		8	1	1	-	-	1	-
		8	1	1	1	2	1	-
Quad G3		8	1	1	1	2	1	-
		8	2	2	2	2	1	-
		8	-	-	-	-	1	-
		8	1	1	1	2	1	-
		8	2	2	2	2	1	-
Duo PCIe/104		8	1	1	1	2	1	-
		8	1	1	-	-	1	-

Target Product	Firmware Variant	#LIN	#QDC	#DIV	#MDV	#DEL	#UAS	#EIN
		8	1	1	1	2	1	-
Quad G3 DF		8	1	1	1	2	1	-
	DF	8	1	1	1	2	1	-
		8	1	1	1	2	1	-
	DF	8	1	1	1	2	1	2

Line Input Tool

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Line Input Tool	LIN	1 I/O input line	1 event stream: **LIN<i>**

Line Input Tool Wiring Diagram



Any I/O input line can be selected as the input source.
The tool feeds one I/O Toolbox event stream. The output of the tool instance <i> is named LIN<i>.

Line Input Tool Operation

The Line Input tool detects the rising or the falling edge of the selected I/O input line.
The Line Input tool delivers one event at every rising or falling edge according to the `LINActivation` settings.

Line Input Tool GenICam Features

LineInputToolSelector		
Interface	Root > I/O Toolbox	Enumeration (RW)

LineInputToolSelector

Selects a Line Input Tool.

Possible values:

- `LIN1`: Line Input Tool 1. **Default value.**
- `LIN2`: Line Input Tool 2.
- `LIN3`: Line Input Tool 3.
- `LIN4`: Line Input Tool 4.
- `LIN5`: Line Input Tool 5.
- `LIN6`: Line Input Tool 6.
- `LIN7`: Line Input Tool 7.
- `LIN8`: Line Input Tool 8.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

LineInputToolSource

Interface	Root > I/O Toolbox	Enumeration (RW)

Physical GPIO line used as input for the selected Line Input Tool.

Possible values:

- `DIN11`: Differential input 1 of Internal I/O connector 1. **Default value.**
- `DIN12`: Differential input 2 of Internal I/O connector 1.
- `DIN21`: Differential input 1 of Internal I/O connector 2.
- `DIN22`: Differential input 2 of Internal I/O connector 2.
- `IIN11`: Isolated input 1 of Internal I/O connector 1.
- `IIN12`: Isolated input 2 of Internal I/O connector 1.
- `IIN13`: Isolated input 3 of Internal I/O connector 1.
- `IIN14`: Isolated input 4 of Internal I/O connector 1.
- `IIN21`: Isolated input 1 of Internal I/O connector 2.
- `IIN22`: Isolated input 2 of Internal I/O connector 2.
- `IIN23`: Isolated input 3 of Internal I/O connector 2.
- `IIN24`: Isolated input 4 of Internal I/O connector 2.
- `IOUT11`: Isolated output 1 of Internal I/O connector 1.
- `IOUT12`: Isolated output 2 of Internal I/O connector 1.
- `IOUT21`: Isolated output 1 of Internal I/O connector 2.
- `IOUT22`: Isolated output 2 of Internal I/O connector 2.
- `TTLIO11`: TTL input/output 1 of Internal I/O connector 1.
- `TTLIO12`: TTL input/output 2 of Internal I/O connector 1.
- `TTLIO21`: TTL input/output 1 of Internal I/O connector 2.
- `TTLIO22`: TTL input/output 2 of Internal I/O connector 2.

Note: *Check the Available I/O Lines on page 122 for applicable values.*

LineInputToolActivation

Interface	Root > I/O Toolbox	Enumeration (RW)

LineInputToolActivation
Edge activating the output of the selected Line Input Tool. **Possible values:** • `RisingEdge`: Rising edge. • `FallingEdge`: Falling edge.

Quadrature Decoder Tool

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Quadrature Decoder Tool	QDC	2 paired I/O input lines	1 event stream: **QDC<i>** 1 status bit: **QDC<i>Direction** 1 status word: **QDC<i>Position**

Quadrature Decoder Tool Wiring Diagram

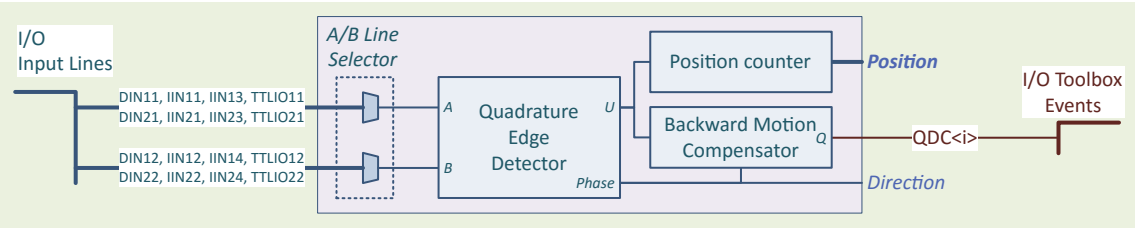
The following pairs of I/O input lines can be selected as the input source:

- For differential input lines: DIN11-DIN12 and DIN21-DIN22
- For TTL input/output lines: TTLIO11-TTLIO12 and TTLIO21-TTLIO22
- For isolated input lines: IIN11-IIN12, IIN13-IIN14, IIN21-IIN22, and IIN23-IIN24

Note: *For the availability, check the Available I/O Lines on page 122.*

The tool feeds one I/O Toolbox event stream. The output of the tool instance <i> is named QDC<i>.
The tool provides also:

- A direction status bit indicating the direction of the motion
- A position status word indicating the position offset



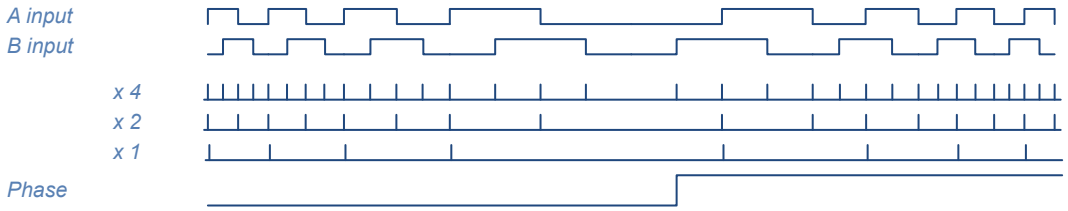
Quadrature Decoder Tool Operation

The Quadrature Decoder Tool decodes the A/B outputs of a **phase-quadrature motion encoder** device.
It delivers 1, 2, or 4 events every A/B cycle, possibly filtered by the backward motion compensator.
The tool includes the following function blocks:

- A quadrature edge detector
- A backward motion compensator
- A 32-bit position counter

The **quadrature edge detector** analyzes the transitions on the A/B lines. It delivers:

- An event stream, named U, having 1, 2, or 4 events every A/B cycle according the `QDCActivation` settings
- An identification of the phase between A and B (A leads B or vice-versa)



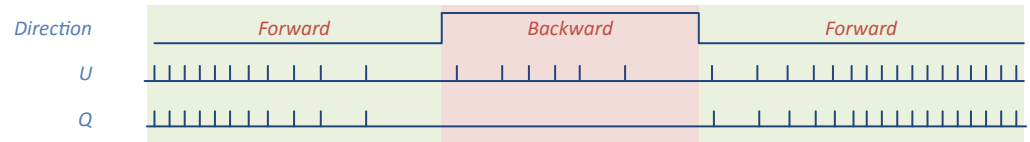
The U stream may be filtered by the backward motion compensator before being delivered to the QDC<i> output.

The phase indication may be inverted according to the `QDCForwardDirection` settings before being delivered to the QDC<i>Direction output.

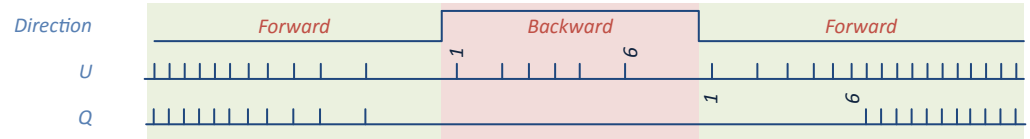
The **backward motion compensator** filters the U stream according to the `QDCOutputMode` setting.

When set to `Unfiltered`, all the events of the U stream are delivered to the QDC<i> output:

When set to `ForwardOnly`, only the events corresponding to the forward direction are delivered to the QDC<i> output:



When set to `FirstPassForwardOnly`, only the events corresponding to the first pass in the forward direction are delivered to the QDC<i> output:



The **position counter** increments by 1 for any U event corresponding to the forward direction and decrements by 1 for the backward direction.

The counter can be reset using the `QuadratureDecoderToolPositionReset` command.

Quadrature Decoder Tool GenICam Features

QuadratureDecoderToolSelector		
Interface	Root > I/O Toolbox	Enumeration (RW)

Selects a Quadrature Decoder Tool.

Possible values:

- `QDC1`: Quadrature Decoder Tool 1. **Default value.**
- `QDC2`: Quadrature Decoder Tool 2.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

QuadratureDecoderToolSources

Interface	Root > I/O Toolbox	Enumeration (RW)

Pair of physical GPIO lines used as A/B inputs for the selected Quadrature Decoder Tool.

Possible values:

- `DIN11_12`: Differential inputs 1 & 2 of Internal I/O connector 1. **Default value.**
- `DIN21_22`: Differential inputs 1 & 2 of Internal I/O connector 2.
- `IIN11_12`: Isolated inputs 1 & 2 of Internal I/O connector 1.
- `IIN13_14`: Isolated inputs 3 & 4 of Internal I/O connector 1.
- `IIN21_22`: Isolated inputs 1 & 2 of Internal I/O connector 2.
- `IIN23_24`: Isolated inputs 3 & 4 of Internal I/O connector 2.
- `TTLIO11_12`: TTL inputs/outputs 1 & 2of Internal I/O connector 1.
- `TTLIO21_22`: TTL inputs/outputs 1 & 2 of Internal I/O connector 2.

Note: *Check the Available I/O Lines on page 122 for applicable values.*

QuadratureDecoderToolActivation

Interface	Root > I/O Toolbox	Enumeration (RW)

Edge activating the output of the selected Quadrature Decoder Tool.

Possible values:

- `RisingEdgeA`: Rising edge of the A signal. One event per encoder cycle.
- `FallingEdgeA`: Falling edge of A signal. One event per encoder cycle.
- `AllEdgesA`: Rising and falling edges of A signal. Two events per encoder cycle.
- `AllEdgesAB`: Rising and falling edges of A and B signals. Four events per encoder cycle.
- `None`: The event is not activated.

Note: *The A output of the quadrature encoder device connects to the first physical GPIO line of the designated pair, e.g. DIN11.*

Note: *The B output of the quadrature encoder device connects to the second physical GPIO line of the designated pair, e.g. DIN12.*

QuadratureDecoderToolForwardDirection

Interface	Root > I/O Toolbox	Enumeration (RW)

A/B phase relationship corresponding to the forward direction of the selected Quadrature Decoder Tool.

Possible values:

- `A_Leads_B`: On forward motion, the A signal leads the B signal
- `B_Leads_A`: On forward motion, the B signal leads the A signal.

QuadratureDecoderToolOuputMode

Interface	Root > I/O Toolbox	Enumeration (RW)

QuadratureDecoderToolOuputMode

Output mode of the selected Quadrature Decoder Tool.

Possible values:

- `Unfiltered`: All the quadrature decoder output events are delivered. **Default value.**
- `ForwardOnly`: Only the events corresponding to the forward direction are delivered.
- `FirstPassForwardOnly`: Only the events corresponding to the first pass in the forward direction are delivered.

QuadratureDecoderToolPosition

Interface	Root > I/O Toolbox	Integer (RO)

Position counter value of the selected Quadrature Decoder Tool.

The position counter is a 32-bit up/down counter that increments by 1 for any event corresponding to the forward direction and decrements by 1 for the backward direction.

Unit: encoder events as defined by `QuadratureDecoderToolActivation`.

Value range: from -2,147,483,648 up to 2,147,483,647.

QuadratureDecoderToolDirection

Interface	Root > I/O Toolbox	Enumeration (RO)

Current direction of the selected Quadrature Decoder Tool

Possible values:

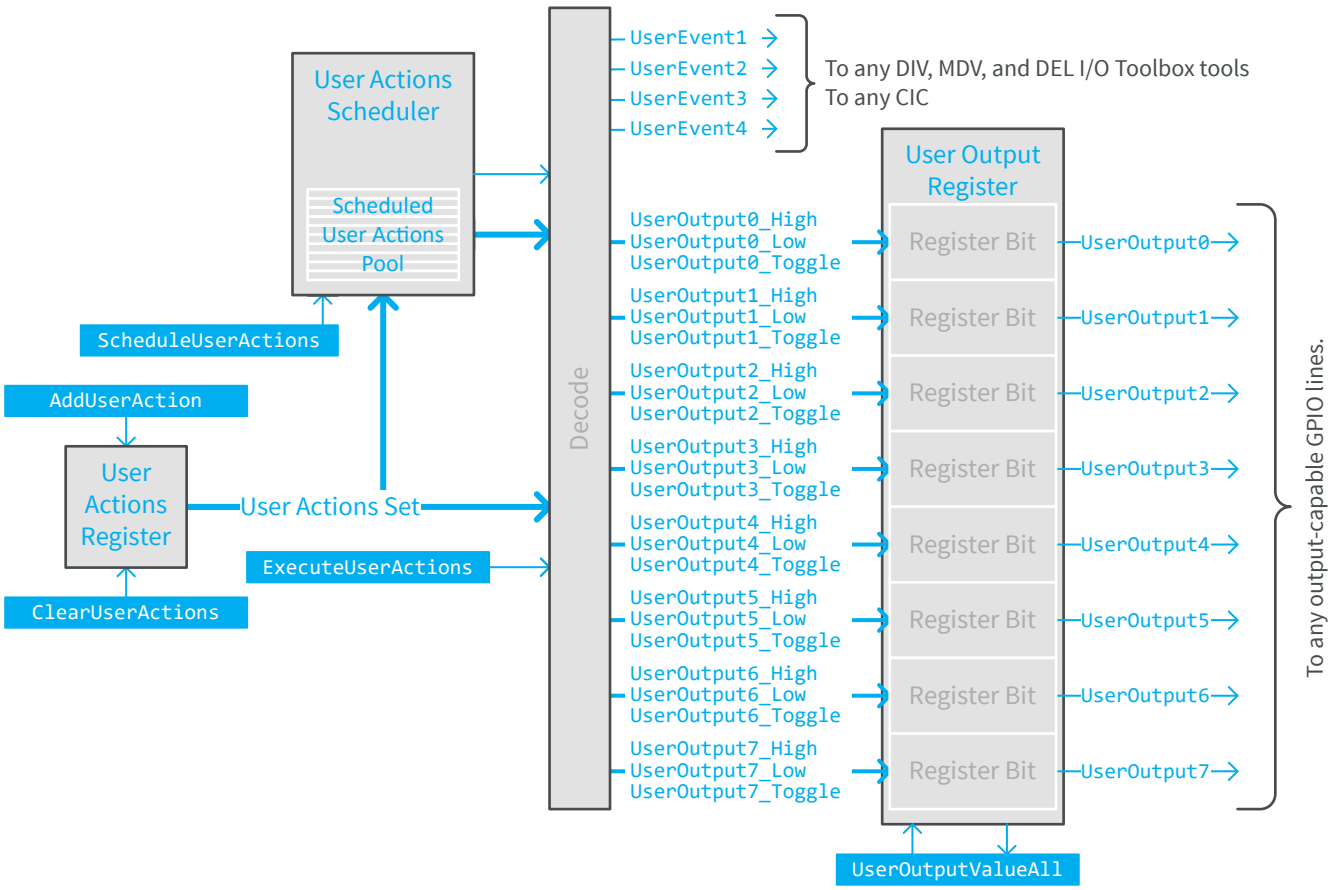
- `Forward`: Forward motion direction.
- `Backward`: Backward motion direction.

QuadratureDecoderToolPositionReset

Interface	Root > I/O Toolbox	Command (WO)

Reset Position counter of the selected Quadrature Decoder Tool.

User Actions Tool



All the firmware variants of all Coaxlink products implement a hardware function block named **User Actions Tool** that allows the application software to perform the following **User Actions**:

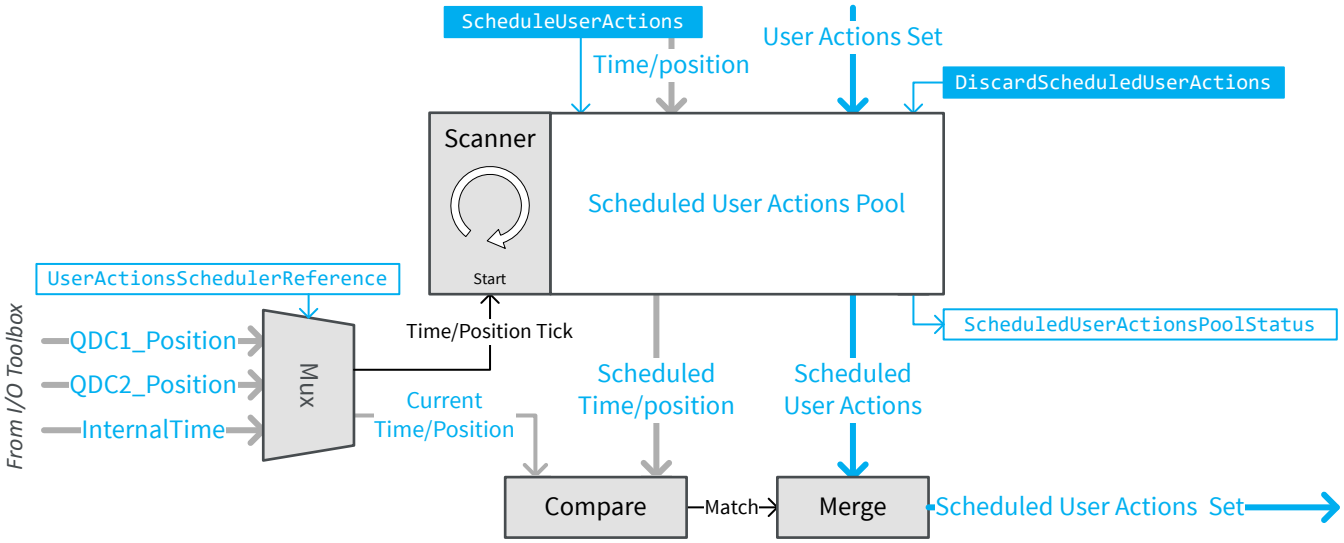
- Generate events on any User Events source.
- Setting high any bit of the User Output Register.
- Setting low any bit of the User Output Register
- Toggling any bit of the User Output Register.

With the `ClearUserActions` and `AddUserActions` GenICam features, the user application can define a **User Actions Set** composed with all the actions to be executed simultaneously.

After having defined the User Actions Set, the application software has two options for its execution:

- Using the `ExecuteUserActions` GenICam feature to execute immediately all the actions defined in the **User Actions Set**.
- Using the `ScheduleUserActions` GenICam feature to delegate the execution of the User Actions Set to the **User Actions Scheduler**.

User Actions Scheduler



The UAS function block allows an application software to postpone the execution of the actions.

Scheduler Reference Settings

Prior to using the User Actions Scheduler, the user application has to select a 32-bit reference counter. This is achieved by assigning one of the following three values to the `UserActionsSchedulerReference` GenICam feature:

- `InternalTime` selects the **Coaxlink card local time**: A monotonic time base that increments by 1 every 1 microsecond and wraps around after about 71 minutes when it reaches the maximum value of 4,294,967,295.
- `QDC1Position` and `QDC2Position` selects the **Position Counter** of the Quadrature Decoder tools QDC1 and QDC2 respectively.

Important: *For correct operation of the UAS with a position reference, both conditions must be satisfied:*

- *The position counter must increment monotonically!*
- *The position counter may not increment faster than every microsecond.*

Note: *To ensure monotonic increments of the QDC position counter:*

- *Set properly `QuadratureDecoderToolForwardDirection` such that the counter increments when the object moves in the forward direction.*
- *If it exists any backward motion, prevent the position counter to decrement by setting the `QuadratureDecoderToolOuputMode` to `ForwardOnly` or to `FirstPassForwardOnly`.*

Schedule User Actions Pool Operation

The Scheduled User Actions Pool is a memory area where the Scheduled User Actions Sets are stored by the user application. The pool has 64 locations.

To add a new Scheduled User Actions to the Pool, the user application must:

1. Ensure that there is at least one free location by getting the value of the `ScheduledUserActionsPoolStatus` GenICam feature,
2. Define a User Actions Set,
3. Determines the time/position 32-bit value when the actions are to be executed,
4. Set this value to `ScheduledUserActions` GenICam feature.

Scheduled User Actions are removed from the pool when they are executed.

The pool can be cleared at any time by executing the `DiscardScheduledUserActions` GenICam command.

Scanner Operation and Scheduled Actions Execution

At every increment of the 32-bit (time or position) reference counter, the Scanner reads all the locations and compares the scheduled reference time/position with the current time/position count value.

When the values are identical, the Scheduled User Actions Set is elected for execution at the end of the scan and removed from the pool.

When multiple sets are elected for execution, their actions are merged.

> ***Note:*** *Merging a set low and a set high action on the same User Output Register bit results into a toggle action.*

At the end of the scan: the merged elected actions are executed simultaneously. The time delay from the reference tick up to the execution of the elected actions is very small (sub-microsecond) and constant.

User Events

User-driven event sources.

All the firmware variants of all Coaxlink products implement 4 User Events sources named `UserEvent1` … `UserEvent4`.

Any User Event can be used:

- As Cycle Trigger source by all the Camera and Illumination Controllers.
- As Cycle Sequence Trigger source by all the Camera and Illumination Controllers.
- As event source by all the Divider tools of the I/O Toolbox.
- As event source by all the Multiplier/Divider tools of the I/O Toolbox
- As event source for both channels of all the Delay) tools of the I/O Toolbox.

User Output Register

User-configurable general purpose output register.

All the firmware variants of all Coaxlink products but 1630 Coaxlink Mono implement an 8-bit User Output Register where bits are named `UserOutput0` … `UserOutput7`.

1630 Coaxlink Mono implements only the 4 lowest bits!

Any User Output Register bit can be used as signal source by any output-capable GPIO line.

The user application has two options to define the state of the User Output Register bits:

- The User Actions option allows to change the state of each bit individually.
- Setting a value to the `UserOutputValueAll` to define the state of all bits.

Getting the value of `UserOutputValueAll` allows the user application to get the state of all bits.

User Actions GenICam Features

UserOutputValueAll		
Interface	Root > User Output Register	Integer (RW)

UserOutputValueAll

Value of all User Output Register bits as a bit-field in a single data word.

Bit values:

- `0`: The value of the corresponding User Output Register bit is low.
- `1`: The value of the corresponding User Output Register bit is high.

Bit assignments:

- Bit 0: Value for bit 0 of the user output register.
- Bit 1: Value for bit 1 of the user output register.
- Bit 2: Value for bit 2 of the user output register.
- Bit 3: Value for bit 3 of the user output register.
- Bit 4: Value for bit 4 of the user output register.
- Bit 5: Value for bit 5 of the user output register.
- Bit 6: Value for bit 6 of the user output register.
- Bit 7: Value for bit 7 of the user output register.

Note: *Check Available User Output Register Bits for applicable values.*

UserActions

Interface	Root > User Output Register	Integer (WO)

Current set of user actions (built with `AddUserAction`) that can be executed immediately (c.f. `ExecuteUserActions`) or scheduled for execution at a specific time/position (c.f. `ScheduleUserActions`).

Value range: from 0 up to 2,147,483,647

AddUserAction

Interface	Root > User Output Register	Enumeration (WO)

AddUserAction

Add an action to the current set of user actions (`UserActions`)

Possible values:

- `UserEvent1`: User Event 1.
- `UserEvent2`: User Event 2.
- `UserEvent3`: User Event 3.
- `UserEvent4`: User Event 4.
- `UserOutput0_High`: Set User Output Register bit 0 high.
- `UserOutput0_Low`: Set User Output Register bit 0 low.
- `UserOutput0_Toggle`: Toggle User Output Register bit 0.
- `UserOutput1_High`: Set User Output Register bit 1 high.
- `UserOutput1_Low`: Set User Output Register bit 1 low.
- `UserOutput1_Toggle`: Toggle User Output Register bit 1.
- `UserOutput2_High`: Set User Output Register bit 2 high.
- `UserOutput2_Low`: Set User Output Register bit 2 low.
- `UserOutput2_Toggle`: Toggle User Output Register bit 2.
- `UserOutput3_High`: Set User Output Register bit 3 high.
- `UserOutput3_Low`: Set User Output Register bit 3 low.
- `UserOutput3_Toggle`: Toggle User Output Register bit 3.
- `UserOutput4_High`: Set User Output Register bit 4 high.
- `UserOutput4_Low`: Set User Output Register bit 4 low.
- `UserOutput4_Toggle`: Toggle User Output Register bit 4.
- `UserOutput5_High`: Set User Output Register bit 5 high.
- `UserOutput5_Low`: Set User Output Register bit 5 low.
- `UserOutput5_Toggle`: Toggle User Output Register bit 5.
- `UserOutput6_High`: Set User Output Register bit 6 high.
- `UserOutput6_Low`: Set User Output Register bit 6 low.
- `UserOutput6_Toggle`: Toggle User Output Register bit 6.
- `UserOutput7_High`: Set User Output Register bit 7 high.
- `UserOutput7_Low`: Set User Output Register bit 7 low.
- `UserOutput7_Toggle`: Toggle User Output Register bit 7.

ClearUserActions

Interface	Root > User Output Register	Command (WO)

Clear the current set of user actions (`UserActions`).

ExecuteUserActions

Interface	Root > User Output Register	Command (WO)

Immediately execute the current set of user actions (`UserActions`).

ScheduleUserActions

Interface	Root > User Output Register	Integer (RW)

ScheduleUserActions

Schedule the current set of user actions (`UserActions`) for execution at given time/position.

Value range: from 0 up to 4,294,967,295

UserActionsSchedulerReference

Interface	Root > User Output Register	Enumeration (RW)

Defines the reference used by the user action scheduler; can only be changed when `ScheduledUserActionsPoolStatus` is `Empty`.

Possible values:

- `InternalTime`: Coaxlink card local time.
- `QDC1Position`: Quadrature Decoder Tool 1 Position.
- `QDC2Position`: Quadrature Decoder Tool 2 Position.

ScheduledUserActionsPoolStatus

Interface	Root > User Output Register	Enumeration (RW)

Reports the status of the pool of scheduled user actions.

Possible values:

- `Empty`: The pool of scheduled user actions is empty.
- `PartiallyFilled`: The pool of scheduled user actions is partially filled.
- `AlmostFull`: The pool of scheduled user actions almost full.

DiscardScheduledUserActions

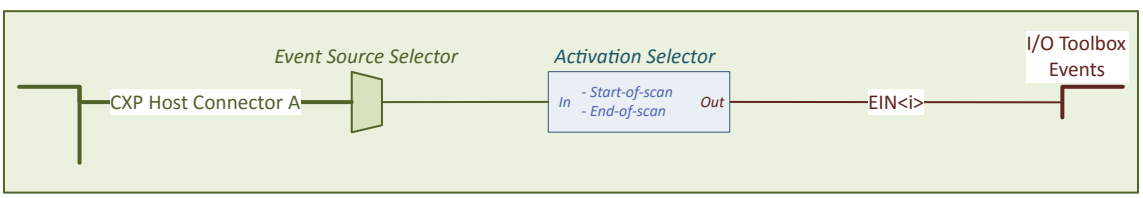
Interface	Root > User Output Register	Command (WO)

Discard all scheduled user actions.

Event Input Tool

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Event Input Tool	EIN	1 Event	1 event stream: **EIN<i>**

Event Input Tool Wiring Diagram



Connector A of the CoaXPress Host Interface is the only source of events that can be selected.
The tool feeds one I/O Toolbox event stream. The output of the tool instance <i> is named EIN<i>.

Event Input Tool Operation

The EIN Input tool decodes the custom GPIO message received on the selected source.

The Event Input tool delivers one event on the reception of a "Start-of-scan message" or an "End-of-scan message" according to the `EINActivation` settings.

Event Input Tool GenICam Features

EventInputToolSelector		
Interface	Root > I/O Toolbox	Enumeration (RW)

Selects a Line Input Tool.

Possible values:

- `EIN1`: Event Input Tool 1. **Default value.**
- `EIN2`: Event Input Tool 2.

EventInputToolSource		
Interface	Root > I/O Toolbox	Enumeration (RW)

CoaXPress connection used as input for the selected Event Input Tool.

Possible values:

- `A`: Connector A of the CoaXPress Host Interface. **Default value.**

EventInputToolActivation		
Interface	Root > I/O Toolbox	Enumeration (RW)

Message activating the output of the selected Event Input Tool.

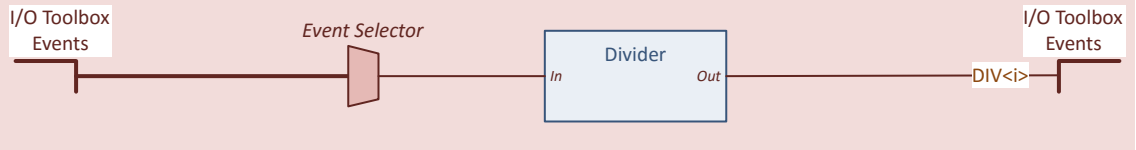
Possible values:

- `StartOfScan`: Receipt of a start of scan message.
- `EndOFScan`: Receipt of an end of scan message.

Divider Tool

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Divider Tool	DIV	1 Toolbox event stream	1 event stream: **DIV<i>**

Divider Tool Wiring Diagram



Any I/O Toolbox event stream can be selected as the input source.

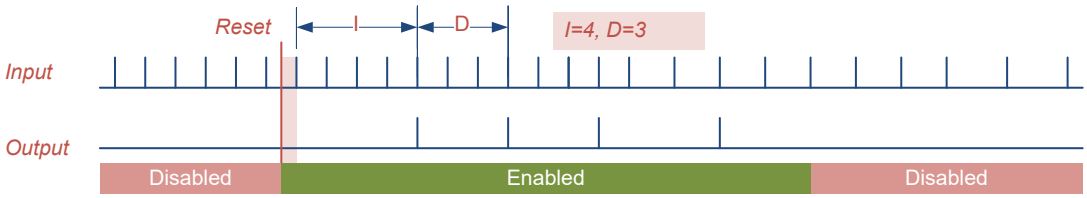
The tool feeds one I/O Toolbox event stream. The output of the tool instance <i> is named DIV<i>.

Divider Tool Operation

The Divider tool divides the input rate by an integer division factor ranging from 1 to 1023.

The Divider tool can be in one of the following two states: the *Disabled* state and the *Enabled* state:

- In the *Enabled* state, the Divider tool delivers the first event after having skipped **I** input events and the subsequent events every **D** input events.
- In the *Disabled* state, the Divider tool stops the events delivery.



The I value, a.k.a. **Divider Tool Initial Offset**, is configurable in the range 0 to 65535.

The D value, a.k.a. **Divider Tool Division Factor**, is configurable in the range 1 to 65535.

The state of the Divider tool is controlled by the user application software.

Divider Tool GenICam Features

DividerToolSelector		
Interface	Root > I/O Toolbox	Enumeration (RW)

Selects a Divider Tool.

Possible values:

- `DIV1`: Divider Tool 1. **Default value.**
- `DIV2`: Divider Tool 2.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

DividerToolSource		
Interface	Root > I/O Toolbox	Enumeration (RW)

DividerToolSource

I/O Toolbox event stream used as input for the selected Divider Tool.

Possible values:

- `NONE`: *Nothing connected.* **Default value.**
- `LIN1`: Line Input Tool 1.
- `LIN2`: Line Input Tool 2.
- `LIN3`: Line Input Tool 3.
- `LIN4`: Line Input Tool 4.
- `LIN5`: Line Input Tool 5.
- `LIN6`: Line Input Tool 6.
- `LIN7`: Line Input Tool 7.
- `LIN8`: Line Input Tool 8.
- `QDC1`: Quadrature Decoder Tool 1
- `QDC2`: Quadrature Decoder Tool 2.
- `DIV1`: Divider Tool 1.
- `DIV2`: Divider Tool 2.
- `MDV1`: Multiplier/Divider Tool 1.
- `MDV2`: Multiplier/Divider Tool 2.
- `DEL11`: Delay Tool 1 Output 1.
- `DEL12`: Delay Tool 1 Output 2.
- `DEL21`: Delay Tool 2 Output 1.
- `DEL12`: Delay Tool 2 Output 2.
- `EIN1`: Event Input Tool 1.
- `EIN2`: Event Input Tool 2.
- `UserEvent1`: User Event 1.
- `UserEvent2`: User Event 2.
- `UserEvent3`: User Event 3.
- `UserEvent4`: User Event 4.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

DividerToolEnableControl

Interface	Root > I/O Toolbox	Enumeration (RW)

Enable control of the selected Divider Tool.

Possible values:

- `Enable`: The output is enabled. **Default value.**
- `Disable`: The output is disabled.

DividerToolDivisionFactor

Interface	Root > I/O Toolbox	Integer (RW)

DividerToolDivisionFactor

Division factor of the selected Divider Tool.

This feature is the **denominator** of the fraction defining the Rate Conversion Ratio (RCR) of the Divider Tool.

RCR = 1/D where:

- D = `DividerToolDivisionFactor`

Value range: from 1 to 65,535.

Default value: 2.

DividerToolInitialOffset

Interface	Root > I/O Toolbox	Integer (RW)

Initial offset of the selected Divider Tool.

This feature defines the number of skipped input events after enabling the Divider tool.

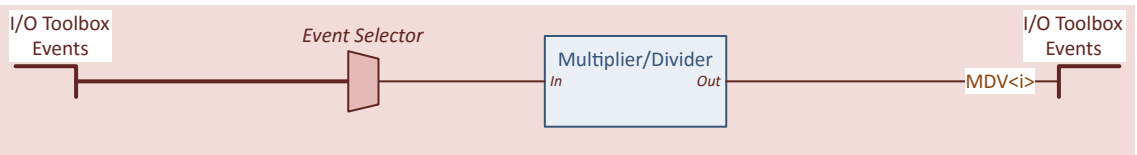
Value range: from 0 to 65,535.

Default value: 0.

Multiplier/Divider Tool

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Multiplier/Divider Tool	MDV	1 Toolbox event stream	1 Toolbox event stream: **MDV<i>**

Multiplier/Divider Tool Wiring Diagram



Any I/O Toolbox event stream can be selected as the input source.

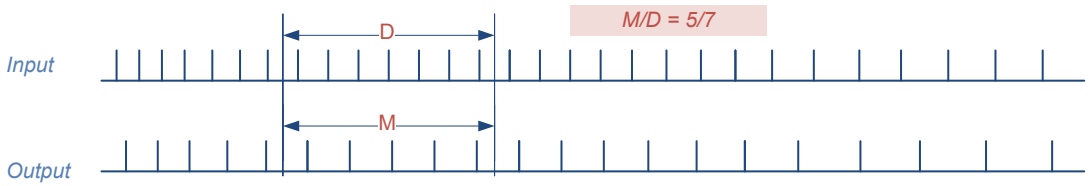
The tool feeds one I/O Toolbox event stream. The output of the tool instance <i> is named MDV<i>.

Multiplier/Divider Tool Operation

The Multiplier/Divider tool multiplies and/or divides the input rate by any rate conversion ratio – RCR – value in the range 0.001 to 1000.0.

The Multiplier/Divider tool measures the time interval between every consecutive input events and adapts the output rate accordingly.

The Multiplier/Divider is **frequency accurate**. The output frequency is strictly proportional to the input frequency provided that the input frequency is stable (or varies slowly). In such conditions, the Multiplier/Divider delivers M events for every D input events.



The Rate Conversion Ratio is configured as the ratio of two float numbers:

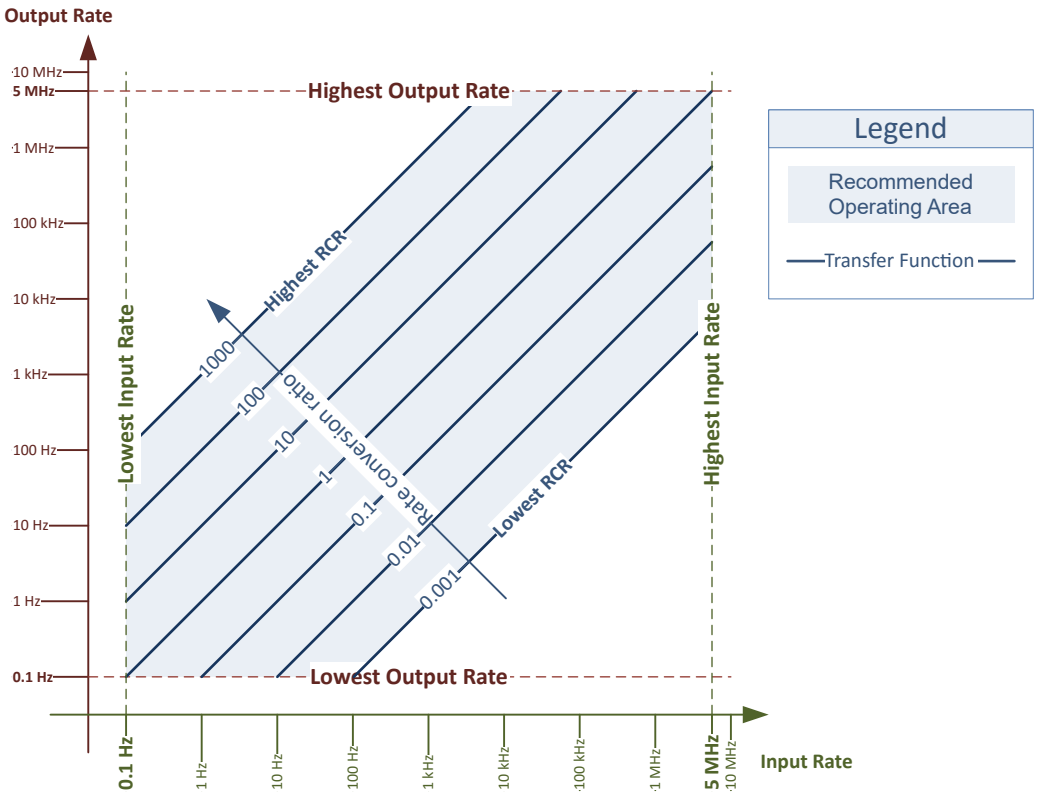
- The M value, a.k.a. **Multiplier Divider Tool Multiplication factor**, is configurable in the range 0.001 to 1000.0.
- The D value, a.k.a. **Multiplier Divider Tool Division Factor**, is configurable in the range 0.001 to 1000.0.

Note: *The effective Rate Conversion Ratio may slightly differ from the specified RCR. However, the RCR relative error remains negligible (less than 1/1000). The effective RCR is reported to the application.*

Note: *Frequency variations of the input event stream are reported to the output event stream with a latency of 1 period of the input event stream. Such a latency induces some phase errors in the output event stream. The phase of the output event stream accumulates retards when the input frequency increases and vice-versa when the input frequency increases. Consequently, the Multiplier/Divider is* ***not phase accurate****.*

Multiplier/Divider Tool Operating Limits

Characteristic	Symbol	Min	Max
Rate Conversion Ratio	RCR	0.001	1000
Input Rate	f_{IN}	0.1 Hz	5 MHz
Output Rate	f_{OUT}	0.1 Hz	5 MHz



Multiplier/Divider Tool GenICam Features

MultiplierDividerToolSelector

Interface	Root > I/O Toolbox	Enumeration (RW)

Selects a Multiplier/Divider Tool.

Possible values:

- `MDV1`: Multiplier/Divider Tool 1. **Default value.**
- `MDV2`: Multiplier/Divider Tool 2.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

MultiplierDividerToolSource

Interface	Root > I/O Toolbox	Enumeration (RW)

I/O Toolbox event stream used as input for the selected Multiplier/Divider Tool.

Possible values:

- `NONE`: *Nothing connected.* **Default value.**
- `LIN1`: Line Input Tool 1.
- `LIN2`: Line Input Tool 2.
- `LIN3`: Line Input Tool 3.
- `LIN4`: Line Input Tool 4.
- `LIN5`: Line Input Tool 5.
- `LIN6`: Line Input Tool 6.
- `LIN7`: Line Input Tool 7.
- `LIN8`: Line Input Tool 8.
- `QDC1`: Quadrature Decoder Tool 1.
- `QDC2`: Quadrature Decoder Tool 2.
- `DIV1`: Divider Tool 1.
- `DIV2`: Divider Tool 2.
- `MDV1`: Multiplier/Divider Tool 1.
- `MDV2`: Multiplier/Divider Tool 2.
- `DEL11`: Delay Tool 1 Output 1.
- `DEL12`: Delay Tool 1 Output 2.
- `DEL21`: Delay Tool 2 Output 1.
- `DEL12`: Delay Tool 2 Output 2.
- `EIN1`: Event Input Tool 1.
- `EIN2`: Event Input Tool 2.
- `UserEvent1`: User Event 1.
- `UserEvent2`: User Event 2.
- `UserEvent3`: User Event 3.
- `UserEvent4`: User Event 4.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

MultiplierDividerToolOutputControl

Interface	Root > I/O Toolbox	Enumeration (RW)

Output control of the selected Multiplier/Divider Tool.

Possible values:

- `Enable`: Output enabled.
- `Disable`: Output disabled.

MultiplierDividerToolMultiplicationFactor

Interface	Root > I/O Toolbox	Float (RW)

Multiplication factor of the selected Multiplier/Divider Tool.

This feature is the **numerator** of the fraction defining the Rate Conversion Ratio (RCR) of the Multiplier/Divider Tool.

RCR = M/D where:

- M = `MultiplierDividerToolMultiplicationFactor`
- D = `MultiplierDividerToolDivisionFactor`

The Multiplier/Divider Tools allows defining any RCR values in the range 0.001 to 1000.0.

Note: *The user may define RCR using any of the following methods:*

- *A ratio of 2 integer numbers by assigning integer values to both the numerator and the denominator.*
- *A single non-integer number greater or smaller than 1 assigned to the numerator leaving the denominator to the default value (1.0).*
- *A single non-integer number greater or smaller than 1 assigned to the denominator leaving the numerator to the default value (1.0).*

Value range: from 0.001 to 1,000.0.

Default value: 1.0.

MultiplierDividerToolDivisionFactor

Interface	Root > I/O Toolbox	Float (RW)

MultiplierDividerToolDivisionFactor

Division factor of the selected Multiplier/Divider Tool.

This feature is the **denominator** of the fraction defining the Rate Conversion Ratio (RCR) of the Multiplier/Divider Tool.

RCR = M/D where:

- M = `MultiplierDividerToolMultiplicationFactor`
- D = `MultiplierDividerToolDivisionFactor`

The Multiplier/Divider Tools allows defining any RCR values in the range 0.001 to 1000.0.

Note: *The user may define RCR using any of the following methods:*

- *A ratio of 2 integer numbers by assigning integer values to both the numerator and the denominator.*
- *A single non-integer number greater or smaller than 1 assigned to the numerator leaving the denominator to the default value (1.0).*
- *A single non-integer number greater or smaller than 1 assigned to the denominator leaving the numerator to the default value (1.0).*

Value range: from 0.001 to 1,000.0.

Default value: 1.0.

MultiplierDividerToolEffectiveMultiplicationFactor

Interface	Root > I/O Toolbox	Integer (RO)

Effective multiplication factor of the selected Multiplier/Divider Tool.

This feature is the **numerator** of the fraction defining the Effective Rate Conversion Ratio (Effective RCR) of the Multiplier/Divider Tool.
Effective RCR = Effective M/Effective D where:

- Effective M =`MultiplierDividerToolEffectiveMultiplicationFactor`
- Effective D = `MultiplierDividerToolEffectiveDivisionFactor`

MultiplierDividerToolEffectiveDivisionFactor

Interface	Root > I/O Toolbox	Integer (RO)

Effective division factor of the selected Multiplier/Divider Tool.

This feature is the **denominator** of the fraction defining the Effective Rate Conversion Ratio (Effective RCR) of the Multiplier/Divider Tool.

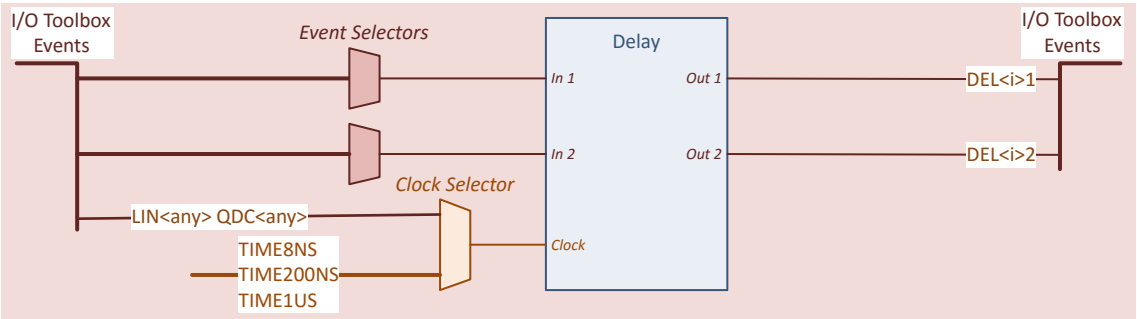
Effective RCR = Effective M/Effective D where:

- Effective M =`MultiplierDividerToolEffectiveMultiplicationFactor`
- Effective D = `MultiplierDividerToolEffectiveDivisionFactor`

Delay Tool

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Delay Tool	DEL	2 Toolbox event streams 1 clock signal	2 Toolbox event stream: **DEL<i>1**, **DEL<i>2**

Delay Tool Wiring Diagram



Any I/O Toolbox event stream can be selected as the input 1 source.

Any I/O Toolbox event stream can be selected as the input 2 source.

The tool feeds two I/O Toolbox event streams. The outputs of the tool instance <i> are named DEL<i>1 and DEL<i>2.

Delay Tool Operation

The event streams applied on either inputs (In1 and In2) are replicated on the corresponding output (Out1 and Out2) after a configurable number of clock tick events.

The delay range is: 0 up to 16,777,215 clock ticks; the same delay applies to both channels.

The clock source is selectable: it can be a time base or a line tool event stream.

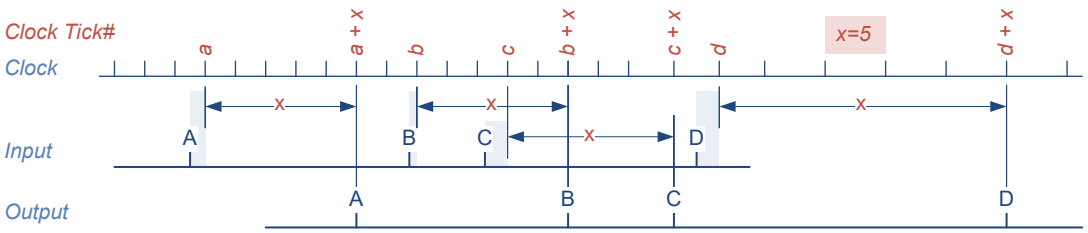
Selecting a time base implements a **time delay** function. The available time bases are:

- A 125 MHz (8 ns period) high accuracy time base allowing delays up to 134 milliseconds by steps of 8 ns
- A 5 MHz (200 ns period) high accuracy time base allowing delays up to 3.35 seconds by steps of 200 ns
- A 1 MHz (1 µs period) high accuracy time base allowing delays up to 16.7 seconds by steps of 1 µs

Selecting a line tool event stream implements a **position offset** function. The available event stream sources are:

- Any available Line Tool
- Any available Quadrature Decoder Tool

Each channel of the Delay tool: operate as a delay line: it may accept a new event while the previous one is not yet delivered. The Delay tool is capable of recording up to 16 distinct events for both channels.



Delay Tool GenICam Features

DelayToolSelector

Interface	Root > I/O Toolbox	Enumeration (RW)

Selects a Delay Tool.

Possible values:

- `DEL1`: Delay Tool 1. **Default value.**
- `DEL2`: Delay Tool 2.

DelayToolSource1

Interface	Root > I/O Toolbox	Enumeration (RW)

I/O Toolbox event stream used as input 1 for the selected Delay Tool.

Possible values:

- `NONE`: *Nothing connected.* **Default value.**
- `LIN1`: Line Input Tool 1.
- `LIN2`: Line Input Tool 2.
- `LIN3`: Line Input Tool 3.
- `LIN4`: Line Input Tool 4.
- `LIN5`: Line Input Tool 5.
- `LIN6`: Line Input Tool 6.
- `LIN7`: Line Input Tool 7.
- `LIN8`: Line Input Tool 8.
- `QDC1`: Quadrature Decoder Tool 1
- `QDC2`: Quadrature Decoder Tool 2.
- `DIV1`: Divider Tool 1.
- `DIV2`: Divider Tool 2.
- `MDV1`: Multiplier/Divider Tool 1.
- `MDV2`: Multiplier/Divider Tool 2.
- `DEL11`: Delay Tool 1 Output 1.
- `DEL12`: Delay Tool 1 Output 2.
- `DEL21`: Delay Tool 2 Output 1.
- `DEL12`: Delay Tool 2 Output 2.
- `EIN1`: Event Input Tool 1.
- `EIN2`: Event Input Tool 2.
- `UserEvent1`: User Event 1.
- `UserEvent2`: User Event 2.
- `UserEvent3`: User Event 3.
- `UserEvent4`: User Event 4.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

DelayToolSource2

Interface	Root > I/O Toolbox	Enumeration (RW)

I/O Toolbox event stream used as input 2 for the selected Delay Tool.

Possible values:

- `NONE`: *Nothing connected.* **Default value.**
- `LIN1`: Line Input Tool 1.
- `LIN2`: Line Input Tool 2.
- `LIN3`: Line Input Tool 3.
- `LIN4`: Line Input Tool 4.
- `LIN5`: Line Input Tool 5.
- `LIN6`: Line Input Tool 6.
- `LIN7`: Line Input Tool 7.
- `LIN8`: Line Input Tool 8.
- `QDC1`: Quadrature Decoder Tool 1
- `QDC2`: Quadrature Decoder Tool 2.
- `DIV1`: Divider Tool 1.
- `DIV2`: Divider Tool 2.
- `MDV1`: Multiplier/Divider Tool 1.
- `MDV2`: Multiplier/Divider Tool 2.
- `DEL11`: Delay Tool 1 Output 1.
- `DEL12`: Delay Tool 1 Output 2.
- `DEL21`: Delay Tool 2 Output 1.
- `DEL12`: Delay Tool 2 Output 2.
- `EIN1`: Event Input Tool 1.
- `EIN2`: Event Input Tool 2.
- `UserEvent1`: User Event 1.
- `UserEvent2`: User Event 2.
- `UserEvent3`: User Event 3.
- `UserEvent4`: User Event 4.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

DelayToolClockSource

Interface	Root > I/O Toolbox	Enumeration (RW)

DelayToolClockSource

I/O Toolbox event stream used as clock input for the selected Delay Tool.

Possible values:

- `NONE`: *Nothing connected.* **Default value.**
- `TIME8NS`: 8 nanoseconds time base.
- `TIME200NS`: 200 nanoseconds time base.
- `TIME1US`: 1 microsecond time base.
- `LIN1`: Line Input Tool 1.
- `LIN2`: Line Input Tool 2.
- `LIN3`: Line Input Tool 3.
- `LIN4`: Line Input Tool 4.
- `LIN5`: Line Input Tool 5.
- `LIN6`: Line Input Tool 6.
- `LIN7`: Line Input Tool 7.
- `LIN8`: Line Input Tool 8.
- `QDC1`: Quadrature Decoder Tool 1.
- `QDC2`: Quadrature Decoder Tool 2.

Check the *I/O Toolbox Composition* on page 97 for applicable values

DelayToolDelayValue

Interface	Root > I/O Toolbox	Integer (RW)

Delay value of the selected Delay Tool.

Unit: time or event according to `DelayToolClockSource`.

Value range: from 0 up to 16,777,215.

Default value: 0.

General Purpose I/O Lines

Set of general purpose digital I/O lines.

I/O Lines Types

Lines Type	Electrical Style	Frequency range	Name Prefix
Differential input lines	RS-422 differential inputs	Up to 5 MHz	DIN
TTL input/output lines	5 Volt-tolerant LVTTL single-ended input or LVTTL (totem-pole, open-emitter and open-collector) single-ended output	Up to 5 MHz	TTLIO
Isolated input lines	Isolated current-sense input compliant with 5 V, 12 V and, 24 V signaling levels	Up to 50 kHz	IIN
Isolated output lines	Isolated 100 mA/30V contact output	Up to 50 kHz	IOUT

Available I/O Lines

I/O Line Set 1

The I/O Line Set 1 is composed of 10 I/O lines. It is available on:

- All Coaxlink PCIe products: Mono Duo Quad Quad G3 Quad G3 DF
- The 3300 HD26F I/O module for Coaxlink Duo PCIe/104 optional accessory of Duo PCIe/104

I/O Line Type	I/O Line Names	Count
Differential input	`DIN11, DIN12`	2
TTL inputs/output	`TTLIO11, TTLIO12`	2
Isolated input	`IIN11, IIN12, IIN13, IIN14`	4
Isolated output	`IOUT11, IOUT12`	2

I/O Line Set 2

The I/O Line Set 2 is composed of 10 I/O lines.

It is available on a selection of Coaxlink products: Duo Quad Quad G3

I/O Line Type	I/O Line Names	Count
Differential input	`DIN21, DIN22`	2
TTL inputs/output	`TTLIO21, TTLIO22`	2

I/O Line Type	I/O Line Names	Count
Isolated input	IIN21, IIN22, IIN23, IIN24	4
Isolated output	IOUT21, IOUT22	2

I/O Lines Usage

Input Lines

The input-capable I/O lines (DIN, TTLIO, IN) can be used as:

- General purpose input: An input signal whose state can be read or monitored by the host application.
- Motion encoder input: A pair of input signals delivered by a quadrature motion encoder and used for triggering the acquisition from the camera.
- Trigger input: An input signal used to trigger the acquisition from the camera.

Output Lines

The output-capable I/O lines (TTLIO, IOUT) can be used as:

- General purpose output: An output signal whose state can be set by the host application.
- Strobe output: An output signal usually used to control a strobe light, in synchronization with the camera.
- Camera trigger output (only available on TTLIO): An output signal generated by the Coaxlink card and used to trigger the camera.

I/O Control Blocks

Every I/O line is controlled through one I/O control block.

Block Diagrams

There a 3 variations of the control block depending on the input/output capabilities of the I/O line:

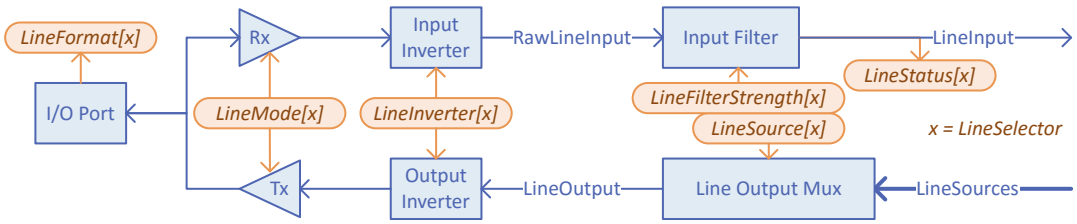


Figure 4: I/O control block for bidirectional I/O lines

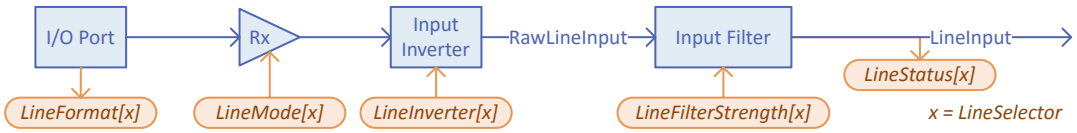


Figure 5: I/O control block for input-only I/O lines

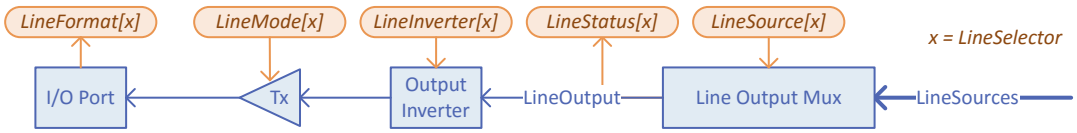


Figure 6: I/O control block for output-only I/O lines

Note: *In the above drawings:*

- *A thin blue line represents one individual electrical signal path*
- *A thick blue line represents a collection of electrical signal paths*
- *The blue arrowhead shows the propagation direction of the electrical signal(s)*
- *A blue shape represents a functional element of the I/O control block*
- *An orange (oval) shape represents a GenICam feature; the text inside being the feature name*
- *An [x] appended to the feature name indicates that the feature is associated with a selector feature (in this case: LineSelector)*
- *The orange arrowhead indicates the access-mode of the feature: read-only features having incoming arrows, writable features having an outgoing arrow.*

Input Path

Input-only and bidirectional I/O lines share a common input path structure including:

- The I/O port block representing the I/O pins on the I/O connector(s)
- The Rx block representing the line receiver circuit
- The Input Inverter block representing the user-configurable logic inverter
- The Input Filter block representing the user-configurable glitch-removal filter

Output Path

Output-only and bidirectional I/O lines share a common output path structure including:

- The I/O port block representing the I/O pins on the I/O connector(s)
- The Tx block representing the line driver circuit
- The Output Inverter block representing the user-configurable logic inverter
- The Line Output Mux block representing the user-configurable source multiplexer

Line Mode Control

For all the I/O lines, the `LineMode` feature reports the configuration of the I/O port.

Only the TTL I/O lines have a user-configurable line mode.

A TTLIO can operate as an input or as an output.

When used as input, no configuration is required since at power-up all TTLIO ports are configured as Inputs.

When used as output, the user has the choice between 3 configurations of the line driver. The line driver can be configured in 3 ways:

- As a totem-pole driver capable of driving low and high.
- As an open-collector driver capable of driving low only
- As an open-emitter driver capable of driving high only.

The two latest configurations allows wired-AND configurations.

I/O Lines Specifics

Line Selector	Bit#	Line Format	Line Mode
DIN11	0	DIFF	Input
DIN12	1	DIFF	Input
DIN21	2	DIFF	Input
DIN22	3	DIFF	Input
IIN11	4	ISO	Input
IIN12	5	ISO	Input
IIN13	6	ISO	Input
IIN14	7	ISO	Input
IIN21	8	ISO	Input
IIN22	9	ISO	Input
IIN23	10	ISO	Input
IIN24	11	ISO	Input
IOUT11	12	ISO	Output
IOUT12	13	ISO	Output
IOUT21	14	ISO	Output
IOUT22	15	ISO	Output
TTLIO11	16	TTL	Input, Output, DriveLow or DriveHigh
TTLIO12	17	TTL	Input, Output, DriveLow or DriveHigh
TTLIO21	18	TTL	Input, Output, DriveLow or DriveHigh
TTLIO22	19	TTL	Input, Output, DriveLow or DriveHigh

The above table summarizes the details of the I/O Control blocks of each I/O line:

- The first column indicates the `LineSelector` value
- The second column indicates the bit position in the integer value reported by `LineStatusAll`
- The third column indicates the value reported by `LineFormat`
- The fourth column indicates the values that can be assigned to `LineMode`

Note: *Check the Available I/O Lines on page 122 for applicable values.*

Line Polarity Control

All the I/O lines are fitted with a polarity control. For bidirectional I/O lines, a single control affects equally both paths. The line polarity is user-configurable through the `LineInverter` control.

Note: *The user is invited to set the polarity control according to the polarity of the external signal in such a way that the Line Input signals of the input path and the LineSources signals of the output path are always using positive logic.*

Filter Control

All the I/O input lines are fitted with a glitch-removal filter. The filter strength is user-configurable through the `LineFilterStrength` control.

The strength control provides 5 positions from 'Lowest' to 'Highest'. The default position is 'Lowest'.

Each position corresponds to a specific filter time constant for each of the 3 I/O input line types:

Line Filter Time Constant per I/O input line type and LineFilterStrength

LineFilterStrength	Differential inputs (DIN)	TTL inputs (TTLIO)	Isolated Inputs (IIN)
Lowest	50 ns	50 ns	500 ns
Low	100 ns	100 ns	1 µs
Medium	200 ns	200 ns	2 µs
High	500 ns	500 ns	5 µs
Highest	1 µs	1 µs	10 µs

The user is invited to set the filter strength according to the quality of the external signal. Select a filter strength such that its time constant is:

- Greater than the longest glitch duration
- Greater than the 10%~90% rise/fall time of the signal
- At least 2 times smaller than the smallest signal pulse duration

The glitch removal filter introduces a latency into the input signal path. The latency is equal to the filter time constant when the incoming signal has clean transitions. The latency may increase significantly in case of bad quality signals.

Line Source Selection

Any output-capable I/O lines is fitted with a source signal multiplexer.

The source signal multiplexers implement a fully-populated signal routing matrix allowing a selection of internal signals to be routed to any output lines.

Selecting a signal source

To select an internal signal to be used as signal source for the line driver of an output capable I/O line:

- Step1: Select an I/O line by assigning the appropriate value to the `LineSelector` feature
- Step2: Assign the appropriate value to the `SignalSource` feature

Output-lines routing matrix

Source Signal	Possible destinations
Any Bit of the User Output Register	Any output line
Any Device Module Camera Trigger Output	Any TTLIO output line

Source Signal	Possible destinations
Any Device Module Strobe Output	Any output line
Steady Low	Any output line

Logical I/O Line State

Logical I/O line state

The (logical) state of an I/O line is the logical state of an electrical signal of the I/O control block:

- For input-capable I/O lines: the *LineInput* signal: a point in the input path of the I/O control block that is located after the Input Inverter.
- For output-only I/O lines: the *LineOutput* signal, a point in the output path of the I/O control block that is located before the Output Inverter.

Getting the state of a single I/O line

- Step1: Select an I/O line by assigning the appropriate value to the `LineSelector` feature
- Step2: Obtain directly the line status by getting the value of the `LineStatus` feature

Getting the state of all I/O lines in a single operation

Get the value of the `LineStatusAll` feature.

Each bit of the integer corresponds to an I/O line. A bit at one corresponds to a line logical state being high.

Physical I/O Line State

The physical state of the I/O line state does not only depend on the value reported when reading `LineStatus` but also on the following I/O block settings: `LineFormat`, `LineMode`, and `LineInverter`

LineFormat	LineMode	LineInverter/LineStatus	Physical I/O line state
DIFF	Input	False/False or True/True	*(VIN+ - VIN-) < VThreshold*
		False/True or True/False	*(VIN+ - VIN-) > VThreshold or unconnected I/O line*
ISO	Input	False/False or True/True	Opto coupler is OFF. The line current is < 1 mA. *Line may be left unconnected or connected with the wrong polarity.*
		False/True or True/False	Opto coupler is ON. The line current is > 1 mA.
ISO	Output	False/False or True/True	Opto coupler is OFF
		False/True or True/False	Opto coupler is ON
TTL	Input	False/False or True/True	The line voltage is < 0.8 Volt.
		False/True or True/False	The line voltage is > 2.0 Volt.
TTL	Output	False/False or True/True	The line is driven LOW.

LineFormat	LineMode	LineInverter/LineStatus	Physical I/O line state
		False/True or True/False	The line is driven HIGH.
TTL	DriveLow	False/False or True/True	The line is driven LOW.
		False/True or True/False	The line voltage is > 2.0 Volt.
TTL	DriveHigh	False/False or True/True	The line voltage is < 0.8 Volt.
		False/True or True/False	The line is driven HIGH.

Line Driver Physical Output States

The line driver output state depends not only on the logical level of the selected Line Output signal but also on the following I/O block settings: `LineFormat`, `LineMode`, and `LineInverter`.

LineFormat	LineMode	LineInverter / LineOutput logical level	Line driver output state
ISO	Output	False / low or True / high	The opto-coupler switch is turned OFF.
		False / high or True / low	The opto-coupler switch is turned ON.
TTL	Output	False / low or True / high	The I/O line is driven LOW.
		False / high or True / low	The I/O line is driven HIGH.
TTL	DriveLow	False / low or True / high	The I/O line is driven LOW.
		False / high or True / low	The I/O line is not driven.
TTL	DriveHigh	False / low or True / high	The I/O line is not driven.
		False / high or True / low	The I/O line is driven HIGH.

Initial States

At power-on, the I/O lines are in the following state:

Differential Inputs

LineInverter = False

LineFilterStrength = Low

TTL Inputs/outputs

LineMode = Input (The line is not driven)

LineInverter = False

LineFilterStrength = Low

Isolated Inputs

LineInverter = False

LineFilterStrength = Low

Isolated Outputs

LineInverter = False
LineFilterStrength = Low
The opto-coupler is OFF.

Digital I/O Control GenICam Features

LineSelector		
Interface	Root > Digital I/O	Enumeration (RW)

Selects one physical GPIO line.

Possible values:

- `DIN11`: Differential input 1 of Internal I/O connector 1. **Default value.**
- `DIN12`: Differential input 2 of Internal I/O connector 1.
- `DIN21`: Differential input 1 of Internal I/O connector 2.
- `DIN22`: Differential input 2 of Internal I/O connector 2.
- `IIN11`: Isolated input 1 of Internal I/O connector 1.
- `IIN12`: Isolated input 2 of Internal I/O connector 1.
- `IIN13`: Isolated input 3 of Internal I/O connector 1.
- `IIN14`: Isolated input 4 of Internal I/O connector 1.
- `IIN21`: Isolated input 1 of Internal I/O connector 2.
- `IIN22`: Isolated input 2 of Internal I/O connector 2.
- `IIN23`: Isolated input 3 of Internal I/O connector 2.
- `IIN24`: Isolated input 4 of Internal I/O connector 2.
- `IOUT11`: Isolated output 1 of Internal I/O connector 1.
- `IOUT12`: Isolated output 2 of Internal I/O connector 1.
- `IOUT21`: Isolated output 1 of Internal I/O connector 2.
- `IOUT22`: Isolated output 2 of Internal I/O connector 2.
- `TTLIO11`: TTL input/output 1 of Internal I/O connector 1.
- `TTLIO12`: TTL input/output 2 of Internal I/O connector 1.
- `TTLIO21`: TTL input/output 1 of Internal I/O connector 2.
- `TTLIO22`: TTL input/output 2 of Internal I/O connector 2.

Note: *Check the Available I/O Lines on page 122 for applicable values.*

LineFormat		
Interface	Root > Digital I/O	Enumeration (RO)

Returns the electrical style of the selected physical GPIO line.

Possible values:

- `DIFF`: The I/O line is accepting LVDS or RS422 level signals. *Applies to differential input lines.*
- `ISO`: The I/O line is optically isolated. *Applies to isolated inputs and isolated outputs lines.*
- `TTL`: The I/O line is accepting or sending TTL level signals. *Applies to TTL input/output lines.*

LineMode

Interface	Root > Digital I/O	Enumeration (RW)

Direction and line driver mode of the selected physical GPIO line.

Note: *The feature is writable only for TTL input/output lines.*

Possible values:

- `Input`: Input line. *See note 1*.*
- `Output`: Output line. *See notes 2* and 4*.*
- `DriveLow`: Driven-low output line. *See notes 3* and 4*.*
- `DriveHigh`: Driven-high output line. *See note 3* and 4*.*

Note: *1* This value is only available for the input-capable I/O lines.*

Note: *2* This value is only available for the output-capable I/O lines.*

Note: *3* This value is only available for the TTL input/output lines.*

Note: *4* For TTL input/output lines only:*

- *The value `Output` selects the **totem-pole** operation mode of the line driver; the I/O line is driven low and high.*
- *The value `DriveLow` selects the **open-collector** operation mode of the line driver; the I/O line is only driven low.*
- *The value `DriveHigh` selects the **open-emitter** operation mode of the line driver; the I/O line is only driven high.*

Default values:

- `Input` for the input-capable GPIO lines.
- `Output` for the output-only GPIO lines.

LineInverter

Interface	Root > Digital I/O	Boolean (RW)

Signal inversion of the selected input or output line.

Possible values:

- `False`: The line signal is not inverted. **Default value.**
- `True`: The line signal is inverted.

Note: *For bidirectional I/O line such as the TTL input/output lines, the settings applies equally to the signal input path and the signal output path!*

LineFilterStrength

Interface	Root > Digital I/O	Enumeration (RW)

LineFilterStrength

Strength of the glitch removal filter of the selected physical GPIO line.

Note: *This feature is only available for input-capable GPIO lines.*

Possible values:

- `Lowest`: Lowest filter strength. **Default value.**
- `Low`: Low filter strength.
- `Medium`: Medium filter strength.
- `High`: High filter strength.
- `Highest`: Highest filter strength.

Note: *For high-speed differential and TTL inputs, the corresponding filter latency delay is respectively 50 nanoseconds (lowest), 100 nanoseconds, 200 nanoseconds, 500 nanoseconds, and 1 microsecond (highest).*

Note: *For low-speed isolated inputs, the corresponding filter latency delay is 10 times longer than for high-speed inputs. Respectively 500 nanoseconds (lowest), 1 microsecond, 2 microseconds, 5 microseconds, and 10 microseconds (highest).*

LineFilterDelay

Interface	Root > Digital I/O	Float (RO)

Returns the latency delay introduced by the glitch removal filter of the selected physical GPIO line.

Note:

This feature is only available for input-capable GPIO lines.

Unit: microseconds.

LineStatus

Interface	Root > Digital I/O	Boolean (RO)

Returns the current status of the selected physical GPIO line.

Possible values:

- `False`: The logical state of the selected physical GPIO line is low.
- `True`: The logical state of the selected physical GPIO line is high.

Note: *For input-capable I/O lines, the reported value is the logical state of the LineInput signal: a point in the input path of the I/O control block that is located after the Input Inverter.*

Note: *For output-only I/O lines, the reported value is the logical state of the LineOutput signal, a point in the output path of the I/O control block that is located before the Output Inverter.*

LineStatusAll

Interface	Root > Digital I/O	Integer (RO)

LineStatusAll

Returns the current state of all available GPIO line signals at time of polling in a single bit field.

Bit values:

- `0`: The logical state of the corresponding GPIO line is low.
- `1`: The logical state of the corresponding GPIO line is high.

Bit assignments:

- Bit 0: DIN11 GPIO line.
- Bit 1: DIN12 GPIO line.
- Bit 2: DIN21 GPIO line.
- Bit 3: DIN22 GPIO line.
- Bit 4: IIN11 GPIO line.
- Bit 5: IIN12 GPIO line.
- Bit 6: IIN13 GPIO line.
- Bit 7: IIN14 GPIO line.
- Bit 8: IIN21 GPIO line.
- Bit 9: IIN22 GPIO line.
- Bit 10: IIN23GPIO line.
- Bit 11: IIN24 GPIO line.
- Bit 12: IOUT11 GPIO line.
- Bit 13: IOUT12 GPIO line.
- Bit 14: IOUT21 GPIO line.
- Bit 15: IOUT22 GPIO line.
- Bit 16: TTLIO11 GPIO line.
- Bit 17: TTLIO12 GPIO line.
- Bit 18: TTLIO21 GPIO line.
- Bit 19: TTLIO22 GPIO line.

LineSource

Interface	Root > Digital I/O	Enumeration (RW)

LineSource

Internal signal sourcing the output of the selected physical GPIO line.

Note:

This feature is only available for output-capable GPIO lines.

Possible values:

- `UserOutput0`: Bit 0 of the user output register.
- `UserOutput1`: Bit 1 of the user output register.
- `UserOutput2`: Bit 2 of the user output register.
- `UserOutput3`: Bit 3 of the user output register.
- `UserOutput4`: Bit 4 of the user output register.
- `UserOutput5`: Bit 5 of the user output register.
- `UserOutput6`: Bit 6 of the user output register.
- `UserOutput7`: Bit 7 of the user output register.
- `Device0Strobe`: Strobe output of device 0.
- `Device1Strobe`: Strobe output of device 1.
- `Device2Strobe`: Strobe output of device 2.
- `Device3Strobe`: Strobe output of device 3.
- `Device0CameraTrigger`: Camera Trigger output of device 0.
- `Device1CameraTrigger`: Camera Trigger output of device 1.
- `Device2CameraTrigger`: Camera Trigger output of device 2.
- `Device3CameraTrigger`: Camera Trigger output of device 3.
- `Device0Stream0StartOfCameraReadout`: Start of camera readout on stream 0 of device 0.
- `Device1Stream0StartOfCameraReadout`: Start of camera readout on stream 0 of device 1.
- `Device2Stream0StartOfCameraReadout`: Start of camera readout on stream 0 of device 2.
- `Device3Stream0StartOfCameraReadout`: Start of camera readout on stream 0 of device 3.
- `Low`: Steady low signal.

Event Signaling And Counting

Extensive user-configurable event-reporting and event-counting mechanism.

Short Description

Coaxlink products feature a powerful event management that allows the application to be notified of the occurrence of various events.

In addition to the GenTL standard EVENT_NEW_BUFFER event, the Coaxlink GenTL producer provide a wide set of custom event sources. The event sources are grouped by types according to the function block and the GenTL module they belong to.

Each custom event source is associated with a counter that counts the number of occurrences.

For each notified custom event, the following event context data is recorded and made available to the application:

- Identifier of the event source
- Time stamp (expressed in microseconds)
- 3 user-defined context data

Each individual event source is configurable:

- The event notification can enabled or disabled.
- The content of each user-defined context data.

Event data are temporarily stored in the Event Queue Buffer. The Coaxlink Driver is notified, using an interruption mechanism, of the availability of one or more event entries in the Coaxlink Event Queue Buffer.

The Coaxlink Driver implements the GenTL signaling mechanism for reporting the occurrence of asynchronous events to the application software.

The EGrabber API provides 3 callback threading models:

- **CallbackOnDemand:** This is the simplest model which gives complete control over when and how callbacks are invoked. Events are processed on demand.
- **CallbackSingleThread**: This model delivers events to callbacks in their chronological order, sequentially, in a dedicated thread context. Events are processed automatically as soon as they are available.
- **CallbackMultiThread**: This model delivers events to callbacks in separate threads (one thread per event DATA type). Events are processed automatically as soon as they are available.

About GenTL Signaling

The Coaxlink Driver implements the Signaling mechanism of a GenTL Provider.

This mechanism is briefly described hereafter. For an extensive description, refer to section 4.2 starting on page 34 of the *GenICam GenTL Standard Version 1.4.*

Event Registration

Source: GenTL specification

Before the GenTL Consumer can be informed about an event, the event object must be registered. After a module instance has been created in the enumeration process an event object can be created with the GCRegisterEvent() function. This function returns a unique EVENT_HANDLE which identifies the registered event object. To get information about a registered event the EventGetInfo() function can be used.

There must be only one event registered per module and event type.

(…)

After an EVENT_HANDLE is obtained the GenTL Consumer can wait for the event object to be signaled by calling the EventGetData() function. Upon delivery of an event, the event object carries data. This data is copied into a GenTL Consumer provided buffer when the call to EventGetData() was successful.

Notification and Data Retrieval

Source: GenTL specification

If the event object is signaled, data was put into the event data queue at some point in time. The EventGetData() function can be called to retrieve the actual data.

(…)

When data is read with this function the data is removed from the queue. Afterwards the GenTL Producer implementation checks whether the event data queue is empty or not. If there is more data available the event object stays signaled and next the call to EventGetData() will deliver the next queue entry. Otherwise the event object is reset to not signaled state.

(…)

The exact type of data is dependent on the event type and the GenTL Producer implementation. The data is copied into a user buffer allocated by the GenTL Consumer. The content of the event data can be queried with the EventGetDataInfo() function. The maximum size of the buffer to be filled is defined by the event type and can be queried using EVENT_INFO_DATA_SIZE_MAX after the buffer is delivered. This information can be queried using the EventGetInfo() function.

Event Types

List of standard and custom GenTL event types.

GenTL identifies the events according to their Type and their relevant object Module.

Standard Event Types

The Coaxlink Driver implements the following standard event type for registration by the GenTL Consumer application:

GenTL Standard Event Type	GenTL Module	Description
EVENT_NEW_BUFFER	Data Stream	Notification on newly filled buffers.

Custom Event Types

Beside the **standard event types**, the GenTL specification provides room for **custom event types**. Custom event types are specific to the GenTL Producer implementation (Coaxlink Driver in this case).

The Coaxlink Driver implements the following custom event types for registration by the GenTL Consumer application:

Event Type	Module	Description
EVENT_CUSTOM_IO_TOOLBOX	Interface (Device)	Notification of I/O Toolbox events
EVENT_CUSTOM_CXP_INTERFACE	Interface (Device)	Notification of CoaXPress Host Interface events
EVENT_CUSTOM_CIC	Device	Notification of Camera and Illumination Control events
EVENT_CUSTOM_DATASTREAM	Data Stream	Notification of CoaXPress data stream events

Note: *The EVENT_CUSTOM_IO_TOOLBOX and the EVENT_CUSTOM_CXP_INTERFACE event types can also be registered on a Device Module.*

Note: *The custom event types are generic; each one gathers multiple event sources.*

Custom Events Sources

List of all available custom event sources grouped by custom event types.

EVENT_CUSTOM_IO_TOOLBOX

EVENT_CUSTOM_IO_TOOLBOX event sources

Event Source	Description
IO_TOOLBOX_LIN1	Line Input Tool 1 – Event output
IO_TOOLBOX_LIN2	Line Input Tool 2 – Event output
IO_TOOLBOX_LIN3	Line Input Tool 3 – Event output
IO_TOOLBOX_LIN4	Line Input Tool 4 – Event output
IO_TOOLBOX_LIN5	Line Input Tool 5 – Event output
IO_TOOLBOX_LIN6	Line Input Tool 6 – Event output
IO_TOOLBOX_LIN7	Line Input Tool 7 – Event output
IO_TOOLBOX_LIN8	Line Input Tool 8 – Event output
IO_TOOLBOX_QDC1	Quadrature Decoder Tool 1 – Event output
IO_TOOLBOX_QDC1_DIR	Quadrature Decoder Tool 1 – Changed direction
IO_TOOLBOX_QDC2	Quadrature Decoder Tool 2 – Event output
IO_TOOLBOX_QDC2_DIR	Quadrature Decoder Tool 2 – Changed Direction
IO_TOOLBOX_DIV1	Divider Tool 1 – Event output
IO_TOOLBOX_DIV2	Divider Tool 2 – Event output
IO_TOOLBOX_MDV1	Multiplier/Divider Tool 1 – Event output
IO_TOOLBOX_MDV2	Multiplier/Divider Tool 2 – Event output
IO_TOOLBOX_DEL11	Delay Tool 1 Output 1 – Event output
IO_TOOLBOX_DEL12	Delay Tool 1 Output 2 – Event output
IO_TOOLBOX_DEL21	Delay Tool 2 Output 1 – Event output
IO_TOOLBOX_DEL22	Delay Tool 2 Output 2 – Event output
IO_TOOLBOX_USER_EVENT_1	User Event 1
IO_TOOLBOX_USER_EVENT_2	User Event 2
IO_TOOLBOX_USER_EVENT_3	User Event 3
IO_TOOLBOX_USER_EVENT_4	User Event 4
IO_TOOLBOX_EIN1	Event Input Tool 1 – Event output

Event Source	Description
IO_TOOLBOX_EIN2	Event Input Tool 2 – Event output

This event type belongs to the Interface Module, it gathers all the event sources located inside the I/O Toolbox.
All the event outputs of all the I/O Toolbox tools are available as event sources.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

EVENT_CUSTOM_CXP_INTERFACE

Event Source	Description
CXP_INTERFACE_CRC_ERROR_CXP_A	A CRC error is detected on the Connection A of the CoaXPress Host Interface.
CXP_INTERFACE_CRC_ERROR_CXP_B	A CRC error is detected on the Connection B of the CoaXPress Host Interface.
CXP_INTERFACE_CRC_ERROR_CXP_C	A CRC error is detected on the Connection C of the CoaXPress Host Interface.
CXP_INTERFACE_CRC_ERROR_CXP_D	A CRC error is detected on the Connection A of the CoaXPress Host Interface.

This event type belongs to the Interface Module, it gathers all the event sources located inside the CoaXPress Host Interface.

EVENT_CUSTOM_CIC

EVENT_CUSTOM_CIC event sources

Event Source	Description
CIC_CAMERA_TRIGGER_RISING_EDGE	Rising edge of the Camera Trigger output signal (Start of Exposure in RC and RG camera control methods)
CIC_CAMERA_TRIGGER_FALLING_EDGE	Falling edge of the Camera Trigger output signal (End of Exposure in RG camera control method)
CIC_STROBE_RISING_EDGE	Rising edge of the Strobe output signal
CIC_STROBE_FALLING_EDGE	Falling edge of the Strobe output signal
CIC_ALLOW_NEXT_CYCLE	A new camera cycle is allowed to start immediately.
CIC_DISCARDED_CIC_TRIGGER	A CIC cycle trigger is discarded.
CIC_PENDING_CIC_TRIGGER	A CIC cycle trigger is recorded, but its execution is delayed until CIC is ready.
CIC_CXP_TRIGGER_ACK	A positive acknowledgment is received in response to a CoaXPress Host to Device Trigger Message.
CIC_CXP_TRIGGER_RESEND	A resent of the CoaXPress Host to Device Trigger Message is executed.
CIC_TRIGGER	CIC trigger

This event type belongs to the Device Module, it gathers all the event sources located inside the Camera and Illumination Controller.

Note: *There is one Camera and Illumination Controller instance per GenTL Device Module. The number of GenTL Device Modules per Coaxlink card is defined by the firmware-variant.*

EVENT_CUSTOM_DATASTREAM

Event Source	Description
DATASTREAM_START_OF_CAMERA_READOUT	The first pixel data of an image frame is written into the on-board FIFO Buffer. Applies to area-scan firmware variants only.
DATASTREAM_END_OF_CAMERA_READOUT	The last pixel data of an image frame is written into the on-board FIFO Buffer. Applies to area-scan firmware variants only.
DATASTREAM_START_OF_SCAN	The first pixel data of an image scan is written into the on-board FIFO Buffer. Applies to line-scan firmware variants only.
DATASTREAM_END_OF_SCAN	The last pixel data of an image scan is written into the on-board FIFO Buffer. Applies to line-scan firmware variants only.
DATASTREAM_REJECTED_FRAME	An image frame is rejected (On-board FIFO Buffer is full) Applies to area-scan firmware variants only.
DATASTREAM_REJECTED_SCAN	An image scan is rejected (On-board FIFO Buffer is full) Applies to line-scan firmware variants only.

This event type belongs to the Data Stream Module, it gathers all the event sources located inside the Image Acquisition Controller.

Custom Events Counter

Counter associated with every custom event.

A 32-bit counter is associated with every custom event source.

The counter cannot be disabled. When it reaches its maximum value, 4 294 967 295 (2^{32} - 1), it wraps around to 0.

At any time, the user application can:

- Reset the counter.
- Read the count value.

The count-value can also be used as user-defined context data by any event source.

Custom Events Configuration

Configuring the notification filter and the context data of custom event sources.

Each individual event source is configurable:

- The event notification can enabled or disabled.
- The content of each user-defined context data.

Event Notification Filter

By default, all notifications are disabled.

The application software must configure the event notification filter according the application needs.

The configuration of the notification filter configuration can be modified at any time without interfering with the event counting function.

User-defined Event Context Data

The last 3 32-bit context data words of the event context data can be configured as follows:

- Event-specific data.
- State of all I/O lines sampled at the event occurrence time
- Count value of any event counter.

Event Specific Context Data

Context data that is available only for a selected set of custom events.

EVENT_DATA_NUMID_CIC_DISCARDED_CIC_TRIGGER

Value of EventSpecific for EVENT_DATA_NUMID_CIC_DISCARDED_CIC_TRIGGER is a bitfield that can be interpreted according to the following definitions:

Bit#	Description
0	Cause: image buffer is full.
1	Cause: camera cycle not complete.
2	Cause: maximum number of pending triggers already reached.
3	Cause: data stream is not active

EVENT_DATA_NUMID_CIC_PENDING_CIC_TRIGGER

Value of EventSpecific for EVENT_DATA_NUMID_CIC_PENDING_CIC_TRIGGER is a bitfield that can be interpreted according to the following definitions:

Bit#	Description
0	Cause: image buffer is full.
1	Cause: camera cycle not complete

EVENT_DATA_NUMID_DATASTREAM_START_OF_SCAN

Value of EventSpecific for EVENT_DATA_NUMID_DATASTREAM_START_OF_SCAN is a bitfield that can be interpreted according to the following definitions:

Bit#	Description
1	Cause:software trigger.
2	Cause: hardware trigger
3	Cause: `DSStartAcquisition` or end of previous scan

EVENT_DATA_NUMID_DATASTREAM_END_OF_SCAN

Value of EventSpecific for EVENT_DATA_NUMID_DATASTREAM_END_OF_SCAN is a bitfield that can be interpreted according to the following definitions:

Bit#	Description
1	Cause:software trigger.
2	Cause: hardware trigger
3	Cause: Cause: reached scan length
4	Cause: `DSStopAcquisition`
5	Cause: internal exception (image buffer almost full)

EVENT_DATA_NUMID_DATASTREAM_REJECTED_FRAME

Value of EventSpecific for EVENT_DATA_NUMID_DATASTREAM_REJECTED_FRAME is a bitfield that can be interpreted according to the following definitions:

Bit#	Description
0	Cause: image buffer is full.
1	Cause: data stream is not active

Event Control GenICam Features (Interface)

EventSelector		
Interface	Root > EventControl	Enumeration (RW)

EventSelector

Select an event.

Possible values:

- `LIN1`: Line Input Tool 1 I/O Toolbox event output.
- `LIN2`: Line Input Tool 2 I/O Toolbox event output.
- `LIN3`: Line Input Tool 3 I/O Toolbox event output.
- `LIN4`: Line Input Tool 4 I/O Toolbox event output.
- `LIN5`: Line Input Tool 5 I/O Toolbox event output.
- `LIN6`: Line Input Tool 6 I/O Toolbox event output.
- `LIN7`: Line Input Tool 7 I/O Toolbox event output.
- `LIN8`: Line Input Tool 8 I/O Toolbox event output.
- `QDC1`: Quadrature Decoder Tool 1.
- `QDC2`: Quadrature Decoder Tool 2.
- `DIV1`: Divider Tool 1.
- `DIV2`: Divider Tool 2.
- `MDV1`: Multiplier/Divider Tool 1 I/O Toolbox event output.
- `MDV2`: Multiplier/Divider Tool 2 I/O Toolbox event output.
- `DEL11`: Delay Tool 1 Output 1.
- `DEL12`: Delay Tool 1 Output 2.
- `DEL21`: Delay Tool 2 Output 1.
- `DEL22`: Delay Tool 2 Output 2.
- `UserEvent1`: User Event 1.
- `UserEvent2`: User Event 2.
- `UserEvent3`: User Event 3.
- `UserEvent4`: User Event 4.
- `EIN1`: Event Input Tool 1.
- `EIN2`: Event Input Tool 2.
- `CrcErrorCxpA`: Detected a data packet CRC error on CoaXPress connection A.
- `CrcErrorCxpB`: Detected a data packet CRC error on CoaXPress connection B.
- `CrcErrorCxpC`: Detected a data packet CRC error on CoaXPress connection C.
- `CrcErrorCxpD`: Detected a data packet CRC error on CoaXPress connection D.

Note: *Check the I/O Toolbox Composition on page 97 for applicable values*

EventNotification

Interface	Root > EventControl	Boolean (RW)

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

Possible values:

- `On`: Activate the notification.
- `Off`: Deactivate the notification. **Default value.**

EventNotificationContext1

Interface	Root > EventControl	Enumeration (RW)

EventNotificationContext1

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_1 (context information value is latched when the event occurs).

Possible values:

- `EventSpecific`: Event-specific context information. **Default value.**
- `LineStatusAll`: Value of `LineStatusAll`
- `PendingCicTriggerCount`: Number of currently pending CIC triggers.
- `LIN1EventCount`: Number of LIN1 (Line Input Tool 1) events.
- `LIN2EventCount`: Number of LIN2 (Line Input Tool 2) events.
- `LIN3EventCount`: Number of LIN3 (Line Input Tool 3) events.
- `LIN4EventCount`: Number of LIN4 (Line Input Tool 4) events.
- `LIN5EventCount`: Number of LIN5 (Line Input Tool 5) events.
- `LIN6EventCount`: Number of LIN6 (Line Input Tool 6) events.
- `LIN7EventCount`: Number of LIN7 (Line Input Tool 7) events.
- `LIN8EventCount`: Number of LIN8 (Line Input Tool 8) events.
- `QDC1EventCount`: Number of QDC1 (Quadrature Decoder Tool 1) events.
- `QDC1DirEventCount`: Number of QDC1Dir (Quadrature Decoder Tool 1 Direction Change) events.
- `QDC2EventCount`: Number of QDC2 (Quadrature Decoder Tool 2) events.
- `QDC2DirEventCount`: Number of QDC2Dir (Quadrature Decoder Tool 2 Ditection Change) events.
- `DIV1EventCount`: Number of DIV1 (Divider Tool 1) events.
- `DIV2EventCount`: Number of DIV2 (Divider Tool 2) events.
- `MDV1EventCount`: Number of MDV1 (Multiplier/Divider Tool1) events.
- `MDV2EventCount`: Number of MDV2 (Multiplier/Divider Tool2) events.
- `DEL11EventCount`: Number of DEL11 (Delay Tool 1Output 1) events.
- `DEL12EventCount`: Number of DEL12 (Delay Tool 1 Output 2) events.
- `DEL21EventCount`: Number of DEL21 (Delay Tool 2 Output 1) events.
- `DEL11EventCount`: Number of DEL22 (Delay Tool 2 Output 2) events.
- `UserEvent1EventCount`: Number of User Event 1 events.
- `UserEvent2EventCount`: Number of User Event 2 events.
- `UserEvent3EventCount`: Number of User Event 3 events.
- `UserEvent4EventCount`: Number of User Event 4 events.
- `EIN1EventCount`: Number of EIN1 (Event Input Tool 1) events.
- `EIN2EventCount`: Number of EIN2 (Event Input Tool 2) events.
- `CrcErrorCxpAEventCount`: Number of CrcErrorCxpA events.
- `CrcErrorCxpBEventCount`: Number of CrcErrorCxpB events.
- `CrcErrorCxpCEventCount`: Number of CrcErrorCxpC events.
- `CrcErrorCxpDEventCount`: Number of CrcErrorCxpD events.
- `CameraTriggerRisingEdgeEventCount`: Number of CameraTriggerRisingEdge events.
- `CameraTriggerFallingEdgeEventCount`: Number of CameraTriggerFallingEdge events.
- `StrobeRisingEdgeEventCount`: Number of StrobeRisingEdge events.
- `StrobeFallingEdgeEventCount`: Number of StrobeFallingEdge events.
- `AllowNextCycleEventCount`: Number of AllowNextCycle events.
- `DiscardedCicTriggerEventCount`: Number of DiscardedCicTrigger events.
- `PendingCicTriggerEventCount`: Number of PendingCicTrigger events.
- `CxpTriggerAckEventCount`: Number of CxpTriggerAck events.
- `CxpTriggerResendEventCount`: Number of CxpTriggerResend events.
- `TriggerEventCount`: Number of Trigger events.
- `StartOfCameraReadoutEventCount`: Number of StartOfCameraReadout events.
- `EndOfCameraReadoutEventCount`: Number of EndOfCameraReadout events.
- `StartOfScanEventCount`: Number of StartOfScan events.
- `EndOfScanEventCount`: Number of EndOfScan events.
- `RejectedFrameEventCount`: Number of RejectedFrame events.
- `RejectedScanEventCount`: Number of RejectedScan events.

EventNotificationContext2

Interface	Root > EventControl	Enumeration (RW)

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_2 (context information value is latched when the event occurs).

Possible values: Same value set as for `EventNotificationContext1`

EventNotificationContext3

Interface	Root > EventControl	Enumeration (RW)

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_3 (context information value is latched when the event occurs).

Possible values: Same value set as for `EventNotificationContext1`

EventCount

Interface	Root > EventControl	Integer (RW)

Number of occurrences of the selected event (32-bit counter).

EventCountReset

Interface	Root > EventControl	Command (WO)

Reset the selected event count.

Event Control GenICam Features (Device)

EventSelector

Device	Root > EventControl	Enumeration (RW)

Select an event.

Possible values:

- `CameraTriggerRisingEdge`: Rising edge of CIC Camera Trigger output signal.
- `CameraTriggerFallingEdge`: Falling edge of CIC Camera Trigger output signal.
- `StrobeRisingEdge`: Rising edge of CIC Strobe output signal.
- `StrobeFallingEdge`: Falling edge of CIC Strobe output signal.
- `AllowNextCycle`: CIC is ready for next camera cycle.
- `DiscardedCicTrigger`: Ignored CIC trigger because CIC is not ready for next camera cycle.
- `PendingCicTrigger`: Delayed CIC trigger until CIC is ready for next camera cycle.
- `CxpTriggerAck`: Received acknowledgement for previous CXP trigger message.
- `CxpTriggerResend`: Resent CXP trigger message (acknowledgement to previous CXP trigger message not received).
- `Trigger`: CIC trigger input.

EventNotification		
Device	Root > EventControl	Boolean (RW)

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

Possible values:

- `On`: Activate the notification.
- `Off`: Deactivate the notification. **Default value.**

EventNotificationContext1		
Device	Root > EventControl	Enumeration (RW)

EventNotificationContext1

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_1 (context information value is latched when the event occurs).

Possible values:

- `EventSpecific`: Event-specific context information. **Default value.**
- `LineStatusAll`: Value of `LineStatusAll`
- `PendingCicTriggerCount`: Number of currently pending CIC triggers.
- `LIN1EventCount`: Number of LIN1 (Line Input Tool 1) events.
- `LIN2EventCount`: Number of LIN2 (Line Input Tool 2) events.
- `LIN3EventCount`: Number of LIN3 (Line Input Tool 3) events.
- `LIN4EventCount`: Number of LIN4 (Line Input Tool 4) events.
- `LIN5EventCount`: Number of LIN5 (Line Input Tool 5) events.
- `LIN6EventCount`: Number of LIN6 (Line Input Tool 6) events.
- `LIN7EventCount`: Number of LIN7 (Line Input Tool 7) events.
- `LIN8EventCount`: Number of LIN8 (Line Input Tool 8) events.
- `QDC1EventCount`: Number of QDC1 (Quadrature Decoder Tool 1) events.
- `QDC1DirEventCount`: Number of QDC1Dir (Quadrature Decoder Tool 1 Direction Change) events.
- `QDC2EventCount`: Number of QDC2 (Quadrature Decoder Tool 2) events.
- `QDC2DirEventCount`: Number of QDC2Dir (Quadrature Decoder Tool 2 Ditection Change) events.
- `DIV1EventCount`: Number of DIV1 (Divider Tool 1) events.
- `DIV2EventCount`: Number of DIV2 (Divider Tool 2) events.
- `MDV1EventCount`: Number of MDV1 (Multiplier/Divider Tool1) events.
- `MDV2EventCount`: Number of MDV2 (Multiplier/Divider Tool2) events.
- `DEL11EventCount`: Number of DEL11 (Delay Tool 1Output 1) events.
- `DEL12EventCount`: Number of DEL12 (Delay Tool 1 Output 2) events.
- `DEL21EventCount`: Number of DEL21 (Delay Tool 2 Output 1) events.
- `DEL11EventCount`: Number of DEL22 (Delay Tool 2 Output 2) events.
- `EIN1EventCount`: Number of EIN1 (Event Input Tool 1) events.
- `EIN2EventCount`: Number of EIN2 (Event Input Tool 2) events.
- `CrcErrorCxpAEventCount`: Number of CrcErrorCxpA events.
- `CrcErrorCxpBEventCount`: Number of CrcErrorCxpB events.
- `CrcErrorCxpCEventCount`: Number of CrcErrorCxpC events.
- `CrcErrorCxpDEventCount`: Number of CrcErrorCxpD events.
- `CameraTriggerRisingEdgeEventCount`: Number of CameraTriggerRisingEdge events.
- `CameraTriggerFallingEdgeEventCount`: Number of CameraTriggerFallingEdge events.
- `StrobeRisingEdgeEventCount`: Number of StrobeRisingEdge events.
- `StrobeFallingEdgeEventCount`: Number of StrobeFallingEdge events.
- `AllowNextCycleEventCount`: Number of AllowNextCycle events.
- `DiscardedCicTriggerEventCount`: Number of DiscardedCicTrigger events.
- `PendingCicTriggerEventCount`: Number of PendingCicTrigger events.
- `CxpTriggerAckEventCount`: Number of CxpTriggerAck events.
- `CxpTriggerResendEventCount`: Number of CxpTriggerResend events.
- `TriggerEventCount`: Number of Trigger events.
- `StartOfCameraReadoutEventCount`: Number of StartOfCameraReadout events.
- `EndOfCameraReadoutEventCount`: Number of EndOfCameraReadout events.
- `StartOfScanEventCount`: Number of StartOfScan events.
- `EndOfScanEventCount`: Number of EndOfScan events.
- `RejectedFrameEventCount`: Number of RejectedFrame events.
- `RejectedScanEventCount`: Number of RejectedScan events.

EventNotificationContext2

Device	Root > EventControl	Enumeration (RW)

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_2 (context information value is latched when the event occurs).

Possible values: Same value set as for `EventNotificationContext1`

EventNotificationContext3

Device	Root > EventControl	Enumeration (RW)

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_3 (context information value is latched when the event occurs).

Possible values: Same value set as for `EventNotificationContext1`

EventCount

Device	Root > EventControl	Integer (RW)

Number of occurrences of the selected event (32-bit counter).

EventCountReset

Device	Root > EventControl	Command (WO)

Reset the selected event count.

Event Control GenICam Features (Data Stream)

EventSelector

Data Stream	Root > EventControl	Enumeration (RW)

Select an event.

Possible values (area-scan firmware variants):

- `StartOfCameraReadout`: The CoaXPress Host interface starts data acquisition of a new image frame.
- `EndOfCameraReadout`: The CoaXPress Host interface stops data acquisition of an image frame.
- `RejectedFrame`: Dropped image frame data.

Possible values (line-scan firmware variants):

- `StartOfScan`: The CoaXPress Host interface starts data acquisition of a new image scan.
- `EndOfScan`: The CoaXPress Host interface stops data acquisition of an image scan.
- `RejectedScan`: Dropped image scan data.

EventNotification

Data Stream	Root > EventControl	Boolean (RW)

EventNotification

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

Possible values:

- `On`: Activate the notification.
- `Off`: Deactivate the notification. **Default value.**

EventNotificationContext1

Data Stream	Root > EventControl	Enumeration (RW)

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_1 (context information value is latched when the event occurs).

Possible values:

- `EventSpecific`: Event-specific context information. **Default value.**
- `LineStatusAll`: Value of `LineStatusAll`
- `PendingCicTriggerCount`: Number of currently pending CIC triggers.
- `LIN1EventCount`: Number of LIN1 (Line Input Tool 1) events.
- `LIN2EventCount`: Number of LIN2 (Line Input Tool 2) events.
- `LIN3EventCount`: Number of LIN3 (Line Input Tool 3) events.
- `LIN4EventCount`: Number of LIN4 (Line Input Tool 4) events.
- `MDV1EventCount`: Number of MDV1 (Multiplier/Divider Tool1) events.
- `MDV2EventCount`: Number of MDV2 (Multiplier/Divider Tool2) events.
- `CrcErrorCxpAEventCount`: Number of CrcErrorCxpA events.
- `CrcErrorCxpBEventCount`: Number of CrcErrorCxpB events.
- `CrcErrorCxpCEventCount`: Number of CrcErrorCxpC events.
- `CrcErrorCxpDEventCount`: Number of CrcErrorCxpD events.
- `CameraTriggerRisingEdgeEventCount`: Number of CameraTriggerRisingEdge events.
- `CameraTriggerFallingEdgeEventCount`: Number of CameraTriggerFallingEdge events.
- `StrobeRisingEdgeEventCount`: Number of StrobeRisingEdge events.
- `StrobeFallingEdgeEventCount`: Number of StrobeFallingEdge events.
- `AllowNextCycleEventCount`: Number of AllowNextCycle events.
- `DiscardedCicTriggerEventCount`: Number of DiscardedCicTrigger events.
- `PendingCicTriggerEventCount`: Number of PendingCicTrigger events.
- `CxpTriggerAckEventCount`: Number of CxpTriggerAck events.
- `CxpTriggerResendEventCount`: Number of CxpTriggerResend events.
- `TriggerEventCount`: Number of Trigger events.
- `StartOfCameraReadoutEventCount`: Number of StartOfCameraReadout events.
- `EndOfCameraReadoutEventCount`: Number of EndOfCameraReadout events.
- `StartOfScanEventCount`: Number of StartOfScan events.
- `EndOfScanEventCount`: Number of EndOfScan events.
- `RejectedFrameEventCount`: Number of RejectedFrame events.
- `RejectedScanEventCount`: Number of RejectedScan events.

EventNotificationContext2

Data Stream	Root > EventControl	Enumeration (RW)

EventNotificationContext2

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_2 (context information value is latched when the event occurs).

Possible values: Same value set as for `EventNotificationContext1`

EventNotificationContext3

Data Stream	Root > EventControl	Enumeration (RW)

Select context information reported in EVENT_DATA_CUSTOM_CONTEXT_3 (context information value is latched when the event occurs).

Possible values: Same value set as for `EventNotificationContext1`

EventCount

Data Stream	Root > EventControl	Integer (RW)

Number of occurrences of the selected event (32-bit counter).

EventCountReset

Data Stream	Root > EventControl	Command (WO)

Reset the selected event count.

OEM Safety Key

User-configurable safety key programming and checking capability.

The OEM Safety Key capability allows the application to:

- Program an "OEM safety key" in the non-volatile memory of the Coaxlink card.
- Retrieve the encrypted version of the OEM safety key just programmed.
- Check a key against the programmed OEM safety key or its encrypted version.

OEM Safety Key

The OEM Safety Key is an application-defined string of characters. Any character except the null character is allowed. The string length is unlimited.

Key Programming

When the application sets the `ProgramOemSafetyKey` GenICam feature with the OEM Safety Key value, the Coaxlink Driver computes an encrypted version of the OEM Safety Key and stores it in the non-volatile memory of the Coaxlink card.

The encrypted value can be retrieved by getting the value of `EncryptedOemSafetyKey` immediately after having set `ProgramOemSafetyKey`.

Important: *Only the same application process having set `ProgramOemSafetyKey` is allowed to retrieve the encrypted value. This is only allowed until any other GenICam feature is set.*

Key Checking

The application has to select one OemSafetyKeyVerification value of the

In order to verify the OEM Safety Key of a Coaxlink card, the application sets a "challenge" value to the `CheckOemSafetyKey[selector]` feature.

When the [selector] argument is set to `EncryptedKey`, the set action terminates normally only when the challenge string is identical to the encrypted OEM Safety Key string.

When the [selector] argument is set to `ProgrammingKey`, the set action terminates normally only when the challenge string is identical to the programming OEM Safety Key string.

When the [selector] argument is set to `ProgrammingKeyOrEncryptedKey`, or omitted, the set action terminates normally only when the challenge string is identical to the original OEM Safety Key string or to the encrypted OEM Safety Key string.

Euresys recommends using the `EncryptedKey` selector. This improves the security level since the programming key doesn't need to appear anywhere in the end user application. Having only the encrypted key, the end user cannot retrieve the original programming key.

OEM Safety Key GenICam Features

OemSafetyKeyVerification		
Interface	Root > OEMSafetyKey	

OemSafetyKeyVerification
Defines which key can be compared with the programmed OEM safety key. **Possible values:** • `ProgrammingKey`: Only the key written to `ProgramOemSafetyKey` can be used to verify the OEM safety key. • `EncryptedKey`: Only the key read from `EncryptedOemSafetyKey` can be used to verify the OEM safety key. **Recommended value.** • `ProgrammingKeyOrEncryptedKey`: Both the key written to `ProgramOemSafetyKey` and the key read from `EncryptedOemSafetyKey` can be used to verify the OEM safety key. **Default value.** Acts as a selector for `CheckOemSafetyKey`.

CheckOemSafetyKey		
Interface	Root > OEMSafetyKey	String (WO)
Write-only string to use for comparing a key (the key written to `ProgramOemSafetyKey` or the key read from `EncryptedOemSafetyKey`) and the programmed OEM safety key.		

ProgramOemSafetyKey		
Interface	Root > OEMSafetyKey	String (WO)
Write-only string to use for programming the non-volatile OEM safety key.		

EncryptedOemSafetyKey		
Interface	Root > OEMSafetyKey	String (RO)
Read-only string that contains the encrypted version of the OEM safety key just programmed with ProgramOemSafetyKey.		

Using OEMSafetyKey

How to program and verify the OEM Safety Key.

Programming Step – Option A

Using GenICam Browser:

- Go to the GenApi tab of the interface module.
- Write a secret key to `ProgramOemSafetyKey`
- Copy the value of `EncryptedOemSafetyKey` and paste it somewhere appropriate.

Note: *There is a direct relationship between the programming key and the encrypted key. A given programming key will always lead to the same encrypted key, even on different computers or with different Coaxlink cards. This makes it possible to read the encrypted key once and hard-code this value in the application that must be protected by the OEM safety key.*

Programming Step – Option B

Using a custom application:

1. Program the OEM safety key of the Coaxlink card by writing a secret key to ProgramOemSafetyKey.

2. Read back the encrypted key by reading EncryptedOemSafetyKey. Write this value somewhere appropriate.

```
grabber.set<InterfacePort>("ProgramOemSafetyKey","plain-text key");                // 1
std::stringencryptedKey=grabber.get<InterfacePort>("EncryptedOemSafetyKey");        // 2
```

Verification Step

In the application that must be protected by the OEM key:

```
InterfacePort>("CheckOemSafetyKey[EncryptedKey]", "encrypted key retrieved in the programming step");
```

Note: *Even if the encrypted key is discovered and an attacker uses it to reprogram cards, the above verification will fail.*