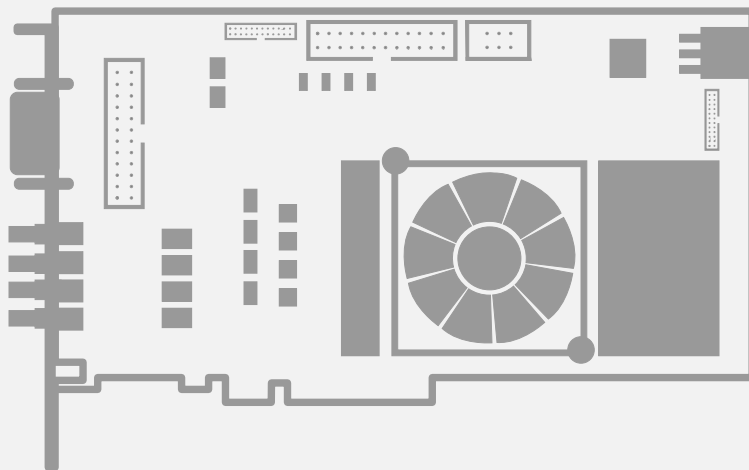


eGrabber

FrameGrabbers Handbook

- 28768 | PC1633-T Coaxlink Quad G3
- 28776 | PC3602-T Coaxlink Octo
- 28781 | PC3621-LH-T Coaxlink Mono CXP-12-LH
- 28782 | PC3622-T Coaxlink Duo CXP-12
- 28783 | PC3623-T Coaxlink Quad CXP-12 Value



Contact us

Website, email

General

www.alliedvision.com/en/contact

info@alliedvision.com

Distribution partners

www.alliedvision.com/en/avt-locations/avt-distributors

Support

www.alliedvision.com/en/support

www.alliedvision.com/en/about-us/contact-us/technical-support-repair/-/rma

Offices

Europe, Middle East, and Africa (Headquarters)

Allied Vision Technologies GmbH

Taschenweg 2a

07646 Stadtroda, Germany

T// +49 36428 677-0 (Reception)

T// +49 36428 677-230 (Sales)

F// +49 36428 677-28

Asia-Pacific | China

Allied Vision Technologies Shanghai Co Ltd.

B-510, Venture International Business Park

2679 Hechuan Road

Minhang District, Shanghai 201103

People's Republic of China

T// +86 21 64861133

Singapore

Allied Vision Technologies Asia Pte. Ltd

82 Playfair Rd, #07-01 D'Lithium

Singapore 368001

T// +65 6634 9027

North, Central, and South America, Canada

Allied Vision Technologies Canada Inc.

300 – 4621 Canada Way

Burnaby, BC V5G 4X8, Canada

T// +1 604 875 8855

USA

Allied Vision Technologies, Inc.

102 Pickering Way - Suite 502

Exton, PA 19341, USA

Toll-free// +1-877-USA-1394

T// +1 978 225 2030

Japan

Allied Vision Technologies

Yokohama Portside Bldg. 10F

8-1 Sakae-cho, Kanagawa-ku

Yokohama-shi, Kanagawa, 221-0052

T// +81 (0) 45 577 9527

This documentation is provided with **eGrabber 25.07.1** (doc build 2202).

<https://www.alliedvision.com>

This documentation is subject to the General Terms and Conditions stated on the website of **Allied Vision Technologies GmbH** and available on <https://www.alliedvision.com/en/information/terms-conditions/>.

Contents

PART I : GETTING STARTED	8
1. Hardware Setup	9
1.1. Precautions for Use of Board Products	9
1.2. PCI Express Card Slot Requirements	10
1.3. PCI Express Card Installation Procedure	11
1.4. Low-Profile Bracket Installation	11
2. Software Setup	12
2.1. Software Setup Procedure	12
2.2. Important Notices	12
Firmware Revisions	12
CPU Requirements	13
Image Buffer Limits	13
Notice for Windows	13
Notice for Linux	14
Notice for NVIDIA RDMA	14
Notices for macOS	15
2.3. Installing eGrabber	16
Installing eGrabber on Windows	16
Installing eGrabber on Linux	17
Installing eGrabber on macOS	17
Command-Line Installation Procedure	19
3. Managing Firmware	20
3.1. What's Firmware?	21
3.2. Firmware Manager Tools	22
3.3. Updating and Installing Firmware	24
3.4. Firmware Recovery Switch	25
4. Firmware Variants	26
PART II : FUNCTIONAL GUIDE	29
1. Architecture of eGrabber-driven frame grabbers	30
1.1. Main Elements	31
1.2. Block Diagrams	33
2. CoaXPress Host Interface	37
2.1. CoaXPress Interface Specifications	38
2.2. Host Connections Maps for Coaxlink Series	41
2.3. CoaXPress Link Configuration	51
2.4. Power Over CoaXPress	52
2.5. CoaXPress I/O Channel	55
2.6. CoaXPress Host To Device Trigger	56
2.7. Advanced Trigger Transmitter Settings	58
2.8. Camera Trigger Jitter Compensation	60
2.9. Trigger Delay Model	63
2.10. CoaXPress LED Lamps	66
2.11. Connection Test	68
2.12. CoaXPress 2.0 Error Counters	69
2.13. CoaXPress Link Validation Tool	71
2.14. Multi-tap CoaXPress Cameras	77
3. On-board Memory	78
3.1. Partition Schemes	79

4. Acquisition Gate	87
5. Area-scan Acquisition	88
5.1. Area-scan Acquisition Principles	89
5.2. High Frame Rate Acquisition	90
5.3. Multi-Stream Acquisition	92
6. Line-scan Acquisition	93
6.1. Line-scan Acquisition Principles	94
6.2. Line-scan Acquisition Use cases	98
6.3. Metadata Insertion	102
6.4. Trigger to Line Latency Compensation	106
7. Pixel Processing	107
7.1. Overview	107
7.2. Configurations	110
7.3. Pixel Unpacking and Alignment	118
7.4. Flat Field Correction	120
What is Flat Field Correction?	121
FFC IP Core Description	124
FFC Wizard Sample Program	127
7.5. Lookup Table Processing	133
Introduction to LUT Processing	134
Monochrome Lookup Table Processing	135
LUT Content Definition	136
LUT Setup Procedure	143
7.6. Bayer CFA Decoding	146
Bayer CFA Decoding Methods	147
Requirements and Performances	149
Using Bayer CFA Decoder	151
Advanced and Legacy Methods	153
7.7. Pixel Binning	157
Binning Configurations	158
Specifications	160
Limitations	161
7.8. Pixel Components Swapping	163
7.9. Endianness Conversion	164
7.10. Pixel Ordering	165
8. Image Data Transfer	166
8.1. Buffer Filling Rules	167
8.2. Image Width Increment Step	169
8.3. Image Data Padding	171
8.4. Horizontal Image Flipping	173
8.5. Vertical Image Flipping	174
8.6. Image Data Unscrambling	175
8.7. Transmission Methods of 1X_2YE Images (Coaxlink series)	180
8.8. Data Stream Statistics	182
8.9. Data Transfer Rate Test Program	184
9. Camera and Illumination Control	187
9.1. Camera Control Principles	188
Camera Cycle	189
Camera Cycle Concatenation Rules	190
Camera Control Methods	192
9.2. Illumination Control Principles	195
Illumination Devices	196
Aligning Camera and Illumination Cycles	197
9.3. Camera and Illumination Controller	198
Camera and Illumination Controller Overview	198

Cycle Timing Machine	199
Multiple Cycle Timings	201
Cycle Manager	202
Cycle Trigger Manager	203
Sequence Manager	205
CIC Output Signals Routing	207
9.4. CIC Timing Diagrams	208
Single Cycle	209
Overlapping Cycles - Single timing	211
Multiple Timings Example	216
Cycle Sequence Timing Diagrams	218
10. General Purpose I/O	220
10.1. I/O Lines Overview	221
10.2. I/O Lines Usage	224
10.3. I/O Control Blocks	225
10.4. Line Format and Line Mode Controls	227
10.5. Line Polarity Control	230
10.6. Line Filter Control	231
10.7. Line Source Selection	232
10.8. Line Source Divider	233
10.9. Logical I/O Line State	234
10.10. Physical I/O Line State	235
10.11. Line Driver Physical Output States	236
10.12. Initial States	237
11. I/O Toolbox	238
11.1. Introducing the I/O Toolbox	239
11.2. I/O Toolbox Composition Tables	241
11.3. Line Input Tool	244
11.4. Quadrature Decoder Tool	245
11.5. Device Link Trigger Tool	248
11.6. User Actions Tool	249
11.7. C2C-Link Synchronization Tool	254
11.8. Delay Tool	256
11.9. Divider Tool	258
11.10. Multiplier/Divider Tool	259
12. Event Signaling And Counting	262
12.1. Introduction	263
12.2. Custom Event Sources	266
EVENT_CUSTOM_CIC	267
EVENT_CUSTOM_CXP_DEVICE	268
EVENT_CUSTOM_CXP_INTERFACE	268
EVENT_CUSTOM_DATASTREAM	270
EVENT_CUSTOM_DEVICE_ERROR	271
EVENT_CUSTOM_IO_TOOLBOX	272
12.3. Event Specific Context Data	274
12.4. About GenTL Signaling	276
13. Advanced Features	277
13.1. C2C-Link	278
C2C-Link Interconnections	279
C2C-Link Electrical Specification	280
Trigger Propagation Delays	282
Cycle Trigger Synchronization	284
C2C-Link Setup Procedure	287
13.2. OEM Safety Key	289
Introducing OEM Safety Key	290

Using OEMSafetyKey	291
PART III : HARDWARE MANUAL	292
1. Mechanical Specification	293
1.1. Product Layouts	294
PC1633-T Coaxlink Quad G3	295
PC3602-T Coaxlink Octo	296
PC3621-LH-T Coaxlink Mono CXP-12-LH	297
PC3622-T Coaxlink Duo CXP-12	298
PC3623-T Coaxlink Quad CXP-12 Value	299
1.2. Camera and Data Output Connectors	300
CoaXPress DIN 4 Connector	301
CoaXPress DIN 8 Connector	302
CoaXPress Micro-BNC 1 Connector	303
CoaXPress Micro-BNC 2 Connector	304
CoaXPress Micro-BNC 4 Connector	305
1.3. GPIO Connectors	306
External I/O Connector	307
External I/O 15-pin Connector	309
Internal I/O 1 Connector	310
Internal I/O 2 Connector	312
1.4. Other Connectors	314
I/O Extension Connector	315
C2C-Link Connector	317
Auxiliary Power Input Connector for PoCXP and GPIO	318
Auxiliary Power Input Connector w/o SenseIN	319
1.5. LEDs and Switches	320
CoaXPress LED Lamps	321
Board Status LED	323
FPGA Status LED	324
Firmware Recovery Switch	325
1.6. Physical Characteristics	326
2. Electrical Specification	327
2.1. Camera Interfaces	328
CoaXPress CXP-6 Host Interface	329
CoaXPress CXP-12 Host Interface	331
2.2. PCI Express Interfaces	333
4-lane Rev 3.0 PCIe end-point	334
8-lane Rev 3.0 PCIe end-point	335
2.3. Power	336
Power Distribution Schemes	337
PC1633-T Coaxlink Quad G3	338
PC3602-T Coaxlink Octo	339
PC3621-LH-T Coaxlink Mono CXP-12-LH	341
PC3622-T Coaxlink Duo CXP-12	342
PC3623-T Coaxlink Quad CXP-12 Value	343
Main Power Input Requirements	344
Auxiliary Power Input	345
I/O Power Output	347
PoCXP Power Output Specifications	349
2.4. I/O Interfaces	351
Differential Input (Version 1)	352
Differential Input (Version 2)	354
TTL Input/Output (Version 1)	356
TTL Input/Output (Version 2)	359
Isolated Input (Version 1)	362

Isolated Input (Version 2) 365

Isolated Input (Version 4) 368

Isolated Output 371

3. Environmental Specification 374

3.1. Storage Conditions 375

3.2. Operating Conditions 376

3.3. Temperature Monitor 377

3.4. Thermal Design Data 380

3.5. Compliance Statements 381

PART IV : GENAPI FEATURES 384

PART I

GETTING STARTED

1. Hardware Setup

1.1. Precautions for Use of Board Products	9
1.2. PCI Express Card Slot Requirements	10
1.3. PCI Express Card Installation Procedure	11
1.4. Low-Profile Bracket Installation	11

1.1. Precautions for Use of Board Products

Electrostatic Sensitive Device Boards may be damaged by electrostatic discharges. Follow the procedure hereby described and apply any general procedure aimed at reducing the risk associated with electrostatic discharge. Damage caused by improper handling is not covered by the manufacturer's warranty.

Electromagnetic Compatibility AVT boards are compliant with electromagnetic compatibility regulatory requirements. To ensure this compliance, the card bracket must be secured with the relevant screw in accordance with the procedure described herein.

Risk of Electrical Shock Do not operate the computer with any enclosure cover removed. During the hardware installation, ensure the AC power cord is unplugged before touching any internal part of the computer.

Risk of Burn Do not touch an operating board. Allow board to cool before handling.

Heating Device It is normal for a board to dissipate some heat during operation. All enclosure covers, including blank brackets, must be fitted correctly to ensure that the fan cools the computer adequately.

Hot Plugging Forbidden Uncontrolled plugging and unplugging of equipment may damage a board. Always switch off the computer and any relevant system device when connecting or disconnecting a cable at the frame grabber or auxiliary board bracket. Failure to do so may damage the card and will void the warranty.

Poor Grounding Protection The computer and the camera can be located in distant areas with individual ground connections. Poor ground interconnection, ground loop or ground fault may induce unwanted voltage between equipment, causing excessive current in the interconnecting cables. This faulty situation can damage the frame grabber or the camera electrical interface. The user must follow proper equipment grounding practices at all ends of the interconnecting cables. In addition, the use of cable assemblies with overall shield solidly connected to the conductive shell of all connectors is recommended. Besides the beneficial effect of cable shielding on electromagnetic compatibility, the shield connection can increase the protection level against grounding problems by temporarily absorbing unwanted fault current.

1.2. PCI Express Card Slot Requirements

Data transfer performance

- Products belonging to the **PCI_3x4** group must be plugged into a x4, x8 or x16 PCI Express Gen 3 card connector providing at least four active lanes.
- Products belonging to the **PCI_3x8** group must be plugged into a x8 or x16 PCI Express Gen 3 card connector providing at least eight active lanes.

Powering capability

- Products belonging to the **25W** group must be plugged into a 25 W PCI Express slot

Cooling

To guarantee reliable operation across the entire operating temperature range and longer lifetime, ensure an adequate cooling of the card:

- The cooling is improved by a higher air flow circulating around the board. This air flow is increased, for example, by using computer case fans.
- Avoid placing a card next to other heat dissipating boards.

1.3. PCI Express Card Installation Procedure

1. Switch off the computer and all connected peripherals (monitor, printer...).
2. Discharge any static electricity that could be accumulated by your body. You can achieve this by touching an unpainted metal part of the enclosure of your computer with a bare hand. Make sure that the computer is linked to the AC power outlet with proper earth connection.
3. Disconnect all cables from your computer, including AC power.
4. Open the computer enclosure, according to the manufacturer instructions, to gain access to the PCI Express slots. Locate an available and adequate PCI Express slot.
5. Remove the blank bracket associated with this location. To achieve this, remove the securing screw and keep it aside for later use in the procedure. Keep the blank bracket in a known place for possible re-use.
6. Unwrap the card packing, take the board and carefully hold it. Avoid any contact of the board with unnecessary items, including your clothes.
7. Gently insert the card into the selected PCIe slot, taking care to push it down fully into the slot. If you experience some resistance, remove the board and repeat the operation. You should attempt to make a perfect board-to-slot mechanical alignment for best results. Ensure that the lower part of the bracket is inserted into the corresponding enclosure fastening.
8. Secure the board with the saved screw.
9. **Optional.** When the camera(s) is (are) powered through the camera cable (PoCXP or PoCL) or when the +12 V power output is required on any System I/O connector, connect a 12 V power source to the Auxiliary Power Input connector using an 8-pin PEG cable terminated with a 2x3 plug + 2x1 plug (mandatory when total AUX power exceeds 75 W) or a 6-pin cable.
10. **Optional.** Establish the connections with the Internal GPIO connector(s) as required by the application.
11. **Optional.** When synchronized acquisition is required for cameras attached to different cards, establish the card-to-card link interconnections.
12. Close the computer enclosure according to the manufacturer instructions.

1.4. Low-Profile Bracket Installation

Products belonging to the **LP** group can also be installed in a low-profile computer. Therefore:

1. Remove the original standard-profile bracket by unscrewing the screw locks
2. Install the low-profile bracket and secure it on the board with the screw locks

2. Software Setup

2.1. Software Setup Procedure	12
2.2. Important Notices	12
2.3. Installing eGrabber	16

2.1. Software Setup Procedure

Prior to use the board, it is necessary to install the driver and update or install the firmware.

- The **eGrabber** driver is available from the *Software Downloads* page of the AVT website:
<https://www.alliedvision.com/en/support/software-downloads/>

2.2. Important Notices

Important notifications to be read before installing and/or using the product on your PC!

Firmware Revisions	12
CPU Requirements	13
Image Buffer Limits	13
Notice for Windows	13
Notice for Linux	14
Notice for NVIDIA RDMA	14
Notices for macOS	15

Firmware Revisions



WARNING

eGrabber driver checks the compatibility of the firmware installed on every frame grabber. For those having an incompatible firmware, the GenTL driver exposes 0 (zero) Device.

If the requirement is not satisfied for all the **eGrabber-driven frame grabbers** in your system, it is *mandatory* to apply the Firmware Upgrade procedure prior to using this version of the driver.



NOTE

Starting with eGrabber 25.03, the firmware revision numbers are listed, for each firmware-variant in the [Firmware Variants table](#)

CPU Requirements

The image converter requires a CPU that has the Supplemental Streaming SIMD Extension 3 (SSSE3) instruction set.

Image Buffer Limits

Maximum buffer size

0xffffffff bytes (4 GiB - 16 B) for all operating systems

Number of buffers

The number of buffers is only limited by available system resources.

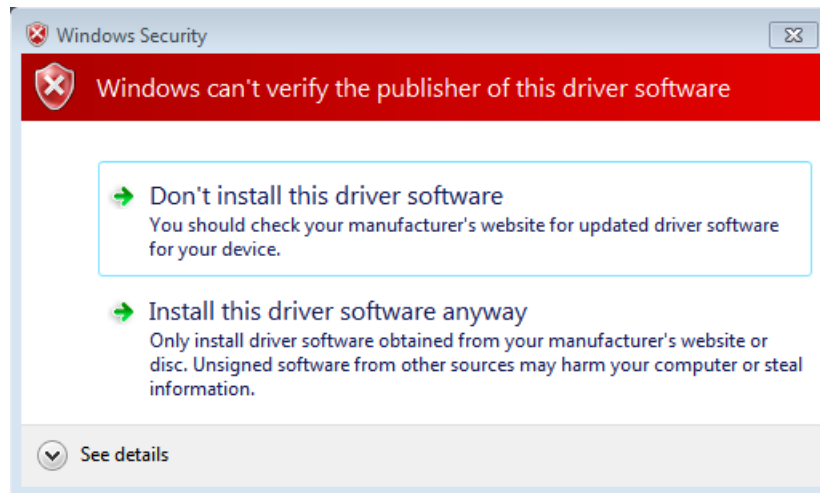
NOTE: when using very large numbers of buffers, DSAnnounceBuffer calls can take longer and longer to complete (or even fail with error code GC_ERR_CUSTOM_IOCTL_BUFFER_ANNOUNCE_FAILED). If this happens, the user should set `DmaEngineOptimization=LowMemoryUsage` in the data stream module.

Notice for Windows

Important notifications to be read before installing and/or using the product on your Windows PC

Missing time-stamping certificate

The following Windows Security warning message may occur at driver installation on Microsoft Windows:



This Windows security warning occurs when the **GlobalSign Root CA - R6** certificate is missing from the Windows certificate store.

This issue can be solved by installing this missing certificate which can be downloaded [here](#) on the GlobalSign website then installed in the Trusted Root Certification Authorities (local computer) certificate store.

Notice for Linux

Important notification to be read before installing and/or using the product on your Linux PC

- **Memento** must be installed prior to **eGrabber**.
- If the **eGrabber** package is already installed, proceed as follows:
 - Uninstall **eGrabber**.
 - Install **Memento**.
 - Re-install **eGrabber**.
- If NVIDIA RDMA is required, read "[Notice for NVIDIA RDMA](#)" on page 14

Notice for NVIDIA RDMA

NVIDIA RDMA is only supported on Linux.

NVIDIA RDMA samples require a NVIDIA GPU that supports RDMA.

The NVIDIA RDMA samples allocate memory on the GPU and announce this memory using `NvidiaRdmaMemory`.

See the following files in the eGrabber sample programs:

- `cpp/egrabber/samples/503-grabn-cuda-rdma-process.*`
- `cpp/nvidia/egrabber-cuda` with the command line argument `cudaRDMA`

Installation instructions:

- NVIDIA CUDA drivers:
 - Follow the installation instructions from: https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=20.04&target_type=deb_network
- NVIDIA driver sources:
 - These are needed to produce the Module.symvers file associated with the installed nvidia driver. This file will be required to install the eGrabber package
 - Select the appropriate driver from <https://www.nvidia.com/download/index.aspx?lang=en-us>
 - Make sure to download the version that matches the nvidia-<version> already installed in /usr/src/
 - Extract the archive with -x option
 - Change to directory kernel in the extracted archive
 - Run make module
 - The file Module.symvers should have been generated
- eGrabber package
 - Extract the egrabber-linux-x86_64 archive
 - Install the package with the following command:
`sudo NVIDIA_KERNEL_PATH=<dir path containing Module.symvers> ./install.sh`

The line Enabling NVIDIA RDMA build should appear during the installation of the eGrabber package.

A successful build can be confirmed if the command `lsmod | grep coaxlink` (or `lsmod | grep grablink`) indicates that coaxlink (or grablink) module depends on nvidia.

Notices for macOS

Important notifications to be read before installing the driver on your Mac

Driver types

Install the **Memento** package corresponding to the **eGrabber** driver type:

eGrabber driver package	Memento package
egrabber-macos-aarch64-dext-<MA.MI.RE.BU>.pkg	memento-macos-aarch64-dext-<MA.MI.RE.BU>.pkg
egrabber-macos-aarch64-kext-<MA.MI.RE.BU>.pkg	memento-macos-aarch64-kext-<MA.MI.RE.BU>.pkg
egrabber-macos-x86_64-kext-<MA.MI.RE.BU>.pkg	memento-macos-x86_64-kext-<MA.MI.RE.BU>.pkg



TIP

dext drivers operate in user-mode using the default Full Security policy level. It is not necessary to change the security setting.

Reduced Security level (only for kext drivers on Mac computers with Apple silicon)

Kernel extensions must be explicitly enabled before the installation of **AVT -aarch64-kext-** packages on Mac computers with Apple silicon.

See <https://support.apple.com/fr-be/guide/security/sec8e454101b/web>

To enable kernel extensions on a Mac with Apple silicon:

1. Enter macOS recovery
2. In **Utilities > Startup Security Utility > Security Policy**
 - a. Select **Reduced Security**
 - b. Check **Allow user management of kernel extensions from identified developers**
3. Restart the system

Step 3. Approval of kernel extension (only for kext drivers on Mac computers with Apple silicon)

After installing **eGrabber** or **Memento AVT-aarch64-kext** packages, newly installed **AVT** kernel extensions must be approved by the administrator in the **Security and Privacy preferences** and the system needs to be restarted.

2.3. Installing eGrabber

Installing eGrabber on Windows	16
Installing eGrabber on Linux	17
Installing eGrabber on macOS	17
Command-Line Installation Procedure	19

Installing eGrabber on Windows

1. Read the "Notice for Windows" on page 13
1. Open the *Software Downloads* page of the AVT website:
<https://www.alliedvision.com/en/support/software-downloads/>
2. Expand the *****TBD***** section to display the file list corresponding to the latest available **eGrabber** release.
3. Select the **egrabber-win-x86_64-25.07.1.*.exe** setup file.
4. Launch the installer tool to install the driver files and software tools on your PC.

NOTE: If you have an existing **eGrabber** installation, the installer tool prompts you to uninstall it before being able to continue. Otherwise, it prompts you for the selection of the destination folder.

Installing eGrabber on Linux

1. Read the "Notice for Linux" on page 14
1. Open the *Software Downloads* page of the AVT website:
<https://www.alliedvision.com/en/support/software-downloads/>
2. Expand the ***TBD*** section to display the file list corresponding to the latest available **eGrabber** release.
3. Download the setup file according to the processor architecture:
 - For an installation on AArch64 (64-bit) processor architecture, select the egrabber-linux-aarch64-25.07.1*.tar.gz setup file.
 - For an installation on x86_64 (64-bit) processor architecture, select the egrabber-linux-x86_64-25.07.1*.tar.gz setup file.
4. Extract the egrabber-linux-<ARCH>-<VERSION>.tar.gz archive, e.g., with `tar -xaf egrabber-linux-<ARCH>-<VERSION>.tar.gz`
5. Run the installation script: `sudo ./egrabber-linux-<ARCH>-<VERSION>/install.sh`



NOTE

- If you have an existing **eGrabber** installation, the installer tool prompts you to uninstall it before being able to continue. Otherwise, it prompts you for the selection of the destination folder.
- After installation, `sudo /opt/euresys/egrabber/shell/check-install.sh` can be executed to check the eGrabber installation and troubleshoot issues.

Installing eGrabber on macOS

1. Read the "Notices for macOS" on page 15.
1. Open the *Software Downloads* page of the AVT website:
<https://www.alliedvision.com/en/support/software-downloads/>
2. Expand the ***TBD*** section to display the file list corresponding to the latest available **eGrabber** release.
3. Execute the installation procedure according to the processor architecture and the driver type:

Installing dext packages on Mac computers with Apple silicon

1. Download egrabber-macos-aarch64-dext-25.07.1.*.pkg
2. After package files have been downloaded with *Safari*, the usual *double-click* to launch the installer will not let you install the package. You shall use instead *control+click* and select *Open* to launch the installer. A window will pop up, click then on *Open* to proceed.

The extension installer applications will be launched automatically and will be waiting for the administrator to approve the newly installed **AVT** extensions in the [Security and Privacy preferences](#).

3. Launch the installer tool to install the driver files and software tools on your PC.

Installing kext drivers packages on Mac computers with Apple silicon

Step 1. Enable kernel extensions

Kernel extensions must be explicitly enabled before the installation of **AVT** -aarch64-kext-packages on Mac computers with Apple silicon.

See <https://support.apple.com/fr-be/guide/security/sec8e454101b/web>

To enable kernel extensions on a Mac with Apple silicon:

1. Enter macOS recovery
2. In [Utilities > Startup Security Utility > Security Policy](#)
 - a. Select [Reduced Security](#)
 - b. Check [Allow user management of kernel extensions from identified developers](#)
3. Restart the system

Step 2 Launch the installer

1. Download egrabber-macos-aarch64-kext-25.07.1.*.pkg
2. After package files have been downloaded with *Safari*, the usual *double-click* to launch the installer will not let you install the package. You shall use instead *control+click* and select *Open* to launch the installer. A window will pop up, click then on *Open* to proceed.

Step 3. Approve kernel extension

After installing **eGrabber** or **Memento AVT** -aarch64-kext packages, newly installed **AVT** kernel extensions must be approved by the administrator in the [Security and Privacy preferences](#) and the system needs to be restarted.

Installing kext drivers packages on Mac computers with an Intel processor

1. Download egrabber-macos-x86_64-kext-25.07.1.*.pkg
2. After package files have been downloaded with *Safari*, the usual *double-click* to launch the installer will not let you install the package. You shall use instead *control+click* and select *Open* to launch the installer. A window will pop up, click then on *Open* to proceed.

Command-Line Installation Procedure

You may want to integrate the boards drivers and eGrabber tools installation into your own application distribution.

eGrabber setup program can be called in command-line mode with your installation options. In this mode, the **eGrabber** installation program does not prompt for user action and does not display any dialog box.

Installation

To perform a command-line installation, call the setup program with the /s flag. The installation is launched in the silent mode, that is no window nor dialog box will appear.

There are a couple of optional flags:

- Use the /a flag to force the installation of all components, including optional ones which are not selected by default.
- Use the /fflag to force the removal of an already installed version before executing the setup file, even if the already installed version is newer.

Installation removal

To automatically remove installed tools, call the setup program with the /u flag.

Use the /s flag to launch the removal program in the silent mode. In this mode, no window nor dialog box will appear.

Reboot during Installation

A reboot may be required after driver installation but will not take place automatically. The reboot must be performed by your application. In this case, the [HKEY_LOCAL_MACHINE\SOFTWARE\Euresys\Common] "RebootNeeded" registry entry should be checked. If it exists and is set to 1, then it should be replaced by 0, and the system must be rebooted.

Error Reporting

After the command line installation, the following registry key is updated and holds the installation status: [HKEY_LOCAL_MACHINE\SOFTWARE\Euresys\Common\LastInstallError].

- The ErrorCode DWORD identifies the error:
 - 0 - There is no error.
 - Any other value - Please contact technical support.
- The Cause string gives a wording of the error.
- The Source string identifies the installer that caused the error.
- The ErrorTime string gives the time and date of the error.

3. Managing Firmware

3.1. What's Firmware? 21

3.2. Firmware Manager Tools 22

3.3. Updating and Installing Firmware 24

3.4. Firmware Recovery Switch 25

3.1. What's Firmware?

Firmware

In this context, "firmware" designates the content of the FPGA (Field Programmable Gate Array) device of a card.

It defines the functionality of the card including the PCI Express end-point.

Firmware EEPROM

The FPGA used on **eGrabber-driven frame grabbers** is RAM-based; it needs to be loaded every power up.

Considering that a PCI Express end point must be ready within 150 milliseconds of the power-up time, the FPGA content, 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.



TIP

The **eGrabber** driver will never modify the content of the FPGA during operation.

Firmware modifications

Any modification of the FPGA content requires a two-step operation:

1. The new firmware is written into the Flash EEPROM of the card using a firmware management tool.
2. The new firmware is activated by cycling the system power.

3.2. Firmware Manager Tools

eGrabber is delivered with two firmware management tools:

- "Firmware Manager - GUI mode" on page 22 : A graphical user interface tool in **eGrabber Studio**,
- "Firmware Manager - Command line mode" on page 22 : A command-line tool named **Firmware Manager Console**.

Firmware Manager - GUI mode

To open the **Firmware Manager** in GUI mode, select one of the following methods:

- From the Windows Start Menu: click on Firmware Manager shortcut in the **Euresys eGrabber** folder
- From the **Welcome Screen** of **eGrabber Studio**, click on the **Firmware Manager** button.

See also: "Firmware Manager (GUI mode)" section in the eGrabber Studio User Guide for a detailed description.

Firmware Manager - Command line mode

Access

The command-line tool is named `coaxlink-firmware.exe`. It is located in the firmware sub-folder of the eGrabber installation folder.

On Windows, to open the **Firmware Manager** in command-line mode, select one of the following methods:

- From the Windows Start Menu: click on Firmware Manager console shortcut in the **Euresys eGrabber** folder
- Open a command prompt and open in the `C:\Program Files\Euresys\Grabber\firmware` folder

On Linux, to open the **Firmware Manager** in command-line mode:

- Open a command shell in the `/opt/euresys/egrabber/firmware` folder

On macOS, to open the **Firmware Manager** in command-line mode:

- Open a command shell in the `/usr/local/opt/euresys/egrabber/firmware` folder.

Main commands

- Executing `coaxlink-firmware --help` displays a help message describing all the command options.
- Executing `coaxlink-firmware gui` starts the **Firmware Manager (Deprecated)** graphical user interface.

- Executing `coaxlink-firmware list` lists the properties of the firmware installed on each card present in the system.
- Executing `coaxlink-firmware update` updates the firmware.
- Executing `coaxlink-firmware install` installs a new firmware variant.

Unless specified with a `--firmware=FILE` option, the tool uses the embedded library.

3.3. Updating and Installing Firmware

**WARNING**

Prior to executing this procedure, read the "Important Notices" section of the release notes!

The **eGrabber** driver comes with all the firmware variants for all the **eGrabber-driven frame grabbers**.

1. Determine the firmware variant that fulfills the functional requirements of your application: e.g. '1-camera', '1-camera, line-scan', '2-camera'. Therefore, check the *Firmware Variants per Product* section of the release notes for the firmware variants that are applicable to your card.
2. Launch a **Firmware Manager tool** to perform a firmware *update* or to *install* a specific firmware variant on your card(s) using the **Firmware Manager** tool in *GUI mode* with **eGrabber Studio** or the **Firmware Manager Console** in *command-line mode*:
 - a. In **eGrabber Studio**, open the **Firmware Manager** pane:
 - Select the card to update
 - Select the firmware variant to install
 - Proceed with the installation
 - b. In command-line mode, to *update* a variant:
coaxlink-firmware update
 - c. In command-line mode, to *install* another firmware variant:
coaxlink-firmware install '[variant-name]'
3. Wait until completion of the firmware update

**WARNING**

Avoid turning off your PC during the firmware update procedure!

4. Repeat the procedure on all your **eGrabber-driven frame grabbers**.
5. **Power off completely your PC** and restart it to activate the newly loaded firmware.

3.4. Firmware Recovery Switch

Switch types and location

The *firmware recovery switch* is implemented with one of the following components:

- 3-pin header and a jumper
- 2-way DIP switch

See also: "Product Layouts" on page 294 in the Handbook to locate the firmware recovery switch. .

Switch positions



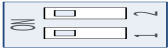
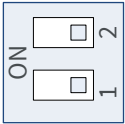
The *firmware recovery switch* has two positions:

Normal position (factory default)

At the next power ON, the latest firmware successfully written into the Flash EEPROM is used to program the FPGA. After FPGA startup completion, the card exhibits the *standard PCI ID* and the driver allows normal operation.

Recovery position

At the next power ON, the last but one firmware successfully written into the Flash EEPROM is used to program the FPGA. After FPGA startup completion, the card exhibits the *recovery PCI ID* and the driver inhibits image acquisition.

Switch type	Normal position	Recovery position
3-pin header and a jumper		
2-way DIP switch		

4. Firmware Variants

This topic shows, for each officially supported product (group), the list of officially available¹ firmware variants provided with **eGrabber 25.07**.

- The **Firmware variant** column shows the name of the firmware variant.
- The **HCMAP** column shows the **Host Connection Map**².
- The **Features** column shows the main features of the firmware variant.
- The **Description** column shows a one-phrase description of the connection scheme.
- The **Rev.** column shows the revision number of the firmware delivered with this release.

PC1633-T Coaxlink Quad G3

Firmware Variant	HCMAP	Features	Description	Rev.
1-camera	1D4	FFC LUT CFA-12	One 1- or 2- or 4-connection area-scan camera	481
1-camera, 4-data-stream	1D4S4		One 1- or 2- or 4-connection area-scan camera, up to 4 data streams	481
1-camera, line-scan	1D4	FFC LUT FLIPX	One 1- or 2- or 4-connection line-scan camera	481
2-camera	2D22	LUT CFA-35-D0	One or two 1- or 2-connection area-scan cameras	481
2-camera, bayer	2D22	CFA-35	One or two 1- or 2-connection area-scan cameras	481
2-camera, line-scan	2D22	LUT	One or two 1- or 2-connection line-scan cameras	481
3-camera	3D211	LUT	One 1- or 2-connection and one or two 1-connection area-scan cameras	481
4-camera	4D1111	LUT	One or two or three or four 1-connection area-scan cameras	481
4-camera, line-scan	4D1111	LUT	One or two or three or four 1-connection line-scan cameras	481

¹ Excluding custom firmware variants

² Specific assignment of the Device connections to the Host connectors

PC3602-T Coaxlink Octo

Firmware Variant	HCMAP	Features	Description	Rev.
1-camera	1D8	LUT CFA-123	One 1- or 2- or 4- or 8-connection area-scan camera	481
1-camera, line-scan	1D8	LUT MI	One 1- or 2- or 4- or 8-connection line-scan camera	481
2-camera	2D44	FFC LUT CFA-125	One or two 1- or 2- or 4-connection area-scan cameras	481
2-camera, line-scan	2D44	LUT FLIPX MI PLANAR	One or two 1- or 2- or 4-connection line-scan cameras	481
3-camera	3D422	LUT	One 1- or 2- or 4-connection and one or two 1- or 2-connection area-scan cameras	481
4-camera	4D2222	LUT	One or two or three or four 1- or 2-connection area-scan cameras	481
4-camera, line-scan	4D2222	LUT MI	One or two or three or four 1- or 2-connection line-scan cameras	481
5-camera	5D41111	LUT	One 1- or 2- or 4-connection and one or two or three or four 1-connection area-scan cameras	481
5-camera, 5D22211	5D22211	LUT	One or two or three 1- or 2-connection and one or two 1-connection area-scan cameras	481
8-camera	8D11111111	LUT	Up to eight 1-connection area-scan cameras	481
8-camera, line-scan	8D11111111	LUT MI	Up to eight 1-connection line-scan cameras	481

PC3621-LH-T Coaxlink Mono CXP-12-LH

Firmware Variant	HCMAP	Features	Description	Rev.
1-camera	1D1	LUT	One 1-connection area-scan camera	481
1-camera, line-scan	1D1	LUT	One 1-connection line-scan camera	481

PC3622-T Coaxlink Duo CXP-12

Firmware Variant	HCMAP	Features	Description	Rev.
1-camera	1D2	LUT CFA-3	One 1- or 2-connection area-scan camera	481
1-camera, line-scan	1D2	LUT	One 1- or 2-connection line-scan camera	481
2-camera	2D11	LUT	One or two 1-connection area-scan cameras	481
2-camera, line-scan	2D11	LUT	One or two 1-connection line-scan cameras	481

PC3623-T Coaxlink Quad CXP-12 Value

Firmware Variant	HCMAP	Features	Description	Rev.
1-camera	1D4	FFC LUT CFA-12 BIN	One 1- or 2- or 4-connection area-scan camera	481
1-camera, line-scan	1D4	FFC LUT BIN MI LT	One 1- or 2- or 4-connection line-scan camera	481
2-camera	2D22	LUT	One or two 1- or 2-connection area-scan cameras	481
2-camera, line-scan	2D22	LUT MI LT	One or two 1- or 2-connection line-scan cameras	481
3-camera	3D211	LUT	One 1- or 2-connection and one or two 1-connection area-scan cameras	481
4-camera	4D1111	LUT	One or two or three or four 1-connection area-scan cameras	481
4-camera, line-scan	4D1111	LUT MI	One or two or three or four 1-connection line-scan cameras	481

Features abbreviations

- *BIN*: Pixel binning
- *CFA-12*: Bayer CFA decoding - Methods 1 and 2
- *CFA-123*: Bayer CFA decoding - Methods 1, 2, and 3
- *CFA-125*: Bayer CFA decoding - Methods 1, 2, and 5
- *CFA-2-S0*: Bayer CFA decoding - Method 2 on Stream0
- *CFA-3*: Bayer CFA decoding - Method 3
- *CFA-35*: Bayer CFA decoding - Methods 3 and Method 5
- *CFA-35-D0*: Bayer CFA decoding - Methods 3 and 5 on Device0
- *FLIPX*: Horizontal image flipping
- *FFC*: Flat-field correction
- *JPEG-S1*: JPEG encoding on Stream1
- *LLE*: Laser line extraction
- *LT*: Mapping of events from the I/O Toolbox to CoaXPress trigger messages LinkTrigger0 and LinkTrigger1
- *LUT*: Lookup table processing
- *MI*: Metadata insertion
- *PLANAR*: RGB to PLANAR_RGB or BGR to PLANAR_BGR conversions

PART II
FUNCTIONAL GUIDE

1. Architecture of eGrabber-driven frame grabbers

1.1. Main Elements 31

1.2. Block Diagrams 33

1.1. Main Elements

Quick overview of the main functional elements of eGrabber-driven frame grabbers

GenTL hierarchy

Each functional element of a frame grabber is configured and controlled by GenApi features belonging to a GenTL module.

At the top of the hierarchy, there is one *GenTL System Module* per Host PC. It binds all the *GenTL Interface Modules* of a Host PC.

There is one *GenTL Interface Module* for each frame grabber. It binds all the *GenTL Device Modules* of a frame grabber.

There is one *GenTL Device Module* for each camera (or imaging device) attached to a frame grabber. 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*.

NOTE: The maximum number of cameras that can be attached to a frame grabber is determined by the installed firmware variant.

There is one *GenTL Data Stream Module* for each data stream delivered by a camera attached to a frame grabber. It gathers the elements involved into the image build-up and transport from the imaging device to a pool of GenTL buffers.

NOTE: The maximum number of data-stream for a camera attached to a frame grabber is determined by the installed firmware variant.

There is one *GenTL Buffer Module* for each image buffer.

Main elements of the Interface Module

General purpose I/O lines

The "General Purpose I/O" on page 220 block gathers all the I/O ports of the card.

I/O Toolbox

The "I/O Toolbox" on page 238 block gathers a collection of tools used to build event streams from trigger and encoder devices attached to the I/O port inputs.

NOTE: These elements are common to- (or can be shared by-) all the GenTL Device Modules managed by the frame grabber.

Main elements of the Device Module

Camera and illumination controller

The "Camera and Illumination Control" on page 187 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: This element is common to- (or can be shared by-) all the GenTL Data Stream Modules related to that imaging device.

Main elements of the Data Stream Module

Image acquisition controller

This block controls 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

The "Acquisition Gate" on page 87 controls the data extraction and filters out the image data that doesn't need to be acquired.

On-board memory

The Image data partitions of the "On-board Memory" on page 78 temporarily stores the raw image data together with related metadata such as image size, pixel type, time-stamp...

Pixel Processing

The "Pixel Processing" on page 107 block performs on-the-fly pixel processing and data formatting.

Image Data Transfer

The "Image Data Transfer" on page 166 block transfers the image data to the destination buffer.

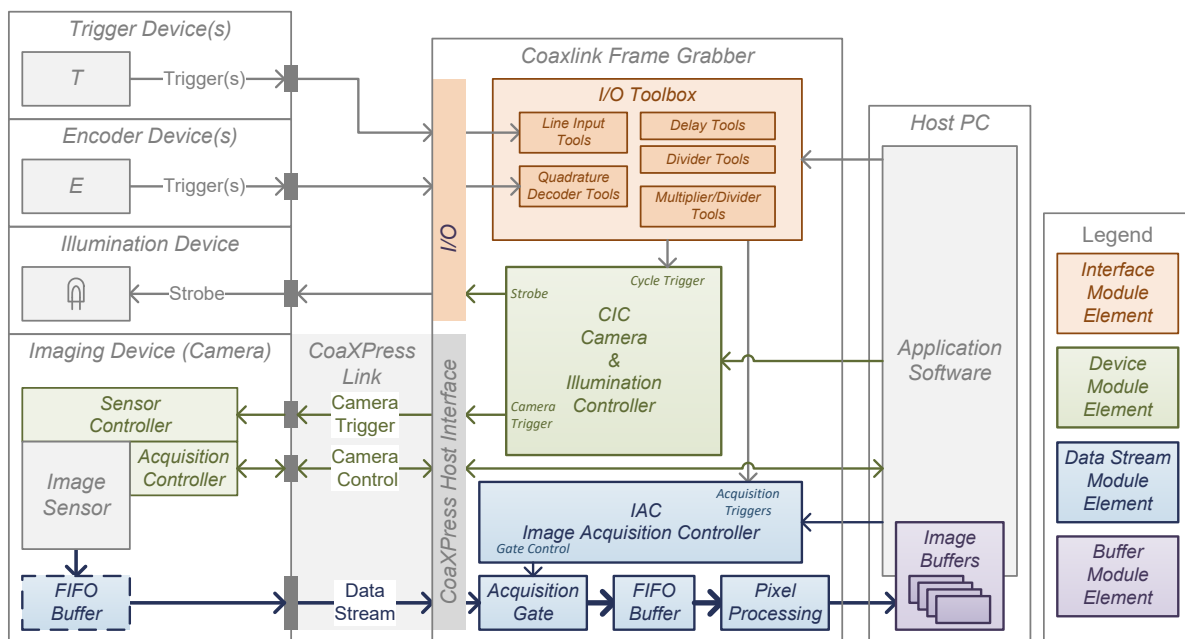
1.2. Block Diagrams



NOTE

In the diagrams hereafter, the main elements are represented by rectangles and their relations are represented by line segments with arrows indicating the direction of the signal or the data flow. The filling color of the rectangle indicates the level in the GenTL hierarchy as described in the legend.

1-camera, 1-data-stream



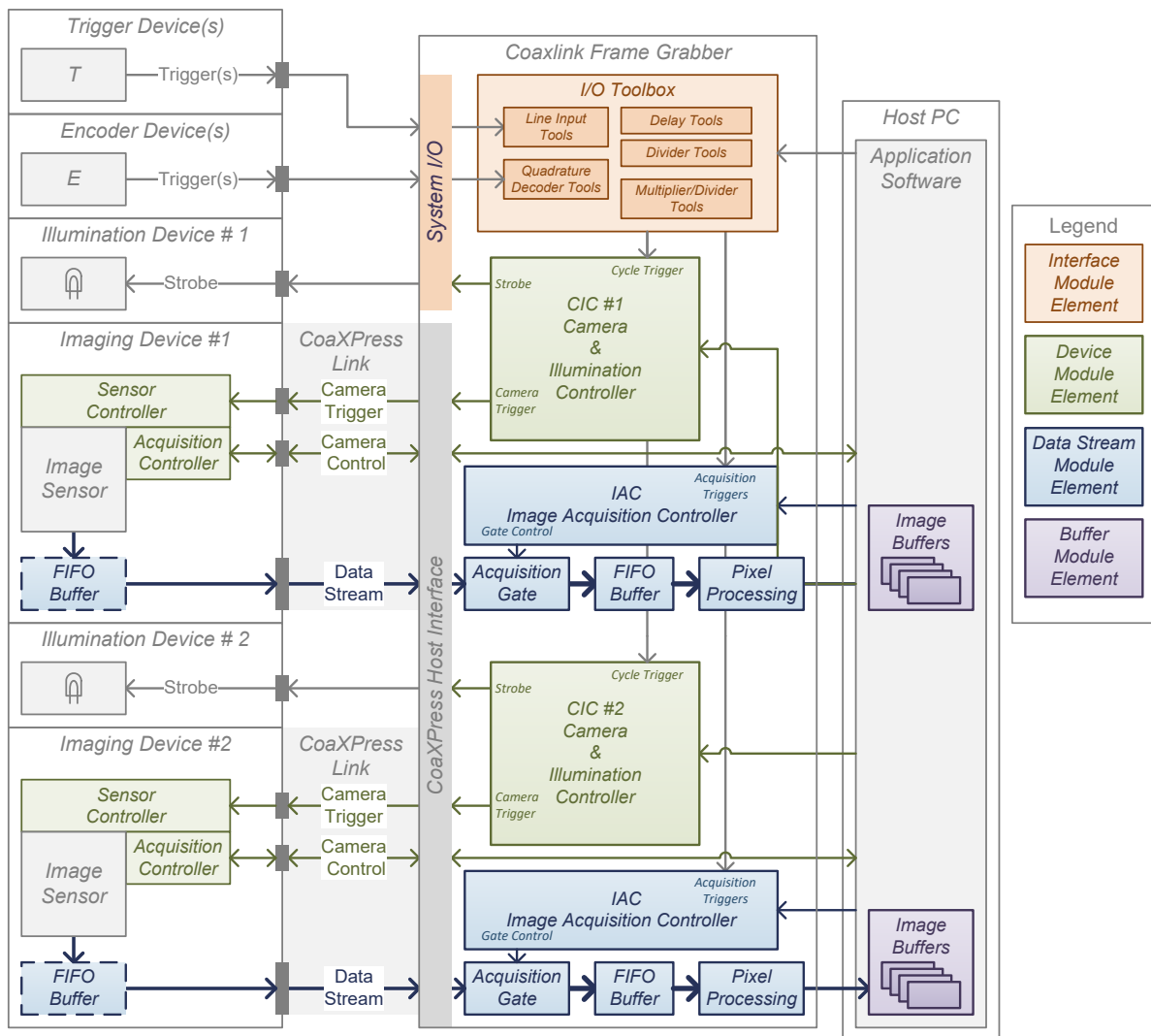
1-camera, 1-data-stream image acquisition system



NOTE

this configuration applies only when a *1-camera* or a *1-camera, line-scan* firmware variant is installed.

2-camera, 1-data-stream



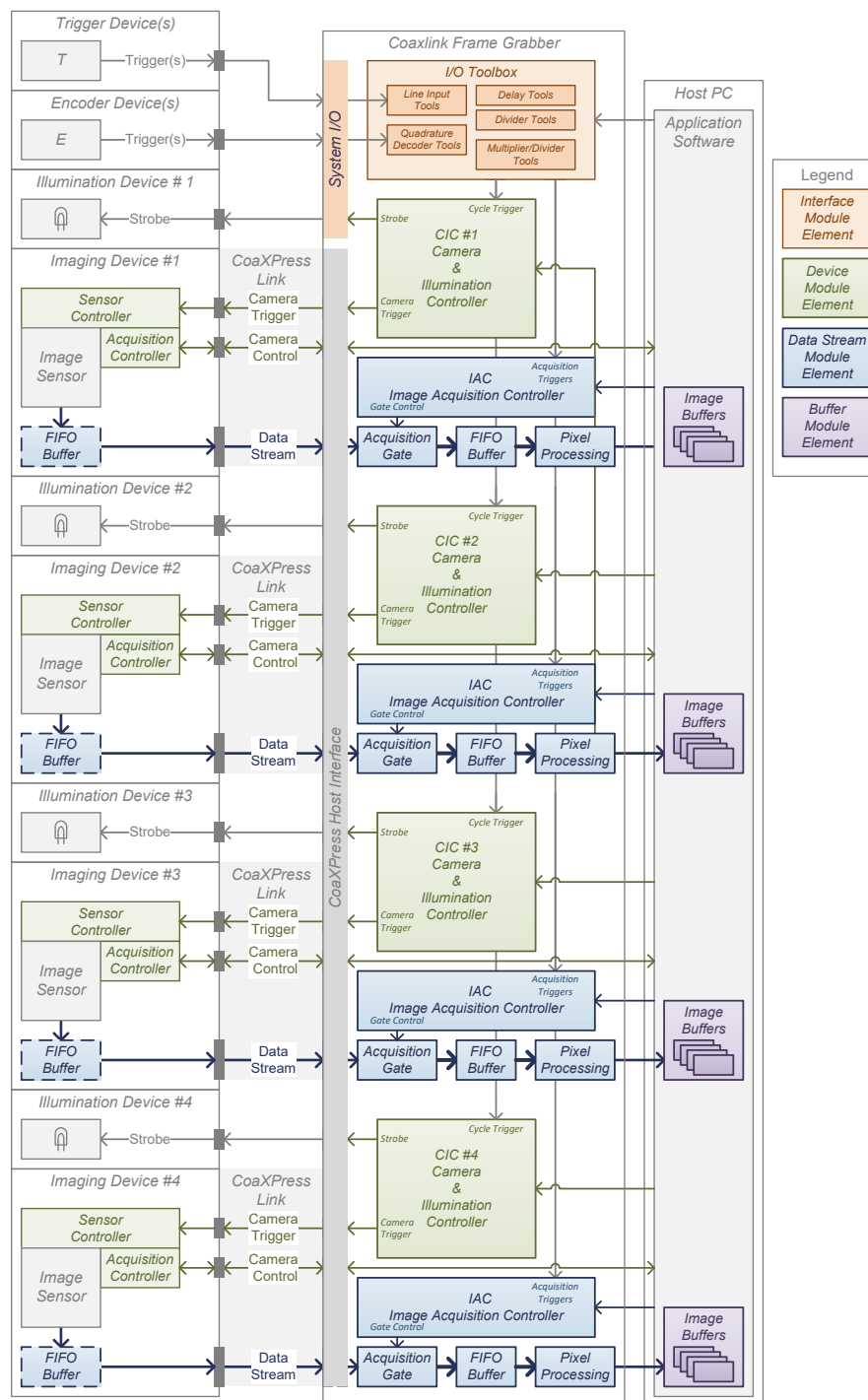
2-camera, 1-data-stream image acquisition system



NOTE

this configuration applies only when a 2-camera or a 2-camera, line-scan firmware variant is installed.

4-camera, 1-data-stream



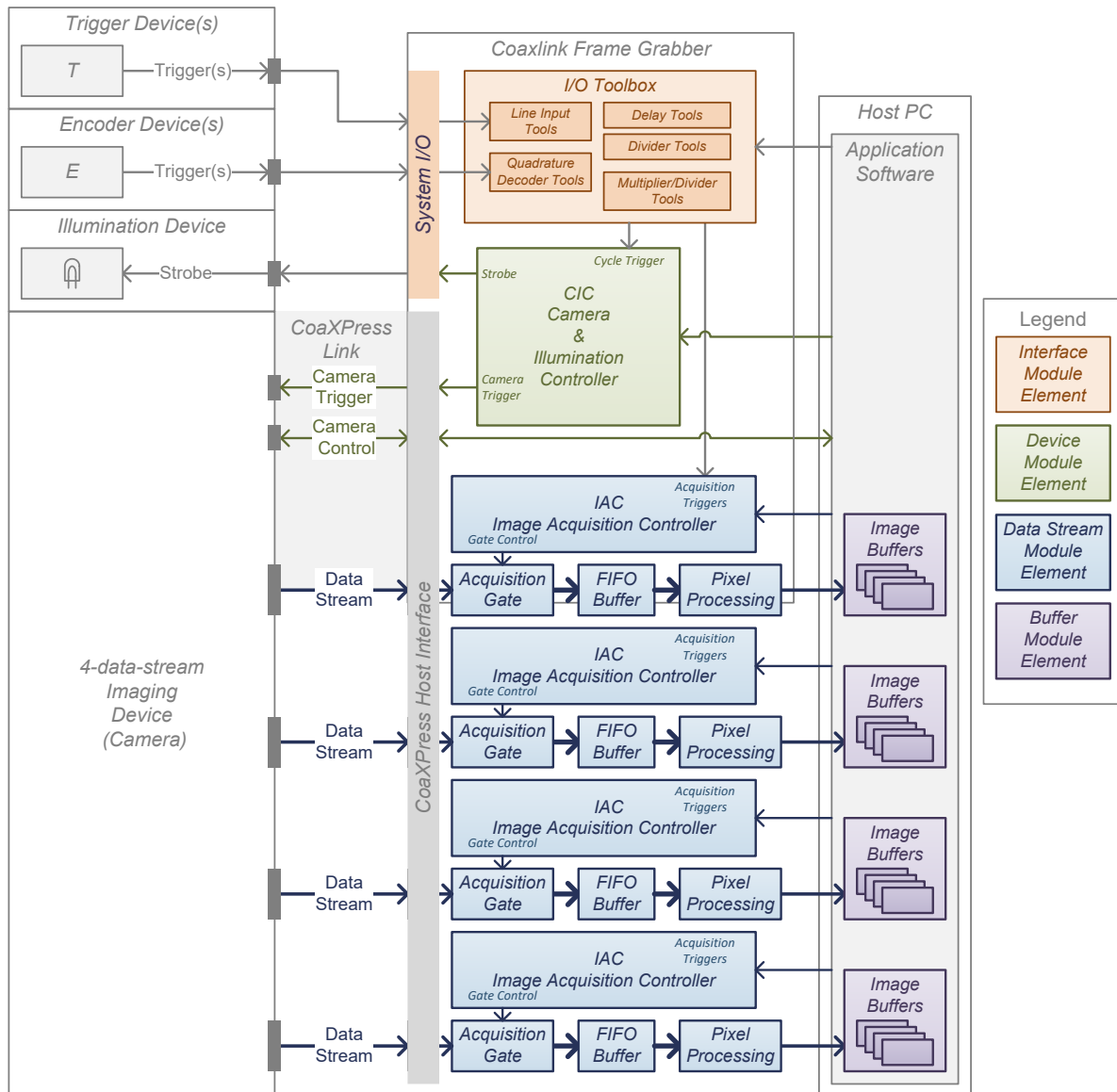
4-camera, 1-data-stream image acquisition system



NOTE

this configuration applies only when a 4-camera or a 4-camera, line-scan firmware variant is installed.

1-camera, 4-data-stream



1-camera, 4-data-stream image acquisition system



NOTE

this configuration applies only when a 1-camera,4-data-stream firmware variant is installed.

2. CoaXPress Host Interface

2.1. CoaXPress Interface Specifications	38
2.2. Host Connections Maps for Coaxlink Series	41
2.3. CoaXPress Link Configuration	51
2.4. Power Over CoaXPress	52
2.5. CoaXPress I/O Channel	55
2.6. CoaXPress Host To Device Trigger	56
2.7. Advanced Trigger Transmitter Settings	58
2.8. Camera Trigger Jitter Compensation	60
2.9. Trigger Delay Model	63
2.10. CoaXPress LED Lamps	66
2.11. Connection Test	68
2.12. CoaXPress 2.0 Error Counters	69
2.13. CoaXPress Link Validation Tool	71
2.14. Multi-tap CoaXPress Cameras	77






2.1. CoaXPress Interface Specifications

Functional specifications summary of the CoaXPress camera interface

See also: "Camera Interfaces" on page 328 in the hardware manual for electrical specifications.

Connectivity requirements

Physical medium and speed range

Medium Speed range	Availability
Coaxial (Copper) CXP-1 up to CXP-6	 
Coaxial (Copper) CXP-1 up to CXP-12	  
QSFP+ (Fiber) CXP-1 up to CXP-12	

Device count, connection count

The Host Interface of **Coaxlink frame grabbers** requires a specific assignment of the Device connections to the Host connectors. Such assignment is named *Host Connections Map*.
The Host Connections Map is hard-coded in the product firmware variant.
NOTE: The Coaxlink frame grabber and the firmware variant must be selected according to the required mapping!

See also: "Host Connections Maps for Coaxlink Series" on page 41

Data stream count


Stream count	Availability
1 up to 4	 (1-camera, 4-data-stream)
1	Other firmware variants

Image format requirements











Stream packets

Packet size	Applicable products
Up to 16,384 Bytes	All products

Supported pixel formats

Pixel Format (PFNC name)	Usage
Mono8, Mono10, Mono12, Mono14, Mono16	Monochrome cameras 8-/10-/12-/14-/16-bit per pixel
BayerGR8, BayerRG8, BayerGB8, BayerBG8 BayerGR10, BayerRG10, BayerGB10, BayerBG10 BayerGR12, BayerRG12, BayerGB12, BayerBG12 BayerGR14, BayerRG14, BayerGB14, BayerBG14 BayerGR16, BayerRG16, BayerGB16, BayerBG16	Bayer CFA color cameras 8-/10-/12-/14-/16-bit per pixel component
RGB8, RGB10, RGB12, RGB14, RGB16	Three-component RGB color cameras 8-/10-/12-/14-/16-bit per pixel component
RGBA8, RGBA10, RGBA12, RGBA14, RGBA16	Four-component RGBA color cameras 8-/10-/12-/14-/16-bit per pixel component
Raw	Undefined format

Image stream format

Stream format Acquisition	Availability
Rectangular image Area-scan	 Prelim (1-camera), (1-camera, 4-data-stream), (2-camera), (2-camera, bayer), (3-camera), (4-camera)  Prelim (1-camera), (1-camera, custom-logic), (2-camera), (3-camera), (4-camera), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera)  Prelim (1-camera)  Prelim (1-camera), (2-camera)  Prelim (1-camera), (2-camera), (3-camera), (4-camera)
Rectangular image Line-scan	 Prelim (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)  Prelim (1-camera, line-scan), (2-camera, line-scan), (2-camera, line-scan, custom-logic), (4-camera, line-scan), (8-camera, line-scan)  Prelim (1-camera, line-scan)  Prelim (1-camera, line-scan), (2-camera, line-scan)  Prelim (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)
Arbitrary image	Not supported

Supported scanning methods

Scanning Geometry (Stream)	Availability
Progressive-scan 1X_1Y	All firmware variants
Progressive-scan 1X-2YE (Dual stream)	<div><div><div>Prelim</div><div>Value 1Z</div></div>(1-camera)</div>
Progressive-scan 1X_1Y2, 1X_2YE, 1X_2YM (Single stream)	<div><div><div>Prelim</div><div>Quad 55</div></div>(1-camera), (1-camera, 4-data-stream), (2-camera), (2-camera, bayer), (3-camera), (4-camera)</div> <div><div><div>Prelim</div><div>Octo</div></div>(1-camera), (1-camera, custom-logic), (2-camera), (3-camera), (4-camera), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera)</div> <div><div><div>Prelim</div><div>Mono 1Z LH</div></div>(1-camera)</div> <div><div><div>Prelim</div><div>Duo 1Z</div></div>(1-camera), (2-camera)</div> <div><div><div>Prelim</div><div>Value 1Z</div></div>(1-camera), (2-camera), (3-camera), (4-camera)</div>
Interlaced-scan	Not supported

2.2. Host Connections Maps for Coaxlink Series


The Host Interface of **Coaxlink frame grabbers** requires a specific assignment of the Device connections to the Host connectors. Such assignment is named *Host Connections Map*.

The Host Connections Map is hard-coded in the product firmware variant.

NOTE: The Coaxlink frame grabber and the firmware variant must be selected according to the required mapping!

1D1 host connections map

Applies to the following firmware variants of **PC3621-LH-T Coaxlink Mono CXP-12-LH:**


 (1-camera), (1-camera, line-scan)

One 1-connection device

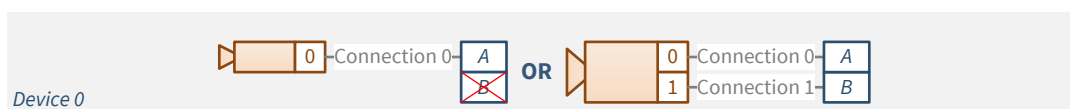


1D2 host connections map

Applies to the following firmware variants of **PC3622-T Coaxlink Duo CXP-12:**

 (1-camera), (1-camera, line-scan)

One 1- or 2-connection device



1D4 host connections map

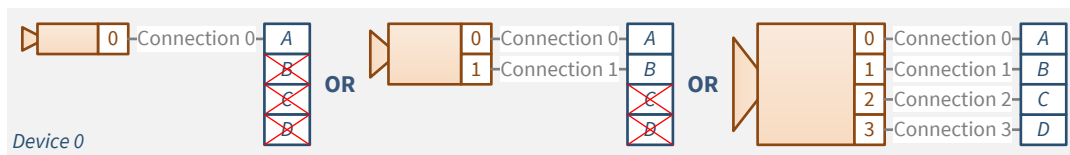
Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value:

QuadG3 ^{Prelim} (1-camera), (1-camera, 4-data-stream), (1-camera, line-scan)

Octo ^{Prelim} (1-camera, custom-logic)

Value12 ^{Prelim} (1-camera), (1-camera, line-scan)

One 1- or 2- or 4-connection device



1D4S4 host connections map

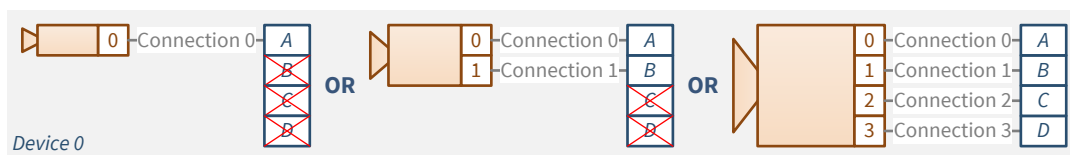
Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value:

QuadG3 ^{Prelim} (1-camera), (1-camera, 4-data-stream), (1-camera, line-scan)

Octo ^{Prelim} (1-camera, custom-logic)


Value12 ^{Prelim} (1-camera), (1-camera, line-scan)

One 1- or 2- or 4-connection device, up to 4 data streams

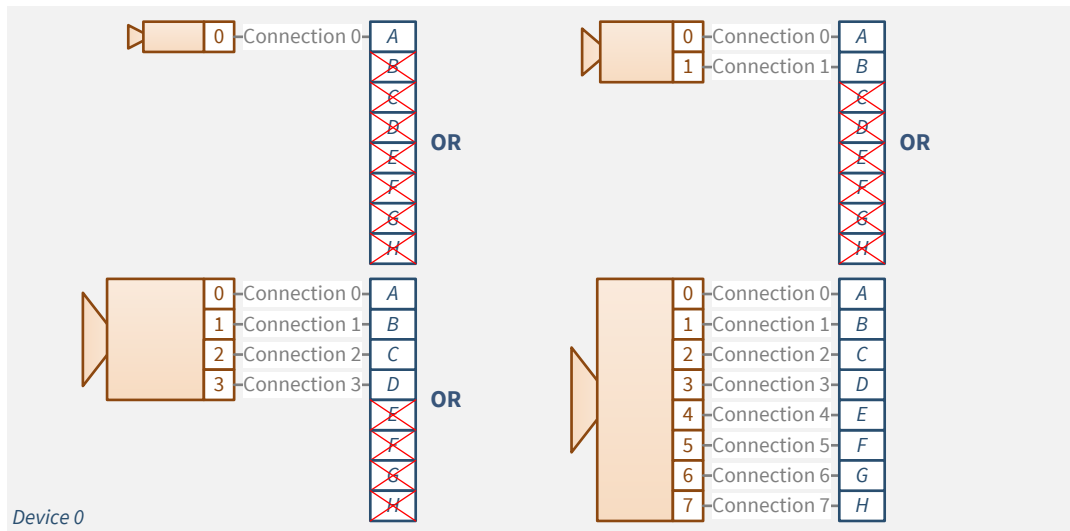


1D8 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:

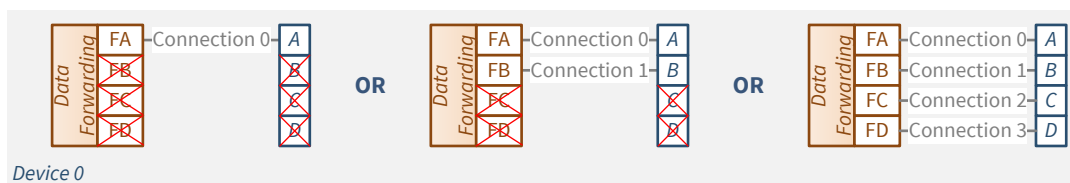
 (1-camera), (1-camera, line-scan)

One 1- or 2- or 4- or 8-connection device



1DF4 host connections map


One 1- or 2- or 4-connection device



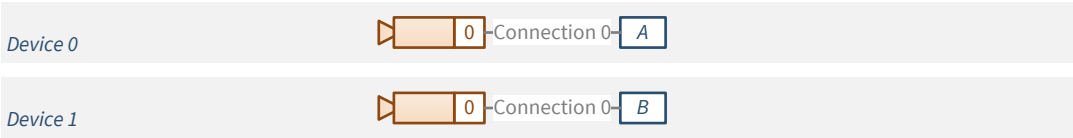
See also: CoaXPress Data Forwarding for the connection schemes of slave Data Forwarding devices.

2D11 host connections map

Applies to the following firmware variants of PC3622-T Coaxlink Duo CXP-12:


 (2-camera), (2-camera, line-scan)


One or two 1-connection devices



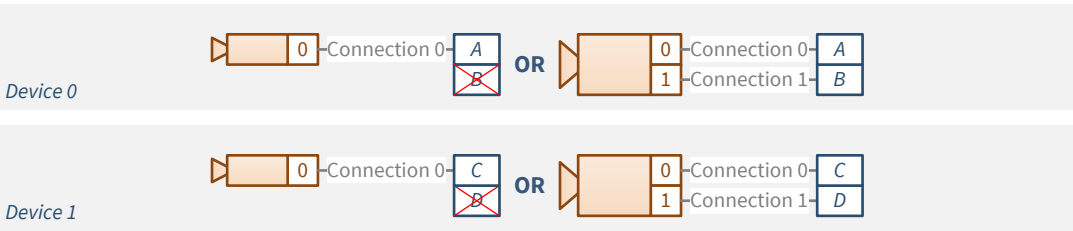
2D22 host connections map

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3 and PC3623-T Coaxlink Quad CXP-12 Value:

 (2-camera), (2-camera, bayer), (2-camera, line-scan)

 (2-camera), (2-camera, line-scan)

One or two 1- or 2-connection devices

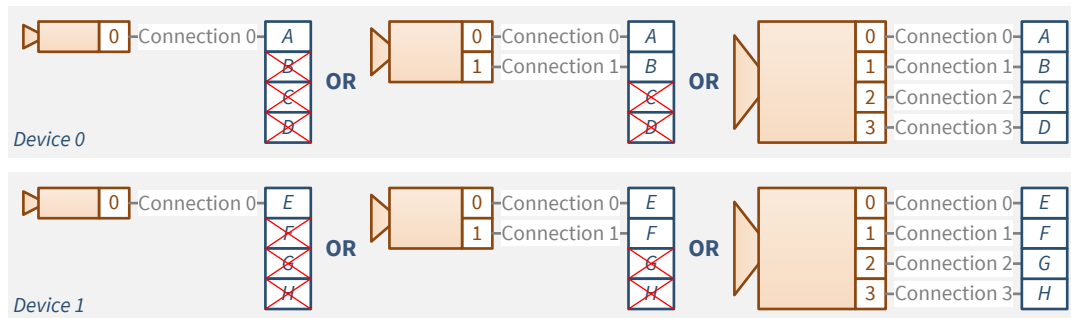


2D44 host connections map

Applies to the following firmware variants of **PC3602-T Coaxlink Octo**:

Octo Prelim (2-camera), (2-camera, line-scan), (2-camera, line-scan, custom-logic)

One or two 1- or 2- or 4-connection devices



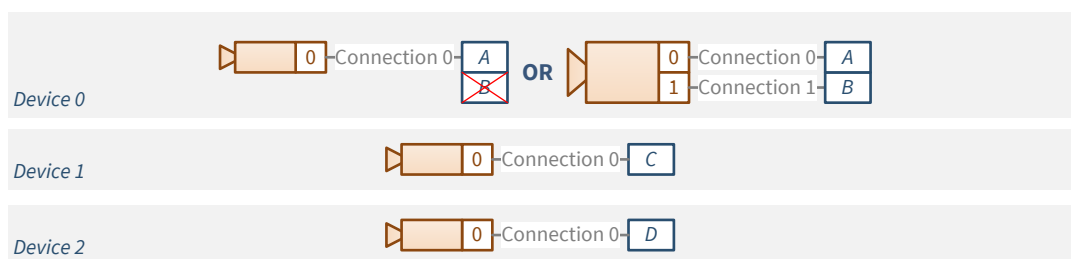
3D211 host connections map

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3** and **PC3623-T Coaxlink Quad CXP-12 Value**:

QuadG3 Prelim (3-camera)


Value12 Prelim (3-camera)

One 1- or 2-connection and one or two 1-connection devices

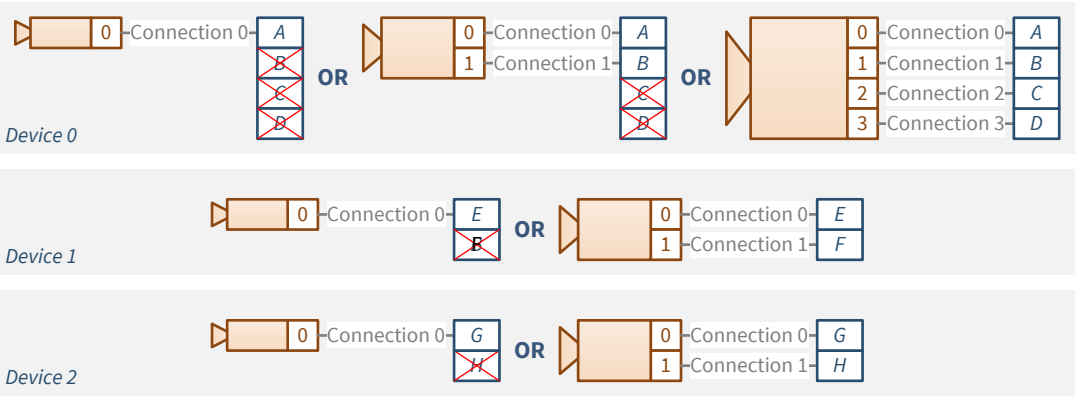


3D422 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:


 (3-camera)


One 1- or 2- or 4-connection and one or two 1- or 2-connection devices



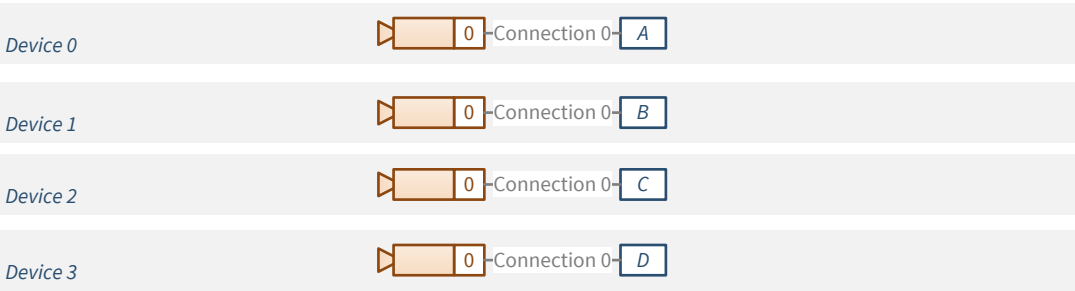
4D1111 host connections map

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3 and PC3623-T Coaxlink Quad CXP-12 Value:

 (4-camera), (4-camera, line-scan)


 (4-camera), (4-camera, line-scan)

One or two or three or four 1-connection devices

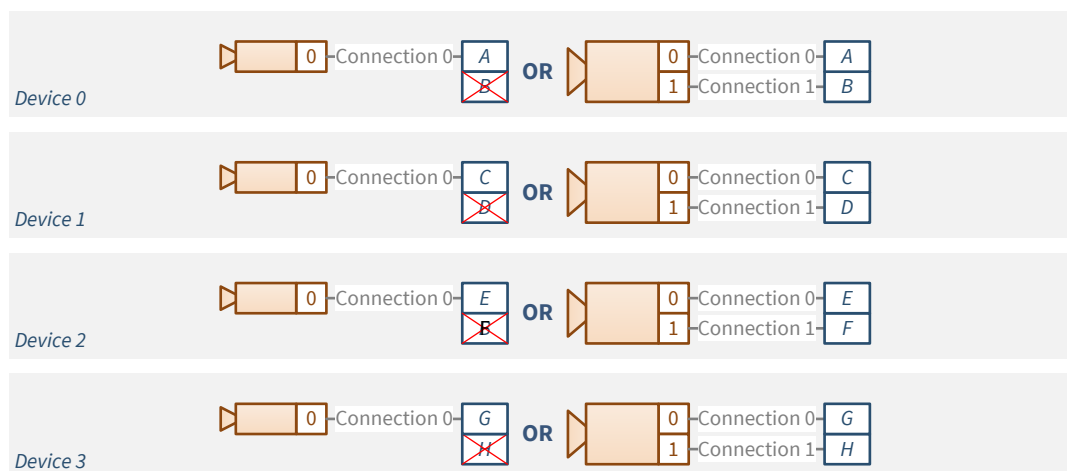


4D2222 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:


 (4-camera), (4-camera, line-scan)

One or two or three or four 1- or 2-connection devices

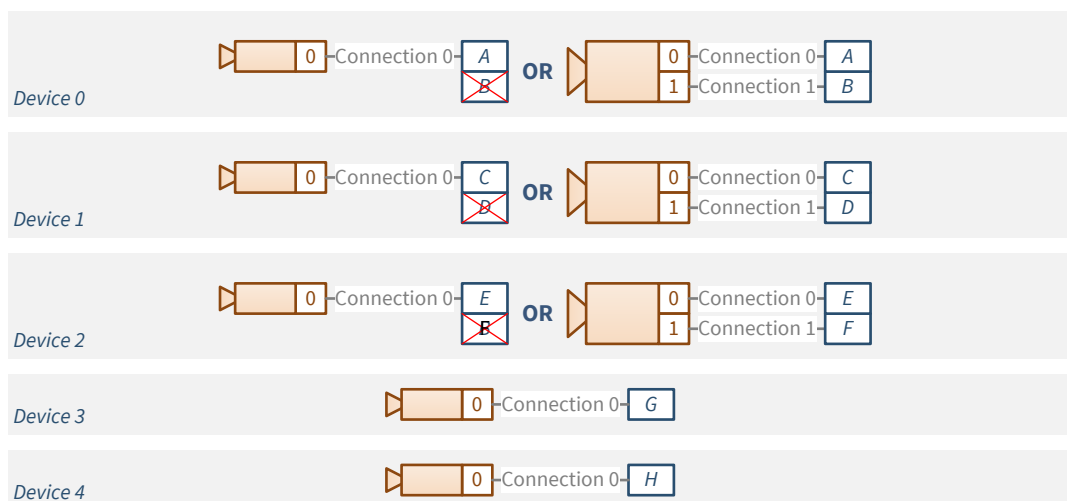


5D22211 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:

 (5-camera, 5D22211)

One or two or three 1- or 2-connection and one or two 1-connection devices

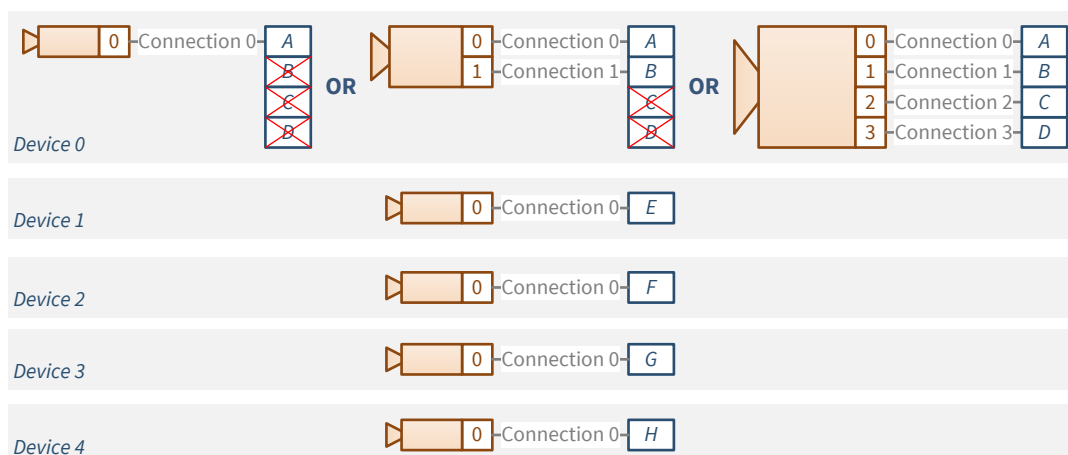


5D41111 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:

 (5-camera)

One 1- or 2- or 4-connection and one or two or three or four 1-connection devices

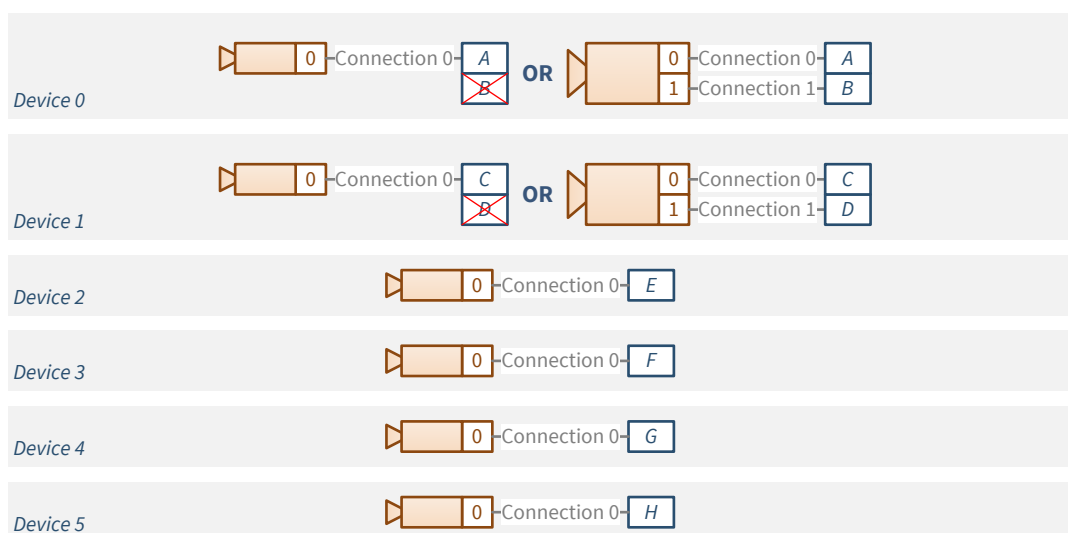


6D221111 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:


 (6-camera)

One or two 1- or 2-connection and one or two or three or four 1-connection devices

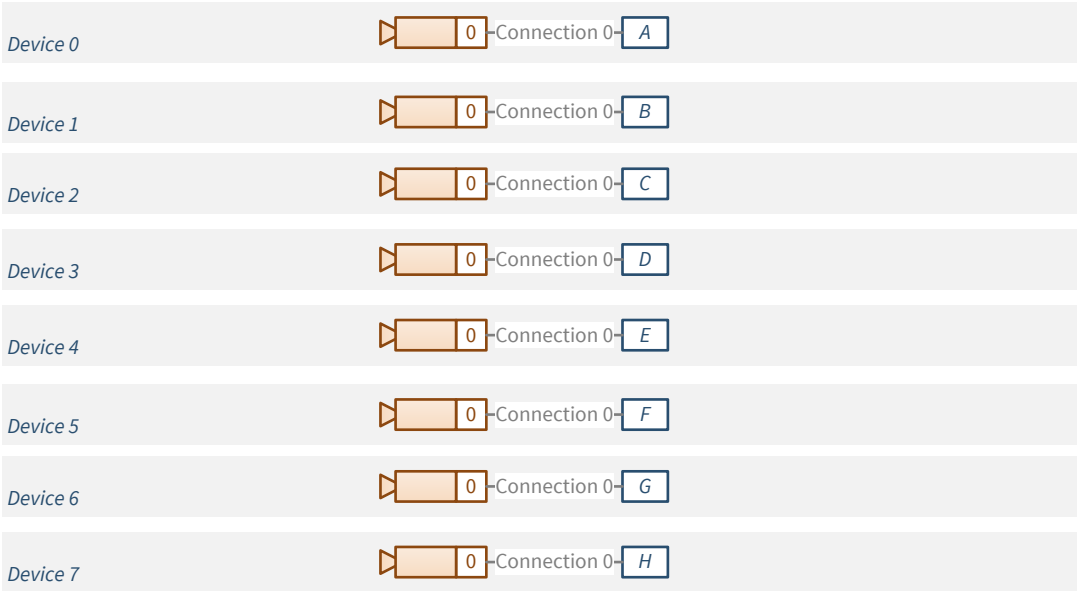


8D11111111 host connections map

Applies to the following firmware variants of PC3602-T Coaxlink Octo:

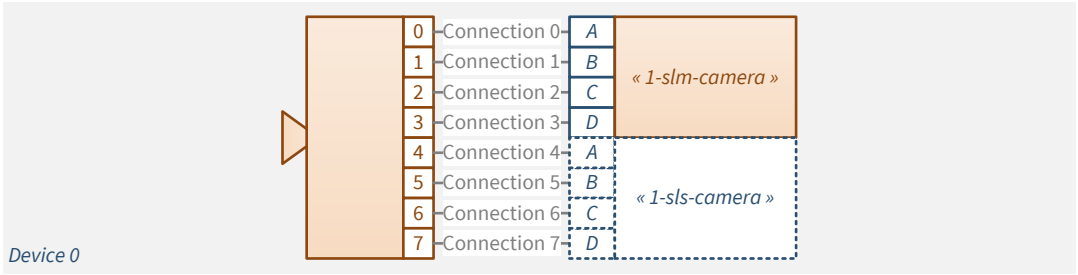
 (8-camera), (8-camera, line-scan)

Up to eight 1-connection devices



1D8SLM4 host connections map

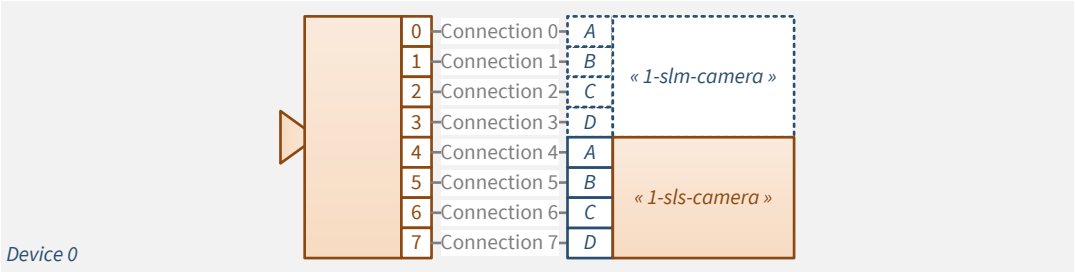
Master 4-connection sub-link of an 8-connection device



See also: 8-connection CoaXPress Cameras for the connection scheme of an 8-connection camera to two Coaxlink cards.

1D8SLS4 host connections map

Slave 4-connection sub-link of an 8-connection device



See also: 8-connection CoaXPress Cameras for the connection scheme of an 8-connection camera to two Coaxlink cards.

2.3. CoaXPress Link Configuration

Automatic Link Configuration

The **eGrabber** driver provides an automatic link discovery and configuration for CoaXPress 1.0 , CoaXPress 1.1 and CoaXPress 2.0 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 the card in terms of camera connection schemes supported by the target 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 device module.

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 Host side

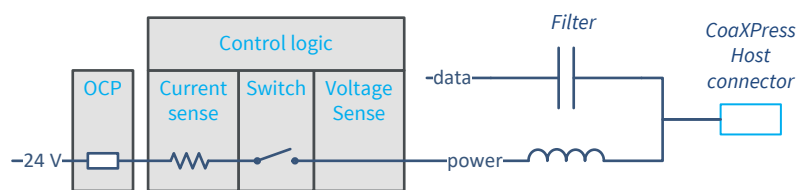
2.4. Power Over CoaXPress

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

Quad G3	Octo	Mono12LH	Duo12	Value12
---------	------	----------	-------	---------

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

Power transmitter unit



PoCXP power transmission unit

The Power Transmitting Unit – PTU – is responsible for a safe delivery of power to the Device. It fulfills the requirements of the CoaXPress standard for a CoaXPress Host, namely:

- It is capable of delivering 17 W (or 25W for selected products) of 24 V DC power per connector to the Device
- It implements an over-current protection device – OCP
- It supports the automatic CoaXPress PoCXP detection method

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

- Disabling or interrupting the automatic power delivery
- Resetting the OCP when tripped
- Measuring the PoCXP output current and the PoCXP output voltage on each connector
- Controlling the range of the PoCXP sense resistance

See also: "Power Distribution Schemes" on page 337 section in the Hardware Manual

Automatic PoCXP control

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



NOTE

Since version 3.1 of the driver, the automatic PoCXP is enabled at system power-up! The application is not anymore required to enable PoCXP powering by issuing a **CxpPoCxpAuto** command.

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

Manual PoCXP control

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**.

PoCXP detection mode control

The `CxpPoCxpDetectionMode` feature of the Interface module selects either the standard or the extended (default) power over CoaXPress detection mode.

When set to `Extended`, the PoCXP device detection of Coaxlink cards is configured for an extended range of resistance values. This allows cameras that are not fully compliant with the range specification of the PoCXP sense resistor to be detected as valid PoCXP cameras and to be powered. This is the default value after initialization.

When set to `Standard` the PoCXP device detection of Coaxlink cards is configured for a restricted range of resistance values, namely $4.7\text{ k}\Omega \pm 10\%$.



WARNING

This setting is not persistent.

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.

See also: "PoCXP Power Output Specifications" on page 349 in the Hardware Manual

Output current and voltage measurements

The `CxpPoCxpCurrent` and the `CxpPoCxpVoltage` features of the Interface module GenApi features report, respectively, the current and the voltage delivered by the PoCXP transmitter unit of the CoaXPress physical Host connection designated by `CxpPoCxpHostConnectionSelector`.

When `CxpHostConnectionSelector` is set to `All`, the `CxpPoCxpCurrent` Interface module GenApi feature reports the sum of currents delivered via PoCXP and the `CxpPoCxpVoltage` Interface module GenApi feature reports the average voltage delivered via PoCXP.



TIP

The total output power delivered by PoCXP is the product of `CxpPoCxpCurrent[All]` and `CxpPoCxpVoltage[All]` values.

2.5. CoaXPress I/O Channel

The Coaxlink cards implement the CoaXPress Host to Device and Device to Host triggers.



According to the CoaXPress 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 high-priority "triggers" between the Host and the Device
- Is used for transmitting high-priority "triggers" between the Device and the Host



NOTE

The CoaXPress 1.0 GPIO is NOT implemented on **Coaxlink frame grabbers**

See also: "CoaXPress Host To Device Trigger" on page 56 for a detailed description of how to configure the Host (frame grabber) to Device (camera) trigger through the CoaXPress Link

See also: and "Device Link Trigger Tool" on page 248 for a detailed description of the I/O Toolbox tool dedicated to receive triggers from the Device (camera)

2.6. CoaXPress Host To Device Trigger

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 implements one CoaXPress Host to Device trigger transmitter for *each connected Device*.

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** GenApi 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, ... LIN8 for Device7.
 - Any input-capable GPIO line can be used as trigger source by configuring the **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 **INTERNAL**, any I/O Toolbox event can generate a CoaXPress **LinkTrigger0** or **LinkTrigger1** message. Therefore:
 - Select the CoaXPress Host to Device trigger message source by setting the **CxpTriggerMessageSelector** feature of the Device module to **LinkTrigger0** or **LinkTrigger1**
 - Select the I/O Toolbox event by setting **CxpTriggerMessageSource** to the desired value
- When **CameraControlMethod** is set to **NC**, the Host to Device Trigger transmitter is disabled.

Host to Device Trigger Transmitter - Default Settings

The implementation of the CoaXPress Host to Device Trigger transmitter complies with the requirements of the CoaXPress standard for a low-speed CoaXPress Host to Device Trigger 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.
- 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):

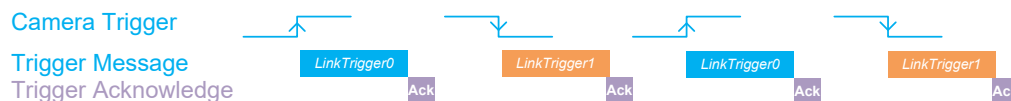
- If the acknowledgment is received before the expiration of the timeout, the transaction terminates normally.
- If no acknowledgment is received within the 20 μ s timeout, the transmitter performs a *retry*: it resends the trigger packet and initiates a new waiting period for the acknowledgment.
- If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

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

Default settings

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

Case 1: `CameraControlMethod = RG`; camera replies immediately with ACK



Case 2: `CameraControlMethod = RC`; camera replies immediately with ACK



Case 3: no reply from camera: retry after timeout (20 μ s); transaction aborted after 3 retries



Trigger message transactions using default settings
Case 1 and case 2: the camera acknowledges each message as expected
Case 3: no acknowledgment from camera. Abort after 3 retries

NOTE: In CoaXPress v1.x, `LinkTrigger0` was called "Rising edge" and `LinkTrigger1` "Falling edge"

Events Reporting

The transmitter reports the following events:

- **CxpTriggerAck:** Received acknowledgment for CoaXPress Host to Device trigger packet.
- **CxpTriggerResend:** Resent CoaXPress Host to Device trigger packet.

See also: "Advanced Trigger Transmitter Settings" on page 58 and "Camera Trigger Jitter Compensation" on page 60

2.7. Advanced Trigger Transmitter Settings

The Host to Device Trigger Transmitter can be customized:

- To send trigger messages only on the rising edge of the source signal using the "[Message Format Control](#)" on page 58
- To configure the acknowledge timeout and the number of retries using the "[Message Acknowledge Control](#)" on page 59

Message Format Control

The Host to Device Trigger transmitter unit provides a "message format" control with the `CxpTriggerMessageFormatGenApi` feature.

Pulse Message Format (Default)

By default, `CxpTriggerMessageFormat` is set to **Pulse**: the transmitter generates 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 `LinkTrigger0` packet from the Host to the Device.
- The transaction initiated by the falling edge transmits a `LinkTrigger1` packet from the Host to the Device.

NOTE: Every trigger pulse requires two distinct CoaXPress I/O Channel transactions!

Rising Edge Message Format

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

The transaction always transmits a `LinkTrigger0` packet from the Host to the Device.

NOTE: Every trigger pulse requires a single CoaXPress I/O Channel transaction.

NOTE: This format does not allow the grabber to control the exposure time!

Toggle Message Format

When `CxpTriggerMessageFormat` is set to **Toggle**, the transmitter generates a CoaXPress I/O Channel Host to Device Trigger transaction on *the rising edge only* of the input pulse.

The transaction alternatively transmits a `LinkTrigger0` packet and a `LinkTrigger1` packet.

NOTE: Every trigger pulse requires a single CoaXPress I/O Channel transaction.

NOTE: This format does not allow the grabber to control the exposure time!

The `CxpTriggerLevel` feature allows the application to set and/or get the current level of the CoaXPress Host to Device Trigger signal.

Message Acknowledge Control

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

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

Enable Acknowledge Checking (Default)

By default, `CxpTriggerAckTimeout` is set to **20.0** (20 microseconds) and `CxpTriggerMaxResendCount` is set to **3**.

The Coaxlink card expects an I/O Channel Acknowledgment packet in response to every Trigger packet. If the acknowledgment packet is not received within the 20 μ s time-out value, the transmitter resends the `LinkTrigger0` packet. It performs up to 3 retries.

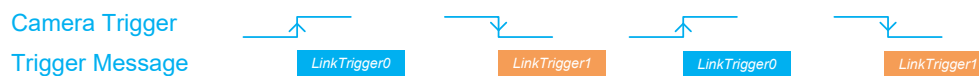
Setting larger `CxpTriggerAckTimeout` values allows more time for the Device to acknowledge the trigger packet.

Disable Acknowledge Checking

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

Alternate settings

Case 1: `CameraControlMethod = RG;CxpTriggerMessageFormat = Pulse; CxpTriggerAckTimeout = 0;`



Case 2: `CameraControlMethod = RC;CxpTriggerMessageFormat = Pulse; CxpTriggerAckTimeout = 0;`



Case 3: `CameraControlMethod = RC;CxpTriggerMessageFormat = Rising; CxpTriggerAckTimeout = 0;`



Case 4: `CameraControlMethod = RC;CxpTriggerMessageFormat = Toggle; CxpTriggerAckTimeout = 0;`



Trigger message transactions using alternate settings to allow higher trigger rates

Alternate settings for fastest trigger rate

The fastest trigger rate of : 595.2 kHz @CXP-10 and CXP-12 link speeds or 297.6 kHz @CXP-6 and lower link speeds can be achieved when:

- `CameraControlMethod = RC` (asynchronous reset camera, camera-controlled exposure),
- `CxpTriggerAckTimeout = 0` (acknowledge checking disabled),
- `CxpTriggerMessageFormat = Rising` or `CxpTriggerMessageFormat = Toggle`.

2.8. Camera Trigger Jitter Compensation

Trigger accuracy

The Host to Device trigger packets are transmitted over the low-speed up-connection of the CoaXPress Link. The transmission of trigger packets can only start at the boundary of a character. This introduces a jitter corresponding to one character transmission time:

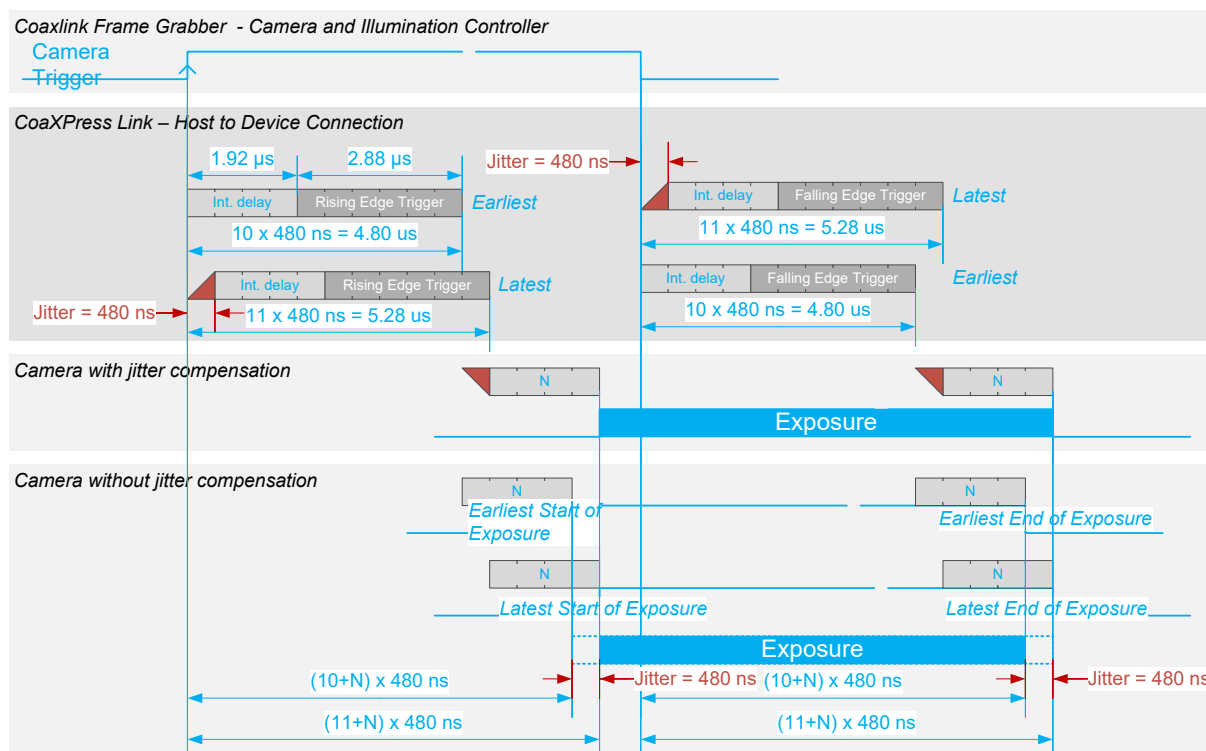
- 240 nanoseconds @CXP-10 and CXP-12 link speeds or
- 480 nanoseconds @CXP-6 and lower link speeds.

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 1/24th of the bit period:

- 1 nanosecond @CXP-10 and CXP-12 link speeds or
- 2 nanoseconds @CXP-6 and lower link speeds.

The receiver (camera) can then use this value to recreate the trigger event with a *fixed latency*. It compensates the transmission jitter by delaying the decoded message by the remaining fraction of one character time.

CoaXPress Camera Trigger transmission timing @CXP-6 and lower link speeds



CIC Camera Trigger to Sensor Exposure timing diagrams @CXP-6 and lower link speeds

The above diagram shows the time delay required to propagate Camera Trigger events from the frame grabber up to the camera 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 (or the falling) edge of the Camera Trigger signal (inside the Coaxlink card) up the CoaXPress link is composed of:

- A variable delay of 0-480 ns corresponding to the time delay until the next character boundary on the low-speed CoaXPress connection.
- A fixed delay of 1.92 μs corresponding to a 4-character pipeline delay in the Trigger Transmitter implementation.
- A fixed delay of 2.88 μs corresponding to a 6-character message transmission time.

The delay from the CoaXPress Link to the effective start (or end) of exposure is camera-dependent. In the above drawing, this delay is assumed to be N character times (with N=4).

Jitter-compensated cameras

When the camera implements the CoaXPress jitter compensation, the one-character jitter (480 ns) introduced by the transmitter can be entirely compensated.

The overall latency is *fixed* but it remains camera-dependent: the lowest possible latency is $11 \times 480 \text{ ns}$, i.e. $5.28 \mu\text{s}$.

The residual jitter after compensation can be as low as 4 ns .

Jitter-Uncompensated cameras

When the camera doesn't implement the CoaXPress jitter compensation, the one-character jitter (480 ns) introduced by the transmitter remains.

The overall latency is *variable* and camera-dependent: the lowest possible latency is $(10 \sim 11) \times 480 \text{ ns}$, i.e. $(4.80 \sim 5.28) \mu\text{s}$.

CoaXPress Camera Trigger transmission timing @CXP-10 and CXP-12 link speeds

The delay from the rising (or the falling) edge of the **Camera Trigger** signal (inside the Coaxlink card) up the CoaXPress link is composed of:

- A variable delay of $0\text{-}2400 \text{ ns}$ corresponding to the time delay until the next character boundary on the low-speed CoaXPress connection.
- A fixed delay of $0.96 \mu\text{s}$ corresponding to a 4-character pipeline delay in the Trigger Transmitter implementation.
- A fixed delay of $1.44 \mu\text{s}$ corresponding to a 6-character message transmission time.

The delay from the CoaXPress Link to the effective start (or end) of exposure is camera-dependent. In the above drawing, this delay is assumed to be N character times (with $N=4$).

Jitter-compensated cameras

When the camera implements the CoaXPress jitter compensation, the one-character jitter (240 ns) introduced by the transmitter can be entirely compensated.

The overall latency is *fixed* but it remains camera-dependent: the lowest possible latency is $11 \times 240 \text{ ns}$, i.e. $2.64 \mu\text{s}$.

The residual jitter after compensation can be as low as 2 ns .

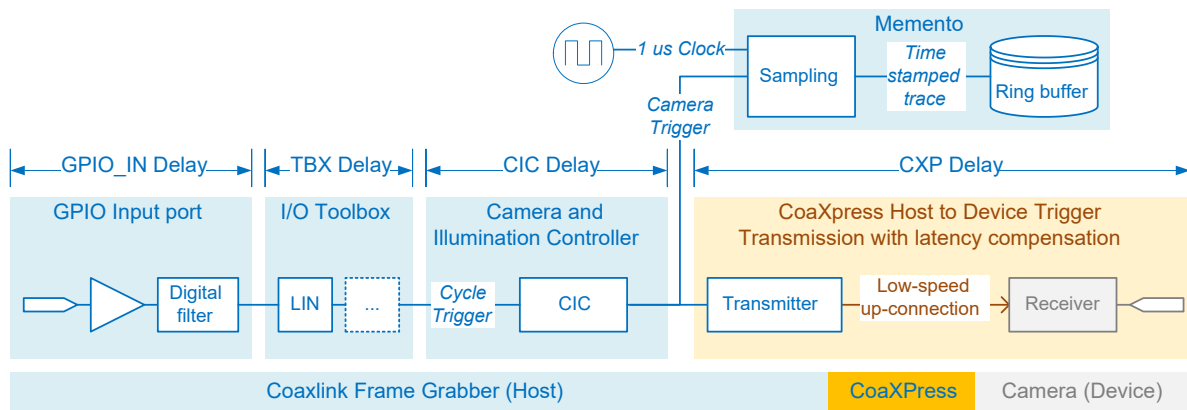
Jitter-Uncompensated cameras

When the camera doesn't implement the CoaXPress jitter compensation, the one-character jitter (240 ns) introduced by the transmitter remains.

The overall latency is *variable* and camera-dependent: the lowest possible latency is $(10 \sim 11) \times 240 \text{ ns}$, i.e. $(2.40 \sim 2.64) \mu\text{s}$.

2.9. Trigger Delay Model

This topic describes a timing model of a trigger signal applied to any GPIO input port of a Coaxlink card up to the output of the CoaXPress Host to Device Trigger receiver inside the camera.



Elaboration of the Camera Trigger event

Three functional blocks are involved in the elaboration of the Camera Trigger event:

- The GPIO Input port receiving and cleaning the electrical signal
- The I/O Toolbox tool(s) receiving the cleaned signal and delivering the cycle trigger event to the CIC
- The Camera and Illumination controller

NOTE: the Camera Trigger event is recorder by Memento and time-stamped with 1 MHz clock.

GPIO Input Port — GPIO_IN delay

The delay introduced by the GPIO Input port depends on:

- The electrical type
- The settings of the associated digital filter

The following table shows the typical GPIO-IN delay values with the lowest digital line filter strength:

- $\Delta t_{on/high}$ is the delay value at the low-to-high transition of the input signal or when the optocoupler turns ON.
- $\Delta t_{off/low}$ is the delay value at the high-to-low transition of the input signal or when the optocoupler turns OFF.

GPIO electrical type	$\Delta t_{on/high}$	$\Delta t_{off/low}$
"Differential Input (Version 1)" on page 352	50 ns	50 ns
"Differential Input (Version 2)" on page 354	50 ns	50 ns
Differential Input/Output	50 ns	50 ns
"TTL Input/Output (Version 1)" on page 356	50 ns	50 ns
"TTL Input/Output (Version 2)" on page 359	50 ns	50 ns
TTL Input/5 V CMOS Output	50 ns	50 ns
"Isolated Input (Version 1)" on page 362	2.1 μ s	4.5 μ s
"Isolated Input (Version 2)" on page 365	0.8 μ s	1.35 μ s
Isolated Input (Version 3)	0.6 μ s	0.6 μ s
"Isolated Input (Version 4)" on page 368	6.9 μ s	9.8 μ s

See also: "Line Filter Control" on page 231 for other settings of the digital filter.

I/O Toolbox — TBX delay

The delay introduced by the I/O toolbox depends on its configuration:

- There are no significant delay when the LIN tool is driving directly the CIC Cycle Trigger.
- A constant delay can be introduced when a DEL tool is involved in the generation of the CIC Cycle Trigger event.
- When DIV and MDV tools are involved, a delay depending of the input source frequency and the parameters values can be introduced.

See also: "I/O Toolbox" on page 238 for a detailed description of the I/O Toolbox tools

Camera and Illumination Controller — CIC delay

The delay introduced by the Camera and Illumination Controller depends on its configuration:

- There are no significant delay when **StrobeDelay** is 0 or > 0 providing that the conditions to start a cycle are all satisfied.
- When **StrobeDelay** is negative, the Camera Trigger is delayed accordingly.

See also: "Camera and Illumination Control" on page 187 and "CIC Timing Diagrams" on page 208

Transmission of the Camera Trigger event- CXP Delay

Two functional blocks are involved in the transmission of the **Camera Trigger** event:

- The CoaXPress Host to Device Trigger transmitter inside the Coaxlink card
- The CoaXPress Host to Device trigger receiver of the camera

The transmitter implements the jitter compensation described in the CoaXPress standard.

If the receiver implements also the jitter compensation, the transmission delay is constant despite the asynchronism of the trigger messages on the CoaXPress medium.

In that case, the transmission time only depends upon the link speed:

Link speed	Delay
CXP-1 to CXP-6	5.28 μ s
CXP-10 to CXP-12	2.64 us

See also: "Camera Trigger Jitter Compensation" on page 60 for a detailed explanation



NOTE

The delay introduced by the medium is usually not significant!

2.10. CoaXPress LED Lamps

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.



Each CoaXPress connection is associated with a LED lamp mounted on the bracket (for PCIe cards only).




CoaXPress Host Indicator LED lamps states

States description

Symbol	LED State	Meaning
	Off	No power
	Solid orange	System booting
	AlternateFlash_12_5 green / orange ¹	Connection detection in progress; PoCXP active
	Flash_12_5 orange ²	Connection detection in progress; PoCXP not in use
	AlternateFlash_0_5 red / green	Device/ Host incompatible; PoCXP active
	AlternateFlash_0_5 red / orange	Device/ Host incompatible; PoCXP not in use
	Solid red	PoCXP over-current
	Solid green	Device / Host connected, but no data being transferred
	Flash_1 orange	Device / Host connected, waiting for event (e.g. trigger, exposure pulse)
	Flash_12_5 green	Device / Host connected, data being transferred

¹ Shown for a minimum of 1 second even if the connection detection is faster

² Shown for a minimum of 1 second even if the connection detection is faster

Symbol	LED State	Meaning
	500 ms red pulse ¹	Error during data transfer (e.g. CRC error, single bit error detected)
	AlternateFlash_0_5 green / orange	Connection test packets being sent
	Flash_12_5 red	System error (e.g. internal error)

Flashing states timing definitions

Indication	Frequency	Duty Cycle
Flash_12_5	12.5 Hz	25% (20 milliseconds on, 60 milliseconds off)
Flash_1	1 Hz	20% (200 milliseconds on, 800 milliseconds off)
Flash_0_5	0.5 Hz	50% (1 second on, 1 second off)
AlternateFlash_12_5	12.5 Hz	25% (20 milliseconds on color 1, 60 milliseconds off, 20 milliseconds on color 2, 60 milliseconds off)
AlternateFlash_0_5	0.5 Hz	50% (1 second on color 1, 1 second off, 1 second on color 2, 1 second off)

LED lamps mode control

The **LampMode** feature of the Interface module defines the lamps operation mode:

- When set to **Standard** (default value), the lamps indicate the state of the CoaXPress Link connection.
- When set to **Dark**, all lamps are turned off.
- When set to **Error**, all lamps are turned off unless error conditions are detected.
- When set to **Custom**, all lamps are controlled by **LampCustomValue**, a bitfield where each bit is mapped onto a lamp with 1 for orange and 0 for off by the **LampCustomLedA** ... **LampCustomLedH** boolean features.

¹ In case of multiple errors, there shall be at least two green Flash_12_5 pulses before the next error is indicated

2.11. Connection Test

The CoaXPress Host Interface 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.

2.12. CoaXPress 2.0 Error Counters

The "CoaXPress 2.0 error counters" keep track of errors that the CoaXPress protocol can detect on each individual CoaXPress connection.

Error counters

Connection lock loss counters

There is one counter per CoaXPress host connector instance that counts the number of lock losses encountered by the CoaXPress receiver.

8b/10b encoding error counters

There is one counter per CoaXPress host connector instance that counts the number of 8b/10b encoding errors encountered by the CoaXPress receiver.

Duplicated characters mismatch counters

There are two counters per CoaXPress host connector instance that counts the number of duplicated characters mismatch encountered by the CoaXPress receiver:

- ❑ The first counter counts the occurrences that could be corrected.
- ❑ The second counter counts the occurrences that could NOT be corrected.

CRC error counters

There are three counters per CoaXPress host connector instance that counts the number of CRC errors encountered by the CoaXPress receiver:

- ❑ The first counter counts the occurrences in data packets.
- ❑ The second counter counts the occurrences in control packets.
- ❑ The third counter counts the occurrences in event packets.

Error counters management

The application uses the counters by means of Interface module feature of the **CoaXPressErrorCounters Category**.

Getting the current count value

1. Select a connector instance by setting the **CxpHostConnectionSelector**
2. Read the corresponding Interface module feature (e.g. **CxpLinkLockLossCount**)

Resetting a counter

1. Select a connector instance by setting the **CxpHostConnectionSelector**
2. Write to the corresponding Interface module feature (e.g. **CxpLinkLockLossCountReset**)

Related Interface module GenApi features

- CoaXPressErrorCounters parameters category
- CxpLinkLockLossCount and CxpLinkLockLossCountReset
- Cxp8b10bErrorCount and Cxp8b10bErrorCountReset
- CxpDuplicatedCharactersUncorrectedErrorCount and CxpDuplicatedCharactersUncorrectedErrorCountReset
- CxpDuplicatedCharactersCorrectedErrorCount and CxpDuplicatedCharactersCorrectedErrorCountReset
- CxpStreamDataPacketCrcErrorCount and CxpStreamDataPacketCrcErrorCountReset
- CxpControlPacketCrcErrorCount and CxpControlPacketCrcErrorCountReset
- CxpEventPacketCrcErrorCount and CxpEventPacketCrcErrorCountReset

2.13. CoaXPress Link Validation Tool

Introduction

Short Description

The *CoaXPress Link Validation Tool* (CXLVT) can be used to validate the operational parameters of a CoaXPress Link.

For a *quick test*, run the CXLVT until reaching a confidence level of 100% that the probability of single bit error (PER) is 10^{-10} or better 10^{-11} . This should just take a *few minutes*.

For an *extensive test*, run the CXLVT until reaching a confidence level of 100% that the PER is 10^{-12} or better 10^{-13} . This will take a *few hours*.

See also: http://en.wikipedia.org/wiki/Bit_error_rate for more information about the theory of bit error rate testing.

Host PC requirements

- The Host PC must be equipped with at least one Coaxlink card.
- The driver must be installed on the Host PC.

Camera requirements

- The camera must be capable to generate a static image pattern.

Installation

The CXLVT is included in gentl.exe, a command-line tool that is delivered with the driver. No further installation is required.

gentl ber Command

The CXLVT is invoked with the command ber of gentl.exe.

```
$ gentl ber --help
GenTL Explorer

gentl ber [OPTIONS]
  Measure bit error rate confidence level (a.k.a. link validation tool)

Flags:
  --if=ID          Interface ID
  --dev=ID         Device ID
  --ds=ID          DataStream ID
  --buffers=INT    Buffer count (default: 4)
  --set=SETTINGS   GenApi settings, such as Module.Feature=INT
  --setup=FILE     Path to script to execute before starting stream
  --run=FILE       Path to script to execute concurrently with stream
  --remotexml=FILE Use FILE as register description (default:
                  register description is read from remote device)
  -c --create-only Create a reference pattern and quit (requires
                  --output)
  -i --input=FILE  Input reference pattern file (default:
                  automatically create a reference image before
                  measuring the bit error rate confidence level)
  -o --output=FILE Output reference pattern file (default: no output
                  file)
                  --enable-dump=FILE Enable dump of defective surfaces to files with
                  the given file path prefix

Common flags:
  --cti=LIBPATH    Path to GenTL producer library. Default: use
                  EURESYS_COAXLINK_GENTL64_CTI and
                  GENICAM_GENTL64_PATH environment variables to locate
                  the library.
  -j=N            Limit the number of CPU cores to use to N
                  (default: 2)
  -h --help       Display help message
  -V --version     Print version information
                  --numeric-version Print just the version number
  -v --verbose    Loud verbosity
  -q --quiet      Quiet verbosity
```

gentl ber --help

```
> help
Ber commands:
  levels          show confidence levels
  levels -N       show confidence levels every N seconds
  results         show intermediate bit error rate results
  results -N      show intermediate bit error rate results every N
                  seconds
  report FILE     write current report to a file
  enable-dump FILE enable dump of defective surfaces to files with the
                  given file path prefix
  disable-dump    disable dump of defective surfaces
General commands:
  quiet          set verbosity level to quiet
  normal         set verbosity level to normal
  loud           set verbosity level to loud
  help           display this help message
  exit           exit the CLI
>
```

gentl ber commands

Test procedure

To setup the *CoaxPress Link Validation Tool* proceed as follows:

1. With GenICam Browser (Deprecated):

- ☐ Configure the camera as for normal operation and select a fixed test-pattern as video source
- ☐ Configure the frame grabber as for normal operation

2. Open a command shell and execute gentl ber to start a Read-Eval-Print-Loop

3. Get intermediate results using the resultscommand

- ☐ Check if the number of acquired images counter increases regularly
- ☐ Check the confidence levels

4. Run the test until the required confidence levels are reached. This may require several hours.

Operation

The CoaXPress Link Validation Tool (CXLVT) validates the operational parameters of a CoaXPress Link installation (bit rate, cable type, cable length) resulting in reliable, long-term performance.

CXLVT does this by estimating, with a known confidence level, the probability of single bit errors in a CoaXPress Link setup.

We define:

- *PER*: Probability of single bit error in a digital connection like a CoaXPress Link; this is an unknown quantity that we want to estimate
- *BER*: Bit Error Rate, actually measured by the CXLVT

It is generally accepted that a CoaXPress Link will operate reliably, if $PER < 10^{-12}$. This criterion is similar to the one used in other digital serial image transmission schemes. Of course, a better (lower) PER will provide even more assurance that the operation is reliable.

The CXLVT computes the confidence level (CL), or likelihood, that the PER is less than a set of values (10^{-10} , 10^{-11} , 10^{-12} , 10^{-13} , 10^{-14}), based on the measurement of the BER, during a time sufficiently long to accumulate the necessary evidence.

When started, the CXLVT displays these confidence levels, as evidence accumulates with the passing of time, as illustrated in the screenshots hereafter.

Entering the levels command, during the operation of the CXLVT, displays the confidence levels for the 5 PER values.

```
> levels
```

Confidence level (rounded) that the probability of error is less than:					Elapsed Time
1.0e-10	1.0e-11	1.0e-12	1.0e-13	1.0e-14	H:MM:SS
99.99%	64.97%	9.95%	1.04%	0.10%	0:08:56

```
>
```

Confidence levels reported 8 seconds after start

After 8 seconds, we have reached 99.99 % confidence level that the PER is less than 10^{-10} . The PER might very well be much better than that, but at this stage we have insufficient evidence to conclude that this is the case. The CXLVT must be continued.

By entering the results command, during the operation of the CXLVT, additional information can be displayed, after which the CXLVT continues its normal operation, for as long as necessary, to achieve the required confidence level for a predetermined PER.

```
> results
Intermediate results:
-----
Duration (hours:minutes:seconds):                0:16:37
Duration (seconds):                               997
Acquired images:                                  11975
Bad images:                                       0
Acquired bits:                                    1.986509e11
Bit errors:                                       0.000000e0
Average bit errors per bad image:                 0
Bit rate (bits per second):                       1.992486e8
Bit error rate:                                   0.000000e0
Confidence level (rounded) that the probability of error
is less than 1.0e-10:                             99.99%
              1.0e-11:                             86.28%
              1.0e-12:                             18.01%
              1.0e-13:                              1.96%
              1.0e-14:                              0.19%
>
```

Intermediate results reported after 16 minutes

From this screenshot, we can already conclude that the confidence level that the PER is less than 10^{-11} has risen from 64.97 % to 86.28 %, after 16 minutes.

The CXLVT should be continued until the confidence level that the PER is less than 10^{-12} (at most – a stronger test would be a PER less than 10^{-13}) has reached a satisfactory level (at least 95 %, and 99 % for a stronger result). This may require quite some time, because these outcomes require a significant amount of evidence.

```
Intermediate results:
-----
Duration (hours:minutes:seconds):                3:30:12
Duration (seconds):                               12612
Acquired images:                                  149753
Bad images:                                       0
Acquired bits:                                    2.484223e12
Bit errors:                                       0.000000e0
Average bit errors per bad image:                 0
Bit rate (bits per second):                       1.969729e8
Bit error rate:                                   0.000000e0
Confidence level (rounded) that the probability of error
is less than 1.0e-10:                             100.00%
              1.0e-11:                             99.99%
              1.0e-12:                             91.66%
              1.0e-13:                             21.99%
              1.0e-14:                              2.45%
```

Intermediate results reported after 3.5 hours

From this screenshot, we can conclude that the confidence level that the PER is less than 10^{-12} has risen from 18.01 % to 91.66 %, after 3.5 hours.

To generate a report, execute the report command.

```
> report ber-report  
Report ber-report-20171107-093451.log successfully created  
> █
```

Generate a report command line

2.14. Multi-tap CoaXPress Cameras

Applies to the following firmware variants of PC3623-T Coaxlink Quad CXP-12 Value:

 (1-camera)

1X_2YE geometry

Multi-tap is controlled through the following GenApi features of the **MultiTapControl** category of the Coaxlink Data Stream module:

- **DeviceTapGeometry**: Image scan format. Supported values:
 - **1X_1Y**: Single tap. Default value.
 - **1X_2YE**: Two zones across vertical direction, pixel extractors at both top and bottom lines.
- **Image1StreamID**: Stream ID of first tap (ignored when **DeviceTapGeometry** is **Geometry_1X_1Y**).
- **Image2StreamID**: Stream ID of second tap (ignored when **DeviceTapGeometry** is **Geometry_1X_1Y**).

See also: "Transmission Methods of 1X_2YE Images (Coaxlink series)" on page 180

3. On-board Memory

Coaxlink frame grabbers are fitted with a large on-board memory.

On-board memory size per product

Product	Memory Size [MB]
PC1633-T Coaxlink Quad G3	1024
PC3602-T Coaxlink Octo	2048
PC3621-LH-T Coaxlink Mono CXP-12-LH	512
PC3622-T Coaxlink Duo CXP-12	1024
PC3623-T Coaxlink Quad CXP-12 Value	4096

On-board memory partitions

The memory is partitioned according to the installed firmware variants:

- For all firmware-variants, there is one *Image data* partition for each stream of each device.
- For firmware variants supporting **FFC**, there is another partition, named *FFC coefficients* for each stream of each device.

See also: "Partition Schemes" on page 79 for a detailed description.

Image data partition

The *Image data* partition is operated as a First-In-First-Out memory to decouple the CoaXPress data flow from the Pixel Processing and the PCI Express data flow.

It absorbs temporary dropouts of the PCI Express data flow ensuring a reliable CoaXPress data acquisition.

It enables burst-mode CoaXPress data acquisition at the highest data rates regardless the limits of the Pixel Processor and the PCI Express interface.

FFC coefficients partition






The *FFC coefficients* partition is used to store gain and offset coefficients for each pixel of the camera.

See also: "FFC IP Core Description" on page 124 for a detailed description.

3.1. Partition Schemes

One device, one 'Image data' partition

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value:



-  (1-camera, line-scan)
-  (1-camera), (1-camera, line-scan)
-  (1-camera), (1-camera, line-scan)
-  (1-camera), (1-camera, line-scan)
-  (1-camera, line-scan)

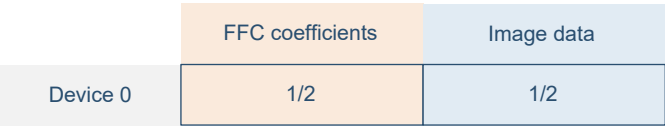


1D partition scheme

One device, two partitions per device: 'FFC coefficients' and 'Image data'

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3 and PC3623-T Coaxlink Quad CXP-12 Value:

-  (1-camera)
-  (1-camera)



1D_FFC partition scheme

Two devices, one 'Image data' partition per device

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12** Value:






-  (2-camera), (2-camera, bayer), (2-camera, line-scan)
-  (2-camera, line-scan)
-  (2-camera), (2-camera, line-scan)
-  (2-camera), (2-camera, line-scan)

	Image data
Device 0	1/2
Device 1	1/2

2D partition scheme

Two devices, two partitions per device: 'CustomLogic' and 'Image data'

Applies to the following firmware variants of **PC3602-T Coaxlink Octo**:

-  (2-camera, line-scan, custom-logic)

	CustomLogic	Image data
Device 0	1/4	1/4
Device 1	1/4	1/4

2D_CL partition scheme

Two devices, two partitions per device: 'FFC coefficients' and 'Image data'

Applies to the following firmware variants of PC3602-T Coaxlink Octo:

 (2-camera)

	FFC coefficients	Image data
Device 0	1/4	1/4
Device 1	1/4	1/4

2D_FFC partition scheme

Three devices, one 'Image data' partition per device

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value:

 (3-camera)

 (3-camera)


 (3-camera)


	Image data
Device 0	1/2
Device 1	1/4
Device 2	1/4

3D partition scheme

Four devices, one 'Image data' partition per device

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value:

 (4-camera), (4-camera, line-scan)

 (4-camera), (4-camera, line-scan)


 (4-camera), (4-camera, line-scan)

	Image data
Device 0	1/4
Device 1	1/4
Device 2	1/4
Device 3	1/4

4D partition scheme

Four data streams, one 'Image data' partition per data stream

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3:


 (1-camera, 4-data-stream)

	Image data
Stream0	1/4
Stream1	1/4
Stream2	1/4
Stream3	1/4

4S partition scheme

Five devices, one 'Image data' partition per device, 5D22211 scheme

Applies to the following firmware variants of PC3602-T Coaxlink Octo:


 (5-camera, 5D22211)

	Image data
Device 0	1/4
Device 1	1/4
Device 2	1/4
Device 3	1/8
Device 4	1/8

5D22211 partition scheme

Five devices, one 'Image data' partition per device, 5D41111 scheme

Applies to the following firmware variants of PC3602-T Coaxlink Octo:

 (5-camera)

	Image data
Device 0	1/2
Device 1	1/8
Device 2	1/8
Device 3	1/8
Device 4	1/8

5D4111 partition scheme

Six devices, one 'Image data' partition per device, 6D221111 scheme

Applies to the following firmware variants of PC3602-T Coaxlink Octo:



	Image data
Device 0	1/4
Device 1	1/4
Device 2	1/8
Device 3	1/8
Device 4	1/8
Device 5	1/8

6D22111 partition scheme

Eight devices, one 'Image data' partition per device

Applies to the following firmware variants of PC3602-T Coaxlink Octo:


 (8-camera), (8-camera, line-scan)

	Image data
Device 0	1/8
Device 1	1/8
Device 2	1/8
Device 3	1/8
Device 4	1/8
Device 5	1/8
Device 6	1/8
Device 7	1/8

8D partition scheme

4. Acquisition Gate

The *Acquisition Gate* controls the image data extraction from the *Image data* partition of the on-board memory. It discards the image data that doesn't need to be acquired and fed to the "Pixel Processing" on page 107 chain.

Area-scan acquisition

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



NOTE

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

See also: "Area-scan Acquisition" on page 88 for more information and configuration instructions.






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.

See also: "Line-scan Acquisition" on page 93 for more information and configuration instructions.

5. Area-scan Acquisition

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value:

-  (1-camera), (1-camera, 4-data-stream), (2-camera), (2-camera, bayer), (3-camera), (4-camera)
-  (1-camera), (1-camera, custom-logic), (2-camera), (3-camera), (4-camera), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera)
-  (1-camera)
-  (1-camera), (2-camera)
-  (1-camera), (2-camera), (3-camera), (4-camera)

5.1. Area-scan Acquisition Principles	89
5.2. High Frame Rate Acquisition	90
5.3. Multi-Stream Acquisition	92

5.1. Area-scan Acquisition Principles

Area-scan imaging

The expression “Area-scan imaging” designates machine vision applications where images are obtained from a camera delivering 1 image frame every camera cycle.

GenTL buffer filling rules – Area-scan firmware variants

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

- The first acquired line data of a frame is, by default, stored at the beginning of a new buffer. When vertical image flipping is enabled by setting **StripeArrangement** to **Geometry_1X_1YE**, the first acquired line data of a frame is stored at the location of the last full line of a new buffer.
- When image transfer to host memory is done, the buffer, possibly partially filled, is made available to the application for processing.
- When the remaining space of a buffer is not sufficient to store a complete frame, the remaining data is handled according to the "**Area-scan Acquisition Principles**" on page 89.

5.2. High Frame Rate Acquisition

The High Frame Rate –HFR– feature allows area-scan applications to store more than one image per buffer.

Why high frame rate acquisition?

The processing overhead linked to the buffer management prevents the Host PC to sustain image acquisition at high frame rates. The upper limit depends on the Host PC; the 'grey' area is typically in the 1 kHz to 5 kHz range.

Storing several images per buffer significantly reduces the processing overhead and enables area-scan applications to reach very high acquisition frame rates in excess of several hundred thousand images per second!

Configuring the number of images to put in one buffer

The **BufferPartCount** GenApi feature of the Data Stream module defines the number of images to put in one buffer.

By default, **BufferPartCount** is 1. The maximum value is 10,000.

Using larger values is recommended for high frame rate applications. The value should be large enough to keep the buffer handling rate below the upper limit sustainable by the Host PC!

The data stream's Height is set to **BufferPartCount** * the camera's Height



NOTE

The value of **BufferPartCount** is only used when the buffer is announced.

Not available on line-scan firmware variants

The data stream's Height is set to **BufferPartCount** * the camera's Height

See also: "310-high-frame-rate" sample program

Managing HFR acquisition

The following commands were added to BUFFER_INFO_CUSTOM_CMD_LIST:

BUFFER_INFO_CUSTOM_PART_SIZE

BUFFER_INFO_CUSTOM_NUM_PARTS

BUFFER_INFO_CUSTOM_NUM_DELIVERED_PARTS

BUFFER_INFO_CUSTOM_PART_TIMESTAMPS

Use BUFFER_INFO_CUSTOM_NUM_DELIVERED_PARTS to get the number of parts available in the buffer


See also: "311-high-frame-rate" sample program

Use BUFFER_INFO_CUSTOM_PART_TIMESTAMPS64-bit array to get the timestamps of each buffer part

See also: "312-high-frame-rate" sample program

5.3. Multi-Stream Acquisition

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3:

 (1-camera, 4-data-stream)

4-data-stream concurrent acquisition

The *1-camera, 4-data-stream* firmware variant of above-listed products allows to connect one area-scan CoaXPress camera that delivers up to 4 independent data streams.

The frame grabber sorts the incoming CoaXPress data blocks according to the value of the 2 least significant bits of the CoaXPress StreamID and feeds four independent data paths.

Each data path is capable of handling the full CoaXPress link bandwidth, namely 2.5 Gigabytes/s.






It includes:

1. A 256 MB **image data partition**,
2. A pixel processor that allows to align 10/12/14-bit data to the LSB or to the MSB of a 16-bit container,
3. A DMA engine that transfers image data directly to the user memory space of the Host PC.

The multistream sample program shows how to create 4 instances of EGrabber and start acquisition on 4 concurrent data streams.

6. Line-scan Acquisition

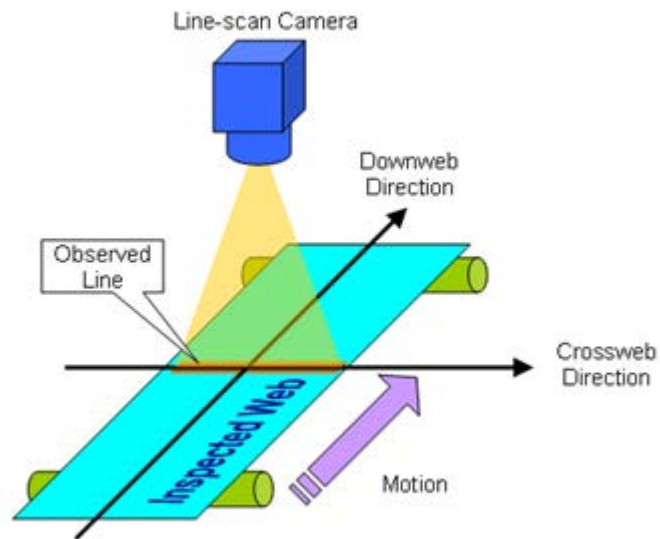
Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value:

-  (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)
-  (1-camera, line-scan), (2-camera, line-scan), (2-camera, line-scan, custom-logic), (4-camera, line-scan), (8-camera, line-scan)
-  (1-camera, line-scan)
-  (1-camera, line-scan), (2-camera, line-scan)
-  (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)

- 6.1. Line-scan Acquisition Principles 94
- 6.2. Line-scan Acquisition Use cases 98
- 6.3. Metadata Insertion 102
- 6.4. Trigger to Line Latency Compensation 106

6.1. Line-scan Acquisition Principles

Line-scan imaging



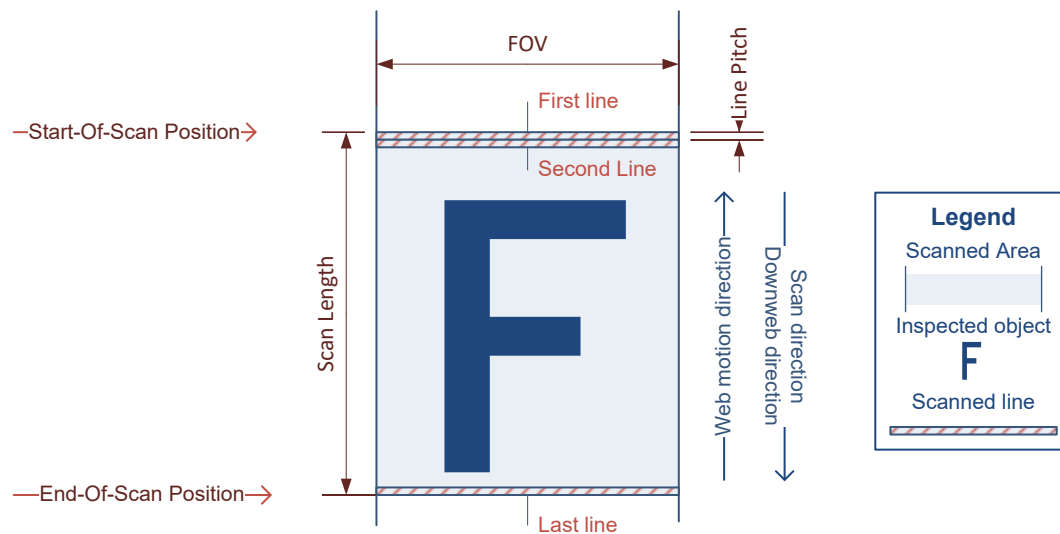
Typical line-scan imaging system

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



Scanning area definitions

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 pitch¹ 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.

NOTE: In this context, the "line pitch" term is a geometrical property. In other sections of this documentation, the term LinePitch² is a memory offset between two consecutive lines in a GenTL Buffer.

¹In the imaging device context, refers to the distance between two consecutive scan lines in an image sensor. For line-scan imagers, "Line Pitch" is determined by the relative displacement between the camera and the object during a camera cycle period.

²In the context of a GenTL image buffer, represents the memory offset between two adjacent pixels along a vertical axis. Also known as buffer width.

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 following methods:

Constant camera cycle method

In this method, the camera operates at a constant cycle rate and the frame grabber captures all the successive lines delivered by the camera.

Variable camera cycle method

In this method, 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.

This 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**" on page 258 decimates the input rate by an integer value, it delivers 1 out of N incoming events.
- The "**Multiplier/Divider Tool**" on page 259 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



Image capture of small and large objects

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 is low: “one image line”.

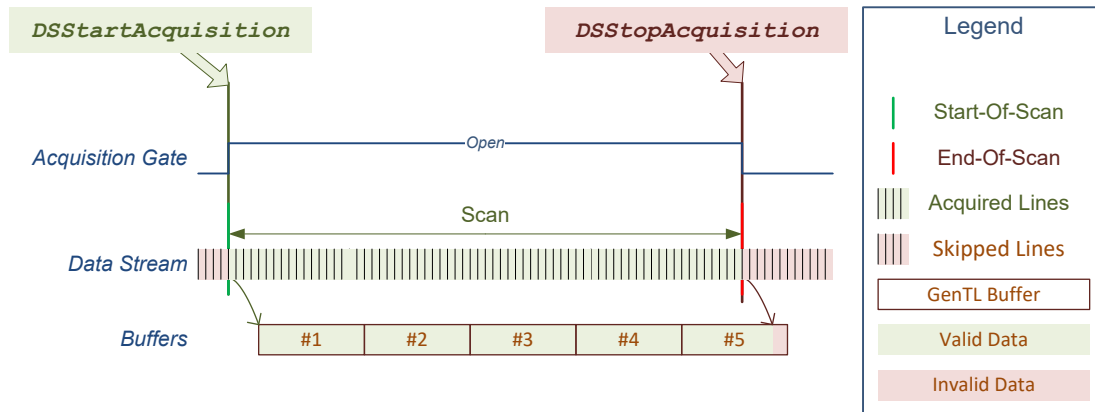
GenTL buffer filling rules – Line-scan firmware variants

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

- The first acquired line data of a scan is, by default, stored at the beginning of a new buffer. When vertical image flipping is enabled by setting **StripeArrangement** to **Geometry_1X_1YE**, the first acquired line data of a scan is stored at the location of the last full line 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 remaining data is handled according to the "**Line-scan Acquisition Principles**" on page 94.
- When the last line data of a scan is acquired, the last buffer, possibly partially filled, is made available to the application for processing.

6.2. Line-scan Acquisition Use cases

Scanning of continuous objects



Scanning of continuous objects

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

The acquisition controller is configured as follows:

- **StartOfScanTriggerSource** = **Immediate**.
- **EndOfScanTriggerSource** = **StopScan**.

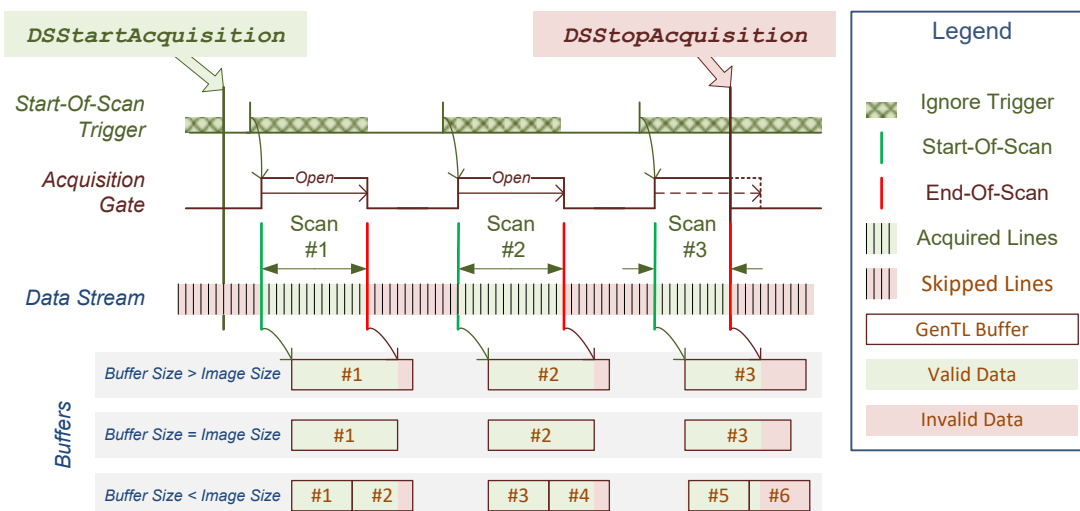
When the **DSStartAcquisition** function is called, the scanning starts at the next line boundary.

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



Scanning of discrete objects with a common scan length

The **eGrabber** acquisition controller is configured as follows:

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

When the **DSSStartAcquisition** 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**.

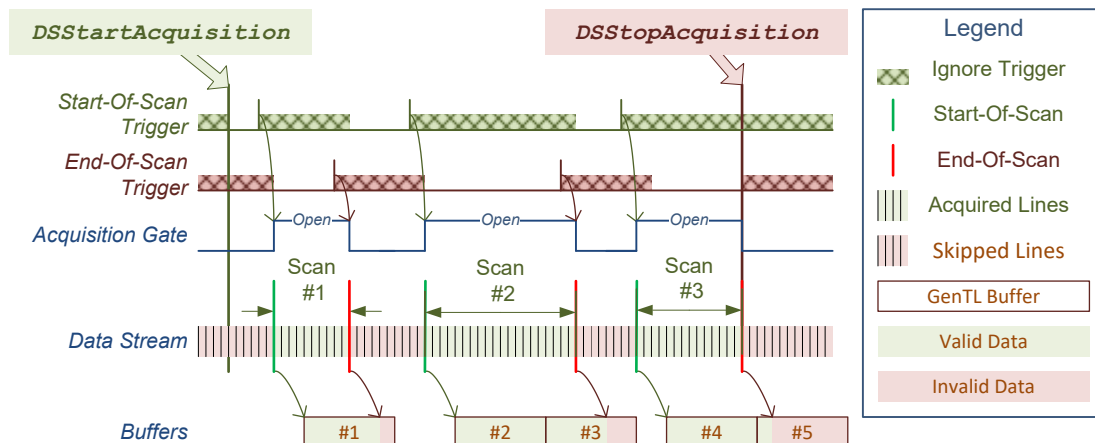
The acquisition controller ignores any Start-of-Scan trigger event while a scanning is in progress.

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 **DSSStopAcquisition** 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



Scanning of discrete objects requiring a variable scan length

The **eGrabber** 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 **DStopAcquisition** 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.

6.3. Metadata Insertion

Applies to the following firmware variants of **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan), (8-camera, line-scan)

 (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)

Introduction

The Metadata Insertion feature allows line-scan applications to insert metadata into a buffer.

Two types of metadata can be inserted into a buffer:

- *Buffer metadata*: 4x32-bit metadata are inserted into an internal header information of the buffer;
- *Line metadata*: 4x32-bit metadata are inserted at the end of each image line within the buffer.

These two types of metadata can be inserted independently or simultaneously.

Requirements

Only available on grabber-controlled cycle-start camera control methods (RG and RC).

The metadata are sampled on the **CycleStart** events of the CIC:

- Buffer metadata are sampled on the **CycleStart** event that initiates the first line of the buffer
- Line metadata are sampled on each **CycleStart** event

Configuration

The **MetadataInsertion** category of the Data Stream module contains the features for configuring metadata insertion.

Each **MetadataInsertion** feature responsible for configuring metadata insertion must be set before starting an acquisition.

When an acquisition is started, the features are locked.

Activation of the metadata

- **BufferMetadataInsertionEnable**: Boolean feature to enable/disable the insertion of 4x32-bit (16 bytes) metadata for the buffer
- **LineMetadataInsertionEnable**: Boolean feature to enable/disable the insertion of 4x32-bit (16 bytes) metadata at the end of each image line

Enabling the insertion of buffer metadata does not affect the payload size of the buffer.

However, enabling the insertion of line metadata does affect the payload size of the buffer.

When `LineMetadataInsertionEnable = True`, the data stream feature `LineWidth` is automatically increased by 16 to count for the size of the line metadata, which in turn may increase the data stream `PayloadSize` (or consume some of the padding part of the buffer, if used).

NOTE: the size of the padding on a line, is always the difference between `LinePitch` and `LineWidth`, if `LinePitch > LineWidth`. When padding is added together with line metadata insertion, the padding part is located after the line metadata (image line - line metadata - padding).

See also: Image Data Padding.



TIP

Enable "**Trigger to Line Latency Compensation**" on page 106 to properly correlate line-data with line-triggers.

Contents of the metadata

The metadata can contain GPC values, QDC positions, and/or I/O line states.

The contents of the metadata are configured with the features `MetadataContent<N>`, where **N** represents the offset, from 0 to 3, of one 32-bit metadata within a chunk of 128-bit metadata.

These four features describe the contents to be reported in both the buffer and line metadata.

The features can report the following contents:

`MetadataContent0`

- `GPC1Value`: Value of General Purpose Counter 1
- `GPC1LatchedValue`: Latched value of General Purpose Counter 1
- `QDC1Position`: Position of Quadrature Decoder Tool 1

`MetadataContent1`

- `GPC2Value`: Value of General Purpose Counter 2
- `GPC2LatchedValue`: Latched value of General Purpose Counter 2
- `QDC2Position`: Position of Quadrature Decoder Tool 2 (if available)

`MetadataContent2`

- `GPC3Value`: Value of General Purpose Counter 3
- `GPC3LatchedValue`: Latched value of General Purpose Counter 3
- `QDC3Position`: Position of Quadrature Decoder Tool 3 (if available)
- `LineStatusAllHi`: High 32-bit part of `LineStatusAll`

`MetadataContent3`

- `GPC4Value`: Value of General Purpose Counter 4
- `GPC4LatchedValue`: Latched value of General Purpose Counter 4
- `QDC4Position`: Position of Quadrature Decoder Tool 4 (if available)
- `LineStatusAll`: Low 32-bit part of `LineStatusAll`

On `CycleStart` event, the selected contents are sampled. Their values will be inserted in the buffer and/or the line metadata.

General Purpose Counter (GPC)

Four 32-bit *General Purpose Counters* can be used in various ways, e.g., to count the number of occurrences of a particular event, to implement a differential counter between two event streams, or even to latch a counter on a particular event. The latch functionality can be useful for permanent period measurements.

The GPCs are selected through `GeneralPurposeCounterSelector` and enabled/disabled with `GeneralPurposeCounterEnable`.

Up to three event sources can be set to define the behavior of a GPC:

- `GeneralPurposeCounterIncrementSource`: selects an event stream used as trigger to increment the selected GPC
- `GeneralPurposeCounterDecrementSource`: selects an event stream used as trigger to decrement the selected GPC
- `GeneralPurposeCounterLatchAndResetSource`: selects an event stream used as trigger to latch and reset the selected GPC

When `GeneralPurposeCounterEnable=False`, the selected counter is reset.

Example 1

To count the difference between `CycleStart` and `StartOfLine` with GPC1:

- Set `GeneralPurposeCounterSelector` to `GPC1`.
- Set `GeneralPurposeCounterIncrementSource` to `CycleStart`.
- Set `GeneralPurposeCounterDecrementSource` to `StartOfLine`.
- Set `GeneralPurposeCounterLatchAndResetSource` to `NONE`.
- Set `GeneralPurposeCounterEnable` to `True`.

To report the value of GPC1 in the metadata:

- Set `MetadataContent0` to `GPC1Value`.

NOTE: the value of this GPC in the metadata will always be greater than 0. That is because the GPC is sampled on `CycleStart`, while `StartOfLine` will always happen after `CycleStart`.

Example 2

To measure the cycle period with GPC2:

- Set `GeneralPurposeCounterSelector` to `GPC2`.
- Set `GeneralPurposeCounterIncrementSource` to `TIME16NS`.
- Set `GeneralPurposeCounterDecrementSource` to `NONE`.
- Set `GeneralPurposeCounterLatchAndResetSource` to `CycleStart`.
- Set `GeneralPurposeCounterEnable` to `True`.

To report the latched value of GPC2 in the metadata:

- Set `MetadataContent1` to `GPC2LatchedValue`.

Getting metadata

- To get the buffer metadata of a buffer, use the commands `BUFFER_INFO_CUSTOM_BUFFER_METADATA_<N>` from `BUFFER_INFO_CUSTOM_CMD_LIST`, where N is the offset from 0 to 3.

The commands return the 32-bit inserted buffer metadata at offset N.

If no buffer metadata was inserted in the buffer, `BUFFER_INFO_CUSTOM_BUFFER_METADATA_<N>` commands return `GC_ERR_NOT_AVAILABLE`.

- To get the line metadata of a buffer, use the command `BUFFER_INFO_CUSTOM_LINE_METADATA_BASE` from `BUFFER_INFO_CUSTOM_CMD_LIST`.

The command returns a pointer to the base address of the inserted line metadata.

If no line metadata was inserted in the buffer, `BUFFER_INFO_CUSTOM_LINE_METADATA_BASE` command returns 0.

- To get the type of content of an inserted metadata in a buffer, use the commands `BUFFER_INFO_CUSTOM_METADATA_CONTENT_<N>` from `BUFFER_INFO_CUSTOM_CMD_LIST`, where N is the offset from 0 to 3.

The commands return an integer that identify the type of content that the buffer and/or line metadata contain at offset N.

The values returned by the `BUFFER_INFO_CUSTOM_METADATA_CONTENT_<N>` commands are translated as follows:


Value	<code>BUFFER_INFO_CUSTOM_METADATA_CONTENT_0</code>	<code>BUFFER_INFO_CUSTOM_METADATA_CONTENT_1</code>	<code>BUFFER_INFO_CUSTOM_METADATA_CONTENT_2</code>	<code>BUFFER_INFO_CUSTOM_METADATA_CONTENT_3</code>
1	GPC1Value	GPC2Value	GPC3Value	GPC4Value
2	GPC1LatchedValue	GPC2LatchedValue	GPC3LatchedValue	GPC4LatchedValue
3	QDC1Position	QDC2Position	QDC3Position	QDC4Position
4			LineStatusAllHi	LineStatusAll


If neither buffer nor line metadata were inserted, `BUFFER_INFO_CUSTOM_METADATA_CONTENT_<N>` commands return `GC_ERR_NOT_AVAILABLE`.

See also: [cpp/330-metadata-insertion sample program](#)

6.4. Trigger to Line Latency Compensation

Applies to the following firmware variants of **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan), (8-camera, line-scan)

 (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)

What is trigger to line latency?

As described in the "**Camera Control Principles**" on [page 188](#) section, a camera cycle is composed of two consecutive phases: Exposure and Readout.

Some cameras require additional time to process and transmit the processed line data to the frame grabber. This time is named *trigger to line latency*.

When the latency becomes significant (when compared to the camera cycle time) the association between a line trigger and the corresponding data is not anymore possible.

Line trigger latency compensation logic

When enabled, the compensation logic evaluates the delay and retards the execution of the `StartOfScan` and/or the `EndOfScan` events to ensure the line data are correlated.

Latency compensation control

The *line trigger latency compensation logic* is configurable via the **TriggerToLineLatencyCompensation** GenApi feature in the DataStream Module.

Possible values are:

- **Disable**: Disable latency compensation
- **Enable**: Enable latency compensation for `StartOfScan` and `EndOfScan` events.
- **EnableForStartOfScanOnly**: Enable latency compensation for `StartOfScan` events only.
- **EnableForEndOfScanOnly**: Enable latency compensation for `EndOfScan` events only.

The default value is **Disable**.



TIP

Enable line trigger latency compensation when **metadata insertion** is enabled!

7. Pixel Processing

7.1. Overview	107
7.2. Configurations	110
7.3. Pixel Unpacking and Alignment	118
7.4. Flat Field Correction	120
7.5. Lookup Table Processing	133
7.6. Bayer CFA Decoding	146
7.7. Pixel Binning	157
7.8. Pixel Components Swapping	163
7.9. Endianness Conversion	164
7.10. Pixel Ordering	165

7.1. Overview

Pixel processing overview of Coaxlink frame grabbers

The Image Pixel Data Processor performs the following successive operations on the image data stream:

CoaXPress bit stream slicing

This operation extracts individual pixel components data from the CoaXPress image data bit stream according to the bit depth – *input-bit-depth* – specified by the '*PixelF*' property of the CoaXPress Image Header.

All components have the same pixel bit depth. Possible values are 8-/10-/12-/14- and 16-bit.

The slicer delivers, for each image line, all the pixel components necessary to build a number of pixels specified by the '*Xsize*' property of the CoaXPress Image Header.

The slicer discards CoaXPress line-padding data.

Unpacking — UNP

This operation unpacks 10-bit, 12-bit, and 14-bit pixel components to 16-bit.

It can be disabled for monochrome and Bayer CFA pixel formats.

See also: "Pixel Unpacking and Alignment" on page 118 for more information and configuration instructions.

Flat Field Correction — FFC

This operation applies a linear gain and offset transformation to each individual pixel components.

See also: "Flat Field Correction" on page 120 for more information and configuration instructions.

Lookup Table processing — LUT

This operation applies a lookup table transformation to each individual pixel components.

See also: "Lookup Table Processing" on page 133 for more information and configuration instructions.

Alignment — ALI

This operation align 10-bit, 12-bit, and 14-bit pixel components to lsb or msb in the 16-bit container.

See also: "Pixel Unpacking and Alignment" on page 118 for more information and configuration instructions.

Bayer CFA decoding — CFA

This operation transforms the raw Bayer CFA data stream issued by the camera into an RGB color data stream.

See also: "Bayer CFA Decoding" on page 146 for more information and configuration instructions.

Pixel Binning — BIN

This operation combines a cluster of 2 x 2 or 4 x 4 pixels into a single pixel by summing or averaging their respective pixel values.

See also: "Pixel Binning" on page 157 for more information and configuration instructions.

Pixel component swapping — SWAP

This operation modifies the component order of multi-components pixel data.

See also: "Pixel Components Swapping" on page 163 for more information and configuration instructions.

Endianness conversion

This operation modifies the byte order of 16-bit pixel component data.

See also: "Endianness Conversion" on page 164 for more information.

Image line build-up

This operation builds concatenates the components data of all pixels of an image line:

- 8-bit pixel components are aligned to byte boundaries
- 16-bit pixel components (possibly expanded by unpacking or lookup table processing) are aligned to word (2-byte) boundaries, the 2 bytes are stored according to the little-endian convention.

Line padding

This operation appends padding bits or bytes to the image line data to reach the next alignment-boundary required by the hardware implementation.

The alignment boundary requirements are product-specific.

The alignment boundary requirements are product-specific, for instance:

- 128-bit for **PC1633-T Coaxlink Quad G3**

Processing Performances

The pixel processor sustain the highest camera pixel rate. Unless specified otherwise, all the above operations are executed while transferring data to the GenTL with a negligible latency.

PCI Express Bandwidth Limitation

When acquiring pixels having a pixel bit depth larger than 8-bit, each pixel is expanded to 16-bit. In these cases, the PCI Express bandwidth limitation of the Host PC may negatively impact the achievable frame- or line-rate.

On-board Memory Bandwidth Limitation

For **FFC use cases**, the on-board memory bandwidth is not sufficient to sustain the full CoaXPress data rate.

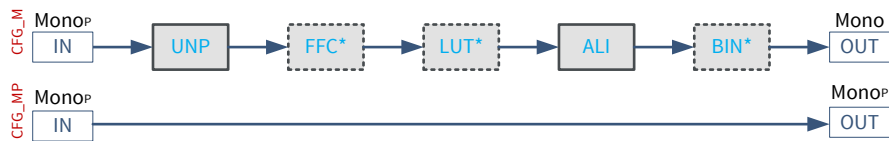
7.2. Configurations

Pixel processing configurations of Coaxlink frame grabbers

This topic shows the applicable pixel data processing configurations of **Coaxlink frame grabbers** for every class of camera pixel formats. For each class, a **drawing** shows the relevant pixel data processing configurations:

- "Configurations for monochrome cameras" on page 111
- "Configurations for Bayer CFA cameras" on page 112
- "Configurations for RGB cameras" on page 115
- "Configurations for RGBa cameras" on page 116

Configurations for monochrome cameras



CFG_M configuration



In the "CFG_M" configuration, the pixel processing chain transform *packed monochrome pixels* into *monochrome* pixels.

The successive processing steps are:

- ❑ UNP: unpacking of 10-, 12- and 14-bit pixels to 16-bit with alignment to lsb.
- ❑ FFC: flat-field correction.
- ❑ LUT: look-up table. Applies only to 8-, 10- and 12-bit pixels!
- ❑ ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixels.
- ❑ BIN: binning.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC, LUT and BIN are optional processing steps that can be activated or not according to the application needs.

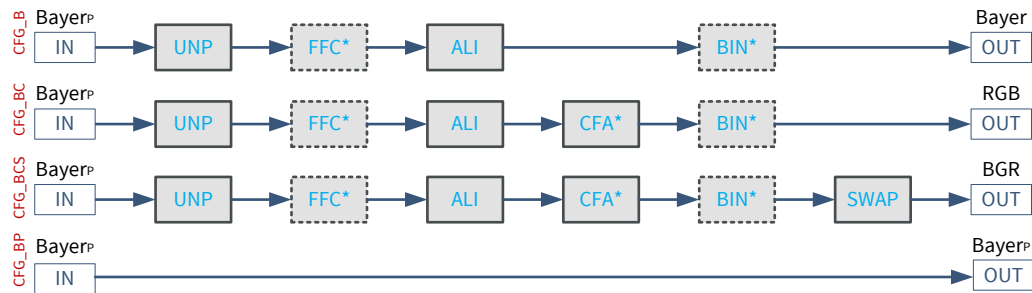
CFG_MP configuration



In the "CFG_MP" configuration, pixel unpacking is turned off.

The pixel processing chain is disabled. Pixels are not processed; packed pixels remain unpacked.

Configurations for Bayer CFA cameras



CFG_B configuration



In the "CFG_B" configuration, the CFA and the SWAP processing steps are disabled. The pixel processing chain transform *packed Bayer CFA pixels* into *Bayer pixels*.

The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixels to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixels.
- BIN: binning.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC and BIN are optional processing steps that can be activated or not according to the application needs.

CFG_BC configuration



In the "CFG_BC" configuration, the CFA processing step is enabled and the SWAP processing step is disabled. The pixel processing chain transform *packed Bayer CFA pixels* into *RGB* pixels.

The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixels to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixels.
- CFA: Bayer CFA decoding.
- BIN: binning.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC and BIN are optional processing steps that can be activated or not according to the application needs.

CFG_BCS configuration



In the "CFG_BCS" configuration, the CFA and the SWAP processing steps are enabled. The pixel processing chain transform *packed Bayer CFA pixels* into *BGR* pixels.

The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixels to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixels.
- CFA: Bayer CFA decoding.
- BIN: binning.
- SWAP: R/B pixel component swapping.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

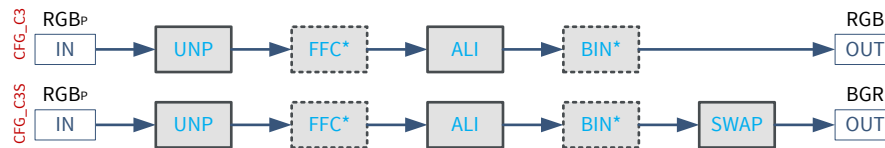
FFC and BIN are optional processing steps that can be activated or not according to the application needs.

CFG_BP configuration



In the "CFG_BP" configuration, pixel unpacking is turned off. The pixel processing chain is disabled. Pixels are not processed. Packed pixels remain unpacked.

Configurations for RGB cameras



CFG_C3 configuration



In the "CFG_C3" configuration, the SWAP processing step is disabled. The pixel processing chain transform *packed RGB pixels* into *RGB* pixels.

The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixel components to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixel components.
- BIN: binning.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC and BIN are optional processing steps that can be activated or not according to the application needs.

CFG_C3S configuration



In the "CFG_C3S" configuration, the SWAP processing step is enabled. The pixel processing chain transform *packed RGB pixels* into *BGR* pixels.

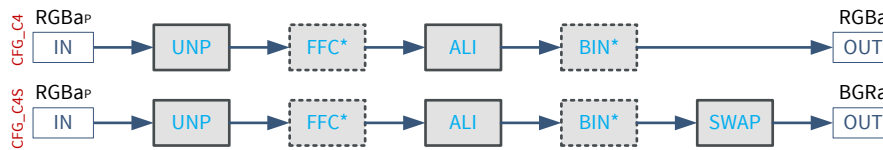
The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixel components to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixel components.
- BIN: binning.
- SWAP: R/B pixel component swapping.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC and BIN are optional processing steps that can be activated or not according to the application needs.

Configurations for RGBa cameras



CFG_C4 configuration



In the "CFG_C4" configuration, the SWAP processing step is disabled. The pixel processing chain transform *packed RGBa pixels* into *RGBa pixels*.

The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixel components to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixel components.
- BIN: binning.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC and BIN are optional processing steps that can be activated or not according to the application needs.

CFG_C3S configuration



In the "CFG_C4S" configuration, the SWAP processing step is enabled. The pixel processing chain transform *packed RGBa pixels* into *BGRa pixels*.

The successive processing steps are:

- UNP: unpacking of 10-, 12- and 14-bit pixel components to 16-bit with alignment to lsb.
- FFC: flat-field correction.
- ALI: alignment to lsb (default) or msb of 10-, 12- and 14-bit pixel components.
- BIN: binning.
- SWAP: R/B pixel component swapping.

UNP and ALI are *mandatory processing step* that must be active and configured according to the application needs.

FFC and BIN are optional processing steps that can be activated or not according to the application needs.

Drawing conventions

The pixel processing configuration drawings use the following conventions:

- Solid rectangle: *mandatory processing step* that must be active and configured according to the application needs.
- Dashed rectangle: *optional processing step* that can be activated or not according to the application needs.
- Names with a * (e.g. LUT*): processing step that is only available on selected products and firmware variants
- Names without *: processing step that is always available

7.3. Pixel Unpacking and Alignment

Introduction

The pixel data processor is capable of unpacking and aligning 10-bit, 12-bit, and 14-bit pixel component data to 16-bit data containers.

The unpacking and the alignment operations are user-configurable through the **UnpackingMode** GenApi feature.

On **Coaxlink frame grabbers**, three options are available:

- **Lsb**: Unpacking and alignment to lsb (Default setting since release 4.3)
- **Msb**: Unpacking and alignment to msb
- **Off**: No unpacking

Unpacking and alignment 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.

- 10-bit pixels: 0000 00<pp pppp pppp>
- 12-bit pixels: 0000 <pppp pppp pppp>
- 14-bit pixels: 00<pp pppp pppp pppp>

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

NOTE: Unpacking 10-bit, 12-bit, and 14-bit pixel components increases the amount of data by 160%, 133%, and 114% 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) and the data bits are not modified.

Unpacking and alignment 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.

- 10-bit pixels: <pppp pppp pp>00 0000
- 12-bit pixels: <pppp pppp pppp> 0000
- 14-bit pixels: <pppp pppp pppp pp>00

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

NOTE: Unpacking 10-bit, 12-bit, and 14-bit pixel components increases the amount of data by 160%, 133%, and 114% 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) and the data bits are not modified.

No unpacking (Coaxlink frame grabbers only)


The packed image data transmitted by the camera through the CoaXPress Link is delivered as is to the output buffer.

NOTE: Keeping packed image data for 10-bit, 12-bit, and 14-bit pixels avoids wasting PCI bandwidth for the transmission of padding zero's.


NOTE: This option is only available for cameras delivering monochrome and Bayer CFA pixels when pixel processing is bypassed!

7.4. Flat Field Correction

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera), (1-camera, line-scan)

 (2-camera)

 (1-camera), (1-camera, line-scan)

What is Flat Field Correction?	121
FFC IP Core Description	124
FFC Wizard Sample Program	127

What is Flat Field Correction?

The *Flat-field correction* ([Wikipedia: FFC](#)) is a method used to correct:

- the differences of light sensitivity between the pixel sensors of a camera
- some artifacts related to the optical system (e.g., non-uniform lighting and [vignetting](#))

The goal is to correct the pixels of the captured (raw) images in such a way that when a uniform background is captured by the system (camera & lens), the resulting output image is uniform.

Formula

This correction is achieved by applying the following operation to each pixel of the raw image:

$$\text{CorrectedPixel} = (\text{RawPixel} - \text{Offset}) * \text{Gain}$$

where both *Offset* and *Gain* coefficients are specific values for each pixel.

The evaluation of the coefficients *Offset* and *Gain* requires a calibration procedure explained in the next paragraph.

Calibration

The calibration procedure to compute the coefficients is done in two steps:

Dark image acquisition

A dark image is acquired by the system. This is typically achieved by covering the lens with the cap. The captured image represents the dark current of the sensors, and is considered as a fixed bias that we want to eliminate when acquiring images in normal conditions. This correction is called dark-frame subtraction.

$$\text{CorrectedPixel} = \text{RawPixel} - \text{DarkPixel}$$

For each pixel, the *DarkPixel* value corresponds to the *Offset* in the above FFC.

Flat image acquisition

A flat image is acquired by the system. For example by capturing a flat (uniform) background, not too bright to avoid saturation and not too dark.

From dark and flat acquisitions, we have enough data to compute the *Gain* value of the FFC.

For this we define *CorrectedPixel* as the pixel value we consider to be correct for the flat image. Let's set this value as the average pixel value of the flat image (*average(Flat)*), corrected by the average of the dark image (*average(Dark)*). In the FFC terms, this gives:

$$\text{average(Flat)} - \text{average(Dark)} = (\text{FlatPixel} - \text{DarkPixel}) * \text{Gain}$$

This leads to the Gain value

$$\text{Gain} = (\text{average(Flat)} - \text{average(Dark)}) / (\text{FlatPixel} - \text{DarkPixel})$$

The same computation is repeated (*width * height*) times, to cover all pixels of both flat and dark images. This results in a specific correction for each pixel of the image.



NOTE

Note: this calibration procedure must be redone if any part of the system is changed, including the camera unit, lighting or optics equipment.

Calibration of color pixel formats

For color pixel formats, we have several ways of computing the value of *average(Flat)*. In all cases, the *Gain* computation is repeated (*width * height * componentsPerPixel*) times to cover all pixel components, which results in specific corrections for each pixel component of the image.

Handling pixel components individually

I.e. in RGB:

- using *average(Flat[Red])* for computing the *Gain* values of *Red* components;
- using *average(Flat[Green])* for computing the *Gain* values of *Green* components;
- using *average(Flat[Blue])* for computing the *Gain* values of *Blue* components.

Handling pixel components together

I.e. in RGB, using *average(average(Flat[Red]), average(Flat[Green]), average(Flat[Blue]))* for computing the *Gain* values of *Red*, *Green* and *Blue* components.

This way of computing the average (i.e. over the pixel components) results in FFC coefficients that also correct the balance between components. Therefore, depending on the quality of the uniform background used to acquire the flat image, the FFC can effectively perform a white balance correction.

FFC IP Core Description

*This topic describes the FFC IP core implemented in the **AVT** frame grabbers FPGA*

The FFC IP core corrects the pixels directly coming from the camera by applying the FFC using the coefficients (*Offset* and *Gain*) corresponding to their locations in the image. Because the correction happens at a very early stage in the pixel processing chain, the other pixel processing functions such as *RedBlueSwap*, *LUT*, and *Bayer Decoding* are performed on corrected pixels.

Gain and Offset Coefficients Format

The coefficients calculated in the calibration procedure can be loaded into the frame grabber, provided they are encoded as follows:

- *Offset* and *Gain* values for one pixel component are packed together into a 16-bit little-endian value:
 - *Gain* is encoded in [Wikipedia:UQ2.8](#) on bits 9..0
 - *Offset* is a 6-bit unsigned integer on bits 15..10
- Coefficients related to pixel component values are treated separately in the same sequence as the pixel components of the image. For example in *RGB8* format, one pixel is encoded as 3 successive 8-bit values (*Red*, *Green*, *Blue*), therefore we need 3 successive 16-bit packed coefficients to correct one *RGB8* pixel.

If the 16-bit packed coefficients are stored (in sequence) in a binary file (let's say '*path/to/coefficients.ffc*'), they can be easily loaded from a **AVT** script by calling `require("coaxlink://ffc/load")(grabber, 'path/to/coefficients.ffc');` where *grabber* is the script variable referencing the grabber to configure.

NOTE: such a binary file can be created by the **AVT ffc-wizard sample application**.

Specifications

Camera Types

The FFC feature is applicable to monochrome, Bayer CFA and RGB Color area-scan or line-scan cameras delivering 8-, 10-, 12- 14- or 16-bit data per pixel component.

Maximum Image Size

The maximum size is determined by the size of the memory used to store *FFC coefficients* and the pixel format:

Size	Maximum Image size [Pixels]		
	Monochrome and Bayer CFA	RGB	RGBa
32 KB	16,384	5,461	4,096
512 MB	268,435,456	89,478,485	67,108,864
1024 MB	536,870,912	178,956,970	134,217,728
2048 MB	1,073,741,824	357,913,941	268,435,456

The location and the size of the *FFC coefficients* storage are determined by the product/firmware variant combination:

Product/firmware-variant	Location (Partition scheme)	Size
PC1633-T Coaxlink Quad G3 (1-camera, line-scan)	FPGA memory	32 KB
PC1633-T Coaxlink Quad G3 (1-camera)	DRAM memory	512 MB
PC3623-T Coaxlink Quad CXP-12 Value (1-camera)	(1D_FFC)	2048 MB
PC3602-T Coaxlink Octo(2-camera)	DRAM memory (2D_FFC)	512 MB



NOTE

For most product/firmware-variants, *FFC coefficients* are stored in a dedicated partition of the on-board DRAM memory.

However, for the line-scan firmware variants of **PC1633-T Coaxlink Quad G3**, the *FFC coefficients* are stored into internal FPGA memory blocks. This allows the FIFO buffer to work at full performance when the FFC is enabled.

Performance

When enabled, the FFC feature adds a significant load to the one-board memory since it fetches an additional 16-bit coefficient data for each processed pixel. When the FFC is enabled, the frame grabber is only able to sustain a fraction of the maximum data rate achievable by the CoaXPress Link. This dimension-less value is named "*Sustainable relative data rate*"

The following table shows the *sustainable relative data rate* for all bit depths and product/firmware variant combinations supporting FFC.

Sustainable Relative Data Rate

Bit depth	Sustainable Relative Data Rate [% of a 4-lane CXP-6 maximum data rate]	
Firmware	PC1633-T Coaxlink Quad G3 (1-camera)	PC3602-T Coaxlink Octo (2-camera)
8-bit	70.0 %	123.2 %
10-bit	77.8 %	136.9 %
12-bit	84.0 %	147.8 %
14-bit	89.1 %	156.8 %
16-bit	93.3 %	164.3 %



NOTE

The "Sustainable Relative Data Rate" is global for all cameras attached to a board. For instance, in the **PC3602-T Coaxlink Octo** - 2-camera 8-bit use case, the sustainable data rate of 123.2% can be split unequally to 100% for a camera and 23.2% for the other camera.



NOTE

eGrabber-driven frame grabbers do not acquire any data during blanking intervals. Line- and frame-blanking intervals do not consume memory bandwidth and therefore must be excluded in the calculation of the camera data rate.



TIP

To avoid latencies, FIFO buffer overflow and loss of frames, **AVT** recommends to limit the (global) camera data rate accordingly.

Enabling the FFC

In the Data Stream module, set the **FfcControl** feature value to **Enable**.

Disabling the FFC

In the Data Stream module, set the **FfcControl** feature value to **Disable**.

FFC Wizard Sample Program

AVT provides the source code of a sample application, called `ffc-wizard`, that computes the coefficients and packs them in a binary file targeting the frame grabber.

The purpose of this sample code is threefold:

- guide the user through the calibration procedure;
- provide a technical and practical translation of what's described in this document;
- provide building blocks for developing custom applications.

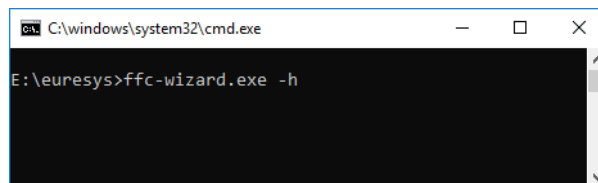
Building

The source code lies in a single source file: `src/ffc-wizard.cpp`. Building the application should be straightforward;

- for Windows, there is a Microsoft Visual Studio project file `ffc-wizard.vcproj`;
- for Linux and macOS, a Makefile is provided.

Usage

The wizard is a console application. The help message is displayed when the flag `-h` (or `--help`) is given:



getting the help message

```
> ffc-wizard.exe --help
Flat Field Correction Wizard
ffc-wizard [OPTIONS]

Options:
  --if=INT           Index of GenTL interface to use
  --dev=INT          Index of GenTL device to use
  --ds=INT           Index of GenTL data stream to use
  --average=INT      Number of images to average (default: 10)
  --roi_x=INT         Horizontal offset in pixels of ROI upper-left corner (default: 0)
  --roi_y=INT         Vertical offset in pixels of ROI upper-left corner (default: 0;
ignored for line-scan)
  --roi_width=INT     Width of ROI (default: whole image)
  --roi_height=INT    Height of ROI (default: whole image; ignored for line-scan)
  --balance           Compute flat image average on all components rather than on each
component
  --linescan          Force line-scan mode i.e. average image lines (automatically
enabled for line-scan cards)
  --dark-setup=SCRIPT Path to setup script for dark acquisitions
  --flat-setup=SCRIPT Path to setup script for flat acquisitions
  --timeout=MS        Maximum time in milliseconds to wait for an image (default: 1000)
```

```
--dark-histogram=FILE    Path to histogram html page of average dark image to output and
open
--flat-histogram=FILE    Path to histogram html page of average flat image to output and
open
--output-ffc=FILE        Path to coefficients output file (Coaxlink ffc binary format)
--load-ffc=FILE          Load coefficients into Coaxlink coefficients partition (default:
computed coefficients)
--no-interact            Skip user interaction
-h --help                Display help message
```

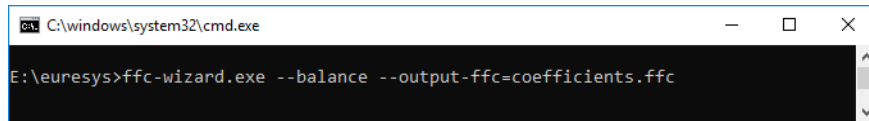
Note: the ROI options allow defining a rectangular region to consider while computing averages,
this is useful to eliminate pixels close to borders in images subject to vignetting.

Options details

Flags	Details
--if, --def, --ds	the indexes of the GenTL modules that identify the data stream to use (and/or to configure)
--average	the number of images to acquire for dark and flat acquisitions; only the average of those acquisitions is further used in the calibration procedure
--roi_x, --roi_y, --roi_width, --roi_height	an optional rectangular region to consider when computing the average pixel value of the flat image (<i>average(Flat)</i>); this impacts the evaluation of the gain value for each pixel
--balance	enables the white balance; i.e. whether the coefficients of color pixel components are computed to balance the component values or not; obviously, this requires the flat background used to acquire the flat image(s) to be as close as possible to a true gray (for which all RGB components would have identical values)
--dark-setup, --flat-setup	optional configuration scripts to run before dark and flat acquisitions
--dark-histogram, --flat-histogram	optional; path to output file showing the histogram of dark and flat image pixel components; this gives a visual overview of the dark current variations as well as the variations in the flat image
--no-interact	normally the wizard waits for the user to setup the system before starting the dark and flat acquisitions; when this flag is used, the wizard does not wait for the user (nor does it open the created histogram html pages)

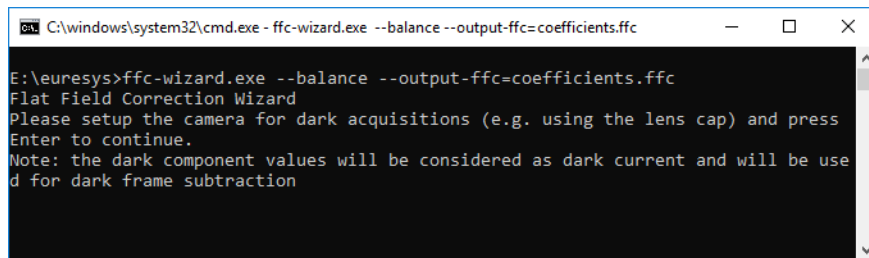
Example

Here is an illustrated example that generates FFC coefficients (written to the file `coefficients.ffc`) using the white balance functionality. The command to run from a console window is the following:



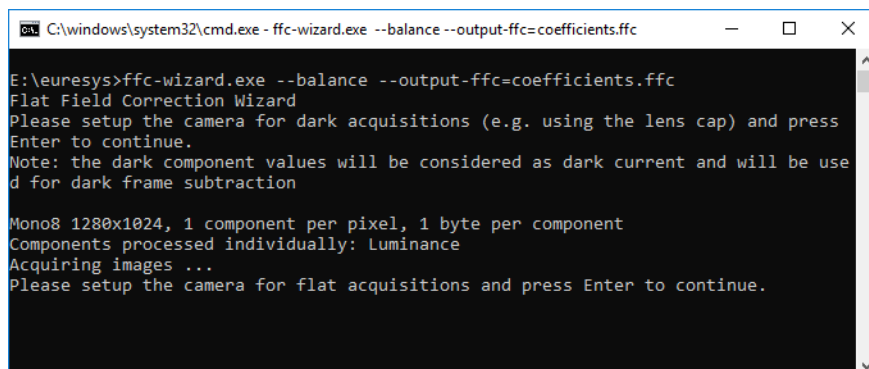
```
C:\windows\system32\cmd.exe
E:\euresys>ffc-wizard.exe --balance --output-ffc=coefficients.ffc
```

The program starts and tells you what to do:



```
C:\windows\system32\cmd.exe - ffc-wizard.exe --balance --output-ffc=coefficients.ffc
E:\euresys>ffc-wizard.exe --balance --output-ffc=coefficients.ffc
Flat Field Correction Wizard
Please setup the camera for dark acquisitions (e.g. using the lens cap) and press
Enter to continue.
Note: the dark component values will be considered as dark current and will be use
d for dark frame subtraction
```

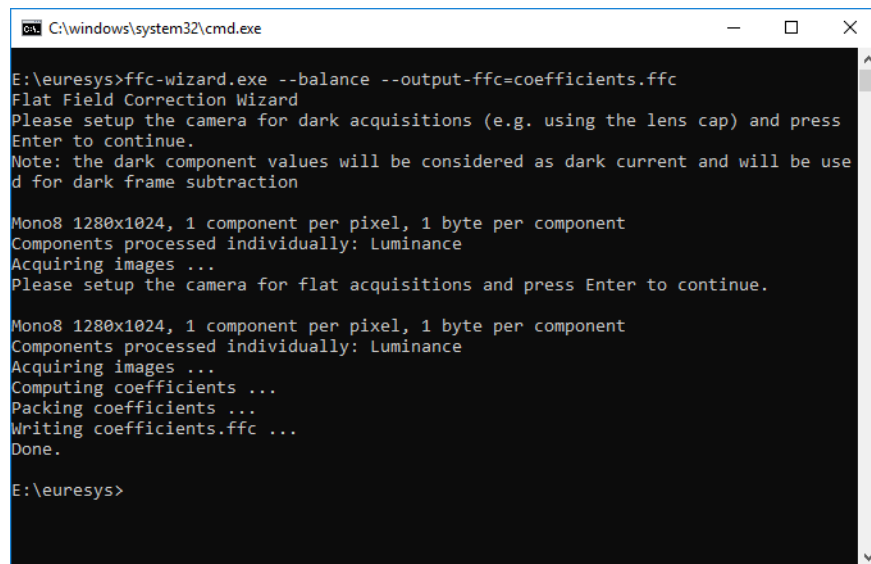
You can prepare your setup for the dark acquisitions and press Enter when you are ready. It will then acquire the series of dark images and display the instructions for the next step



```
C:\windows\system32\cmd.exe - ffc-wizard.exe --balance --output-ffc=coefficients.ffc
E:\euresys>ffc-wizard.exe --balance --output-ffc=coefficients.ffc
Flat Field Correction Wizard
Please setup the camera for dark acquisitions (e.g. using the lens cap) and press
Enter to continue.
Note: the dark component values will be considered as dark current and will be use
d for dark frame subtraction

Mono8 1280x1024, 1 component per pixel, 1 byte per component
Components processed individually: Luminance
Acquiring images ...
Please setup the camera for flat acquisitions and press Enter to continue.
```

You can prepare your setup for the flat acquisitions and press Enter when you are ready. It will then acquire the series of flat images, compute the corresponding coefficients and write them to the file coefficients.ffc



```
C:\windows\system32\cmd.exe

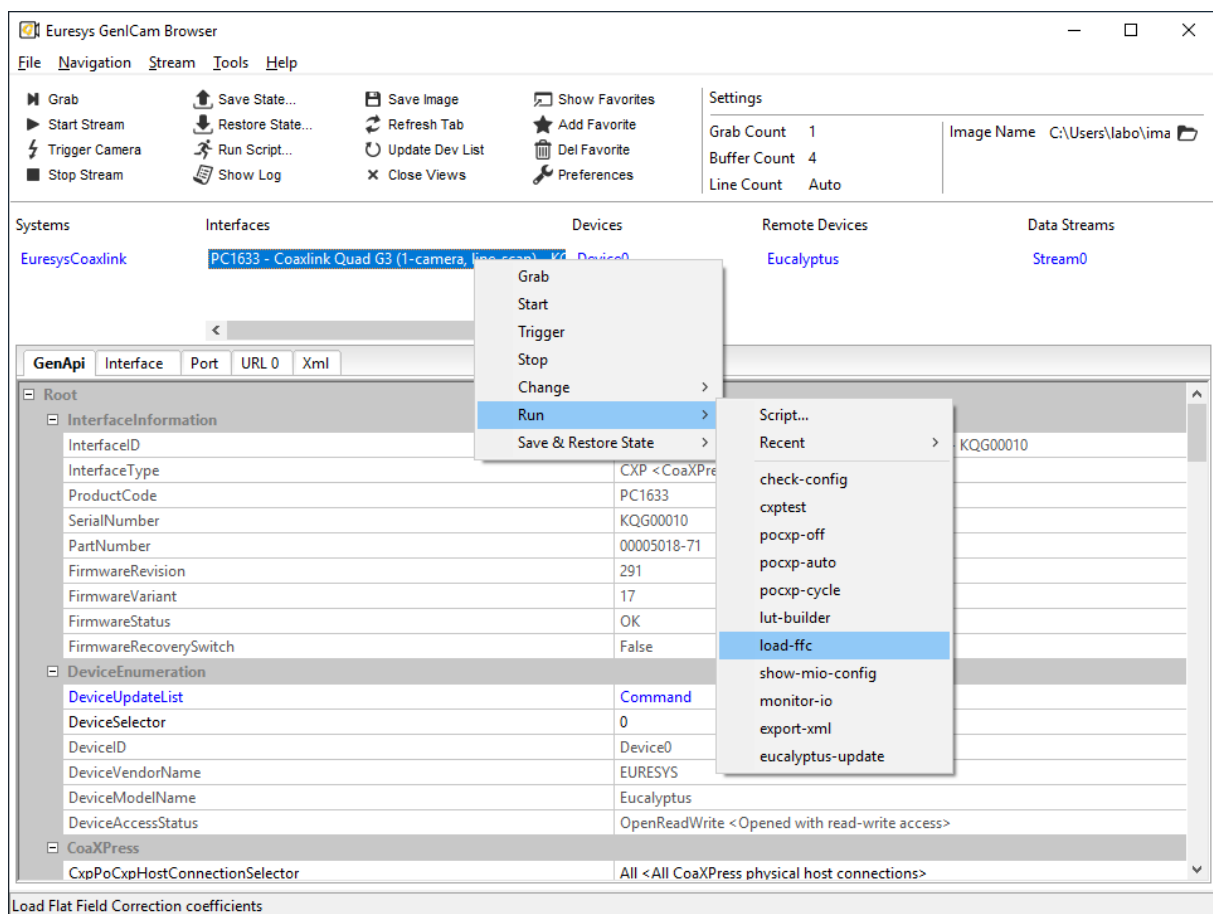
E:\euresys>ffc-wizard.exe --balance --output-ffc=coefficients.ffc
Flat Field Correction Wizard
Please setup the camera for dark acquisitions (e.g. using the lens cap) and press
Enter to continue.
Note: the dark component values will be considered as dark current and will be use
d for dark frame subtraction

Mono8 1280x1024, 1 component per pixel, 1 byte per component
Components processed individually: Luminance
Acquiring images ...
Please setup the camera for flat acquisitions and press Enter to continue.

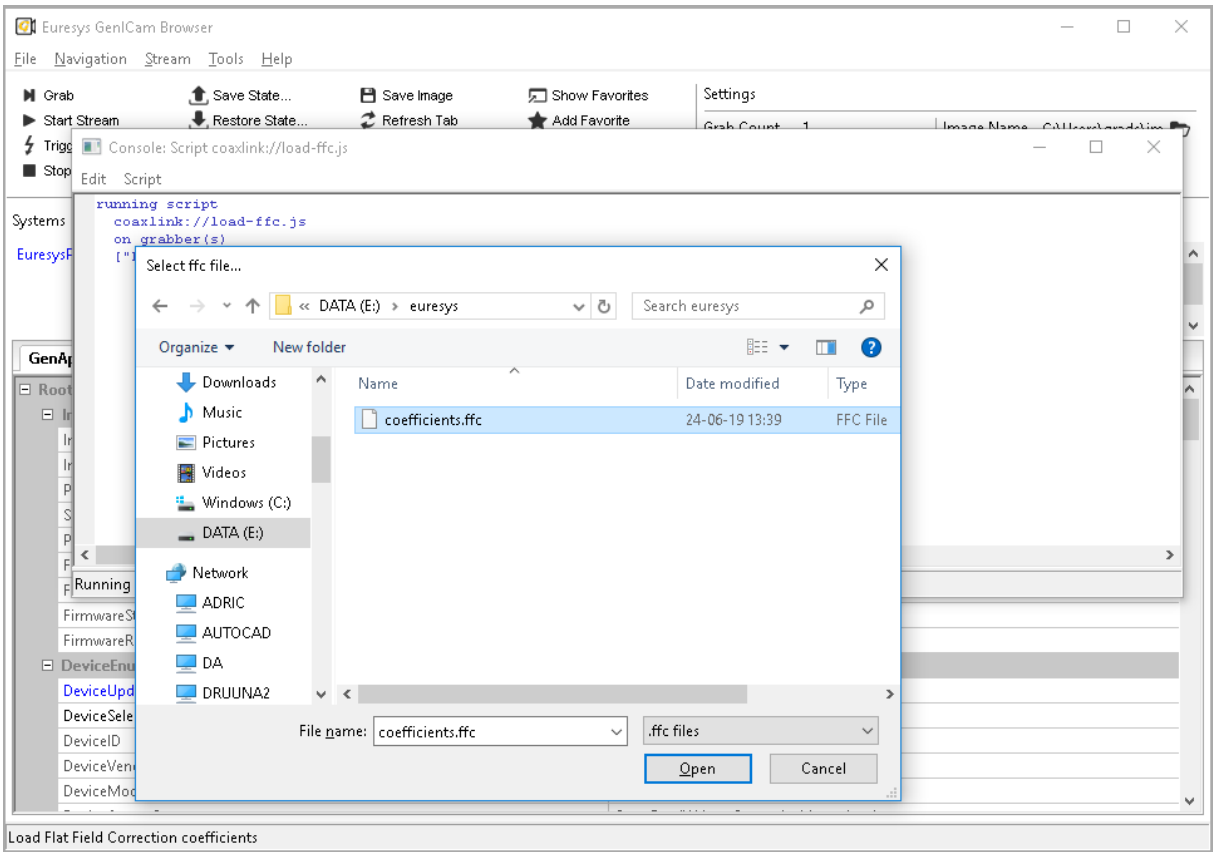
Mono8 1280x1024, 1 component per pixel, 1 byte per component
Components processed individually: Luminance
Acquiring images ...
Computing coefficients ...
Packing coefficients ...
Writing coefficients.ffc ...
Done.

E:\euresys>
```

Later on you can load the coefficients from the **GenICam Browser (Deprecated)** for example, by running the load-ffc script as follows:



Then you can select the previously created file coefficients.ffc



From that moment, the coefficients are loaded into the frame grabber memory and the FFC processing is enabled.

Design

The calibration procedure as well as the packing of the coefficients is controlled by the main function ffcWizard.

ffcWizard tasks

Task	Done by function
acquiring a series of dark images to build an average dark image	acquireImages
acquiring a series of flat images to build an average flat image	acquireImages
computing the gain values for each pixel component	computeGain
using the dark pixel component values as offset values	computeOffset
packing offset and gain values into 16-bit little-endian values	packCoefficients
writing the packed coefficients into a binary file	savePackedCoefficients

Customization

The sample application already supports a few common pixel formats: Mono, RGB, RGBa and Bayer.

Limitation: to limit the complexity of the sample application, we consider (for pixel formats with several components per pixel) that all components have the same size. Supporting pixel formats with different component sizes is still possible by updating the functions `addImage` and `addComponents`.

To support a new pixel format (under the condition of the previous limitation), we need to modify two functions:


1. `Image::getComponentsPerPixel`, to return the number of components per pixel for the new format identified by its **PFNC** name
2. `Image::getComponentFilters`, to return a `std::vector` of `ComponentFilter` objects describing how the pixel components of the new format (identified by its PFNC name) are positioned in the image


The `ComponentFilter` objects are used to separate the processing of the different pixel components while evaluating the Gain and Offset values. For example, in RGB format, the FFC coefficients related to the Red components are computed using the Red components from the dark and flat images.


Please see the source code of `Image::getComponentFilters` for details about pixel component layout configuration.


7.5. Lookup Table Processing


Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo**, **PC3621-LH-T Coaxlink Mono CXP-12-LH**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 **Quad G3** (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan)

 **Octo** (1-camera), (1-camera, custom-logic), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (2-camera, line-scan, custom-logic), (3-camera), (4-camera), (4-camera, line-scan), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera), (8-camera, line-scan)

 **Mono 12 LH** (1-camera), (1-camera, line-scan)

 **Duo 12** (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan)

 **Value 12** (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan)

Introduction to LUT Processing	134
Monochrome Lookup Table Processing	135
LUT Content Definition	136
LUT Setup Procedure	143

Introduction to LUT Processing

eGrabber-driven frame grabbers provide lookup table processing for monochrome pixel formats exclusively!

See also: "Monochrome Lookup Table Processing" on page 135 for a detailed description.

The **eGrabber** driver provides four methods to define the content of lookup tables.

See also: "LUT Content Definition" on page 136

See also: "LUT Setup Procedure" on page 143 to setup lookup tables.

Monochrome Lookup Table Processing

Configurations

The following table lists all the available lookup table configurations for monochrome pixels:

Configuration	Input Pixel Format [PFNC]	Input bits	Output bits
M_8x8	Mono8, Raw	8	8
M_10x8	Mono10	10	8
M_10x10			10
M_10x16			16
M_12x8	Mono12	12	8
M_12x12			12
M_12x16			16



NOTE

Monochrome 8-bit pixels can be transformed into monochrome 8-bit pixels, monochrome 10-bit pixels can be transformed into monochrome 8-bit, 10-bit or 16-bit pixels and, monochrome 12-bit pixels can be transformed into monochrome 8-bit, 12-bit or 16-bit pixels.

Lookup Table Data Sets

A *lookup table data set* is defined as the set of data required to configure one lookup table for each component of a pixel. In the case of monochrome pixels, a lookup table data set includes only one single lookup table content.

The number of lookup table data sets that can be uploaded depends on the lookup table configuration:

Configuration	Data Sets
M_8x8	16
M_10x8, M_10x10, M_10x16	4
M_12x8, M_12x12, M_12x16	1

LUT Content Definition

Methods

The **eGrabber** driver provides four methods to define the content of a lookup table.

Response Control

The *Response Control* method defines the transfer function of a lookup table by means of four parameters: "**Brightness**" on page 137, "**Contrast**" on page 138, "**Visibility**" on page 139 and "**Negative**" on page 140.

The **Brightness** and **Contrast** parameters provide controls similar to the brightness and contrast controls of a television monitor.

The **Visibility** parameter provides control to smoothly reshape the transfer function to cover the full input range.

The **Negative** parameter allows transforming an image into its negative image.

Emphasis

The *Emphasis* method defines the transfer function of a lookup table by means of two parameters: "**Emphasis**" on page 141 and "**Negative**" on page 140.

It allows transforming an image using a power-law expression also known as γ – Gamma – function.

The **Negative** parameter allows transforming an image into its negative image.

Threshold

The *Threshold* method defines a double threshold transformation law by means of five parameters: "**SlicingLevel**, **SlicingBand**, **LightResponse**, **BandResponse** and **DarkResponse**" on page 142.

Table

The *Table* method defines the transfer function of a lookup table in a tabular form.

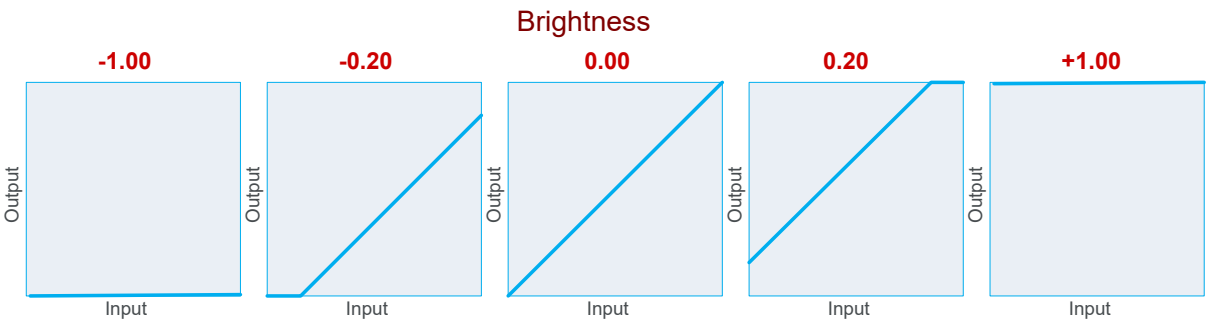
Parameters

Brightness

The **Brightness** parameter exclusively applies to the "Response Control" on page 136 lookup table definition method.

It implements a control similar to the brightness control of a television monitor.

Brightness	Note
-1.00	Minimum value. Darkest output. The whole input range data gets transformed into the full black. This rule applies for any chosen Contrast value.
0.00	Default value. Any increase in the brightness towards +1.00 results into a lighter output. Any decrease of the brightness towards -1.00 results into a darker output.
+1.00	Maximum value. Lightest output. The whole input range data gets transformed into the full white. This rule applies for any chosen Contrast value.



Effect of **Brightness** when all other controls are set to their default value:
Contrast = 1.00; Visibility = 0.00; Negative = FALSE.

Contrast

The Contrast parameter exclusively applies to the "Response Control" on page 136 lookup table definition method.

It implements a control similar to the contrast control of a television monitor.

The slope of the transformation law is the gain, which is non-linearly controlled from the Contrast parameter.

Mathematically, the relationship is:

$$\text{Gain} = 10^{2 \times (\text{Contrast} - 1)}$$

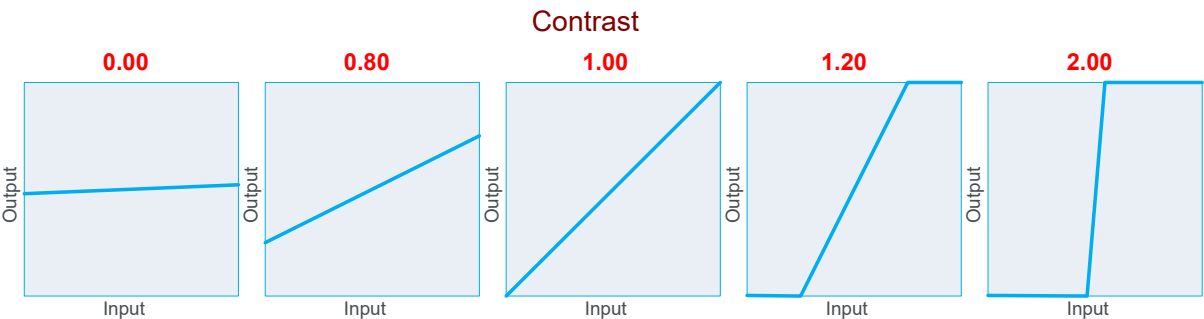
Contrast	Gain	Note
0.00	0.01	Min. Contrast value; smallest gain
1.00	1	Default Contrast value;unity gain
2.00	100	Max. Contrast value; largest gain

To achieve a required given gain, the contrast control should be set to:

$$\text{Contrast} = 1 + (\log_{10} \text{Gain})/2$$

If the required gain is expressed in decibels (dB):

$$\text{Contrast} = 1 + \text{Gain(dB)}/40$$



Effect of Contrast when all other controls are set to their default value:
Brightness = 0.00; Visibility = 0.00; Negative = FALSE

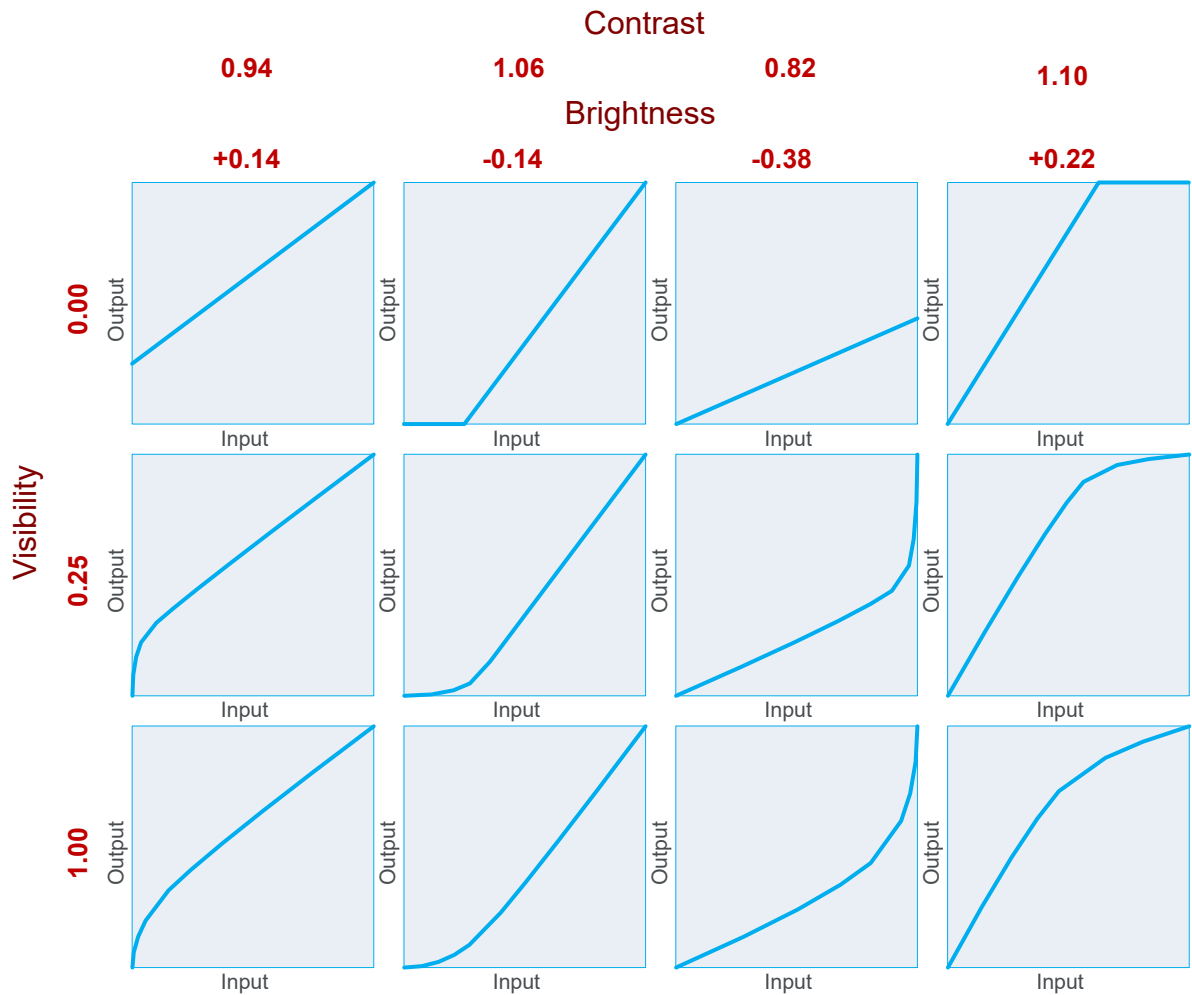
Visibility

The **Visibility** parameter exclusively applies to the "Response Control" on page 136 lookup table definition method.

The operation of **Contrast** and **Brightness** parameters occasionally removes some part of the input dynamics. Very dark regions of the image can be transformed into full black, and become invisible. This holds true for very bright regions, clipping to full white.

The **Visibility** parameter has been created to smoothly reveal these hidden parts in the image.

Visibility	Gain	Note
0.00	0.01	Minimum and default value. This generates the piecewise linear transformation curves. Choosing values closer to +1 generates smoother curves.
1.00	100	Maximum value.



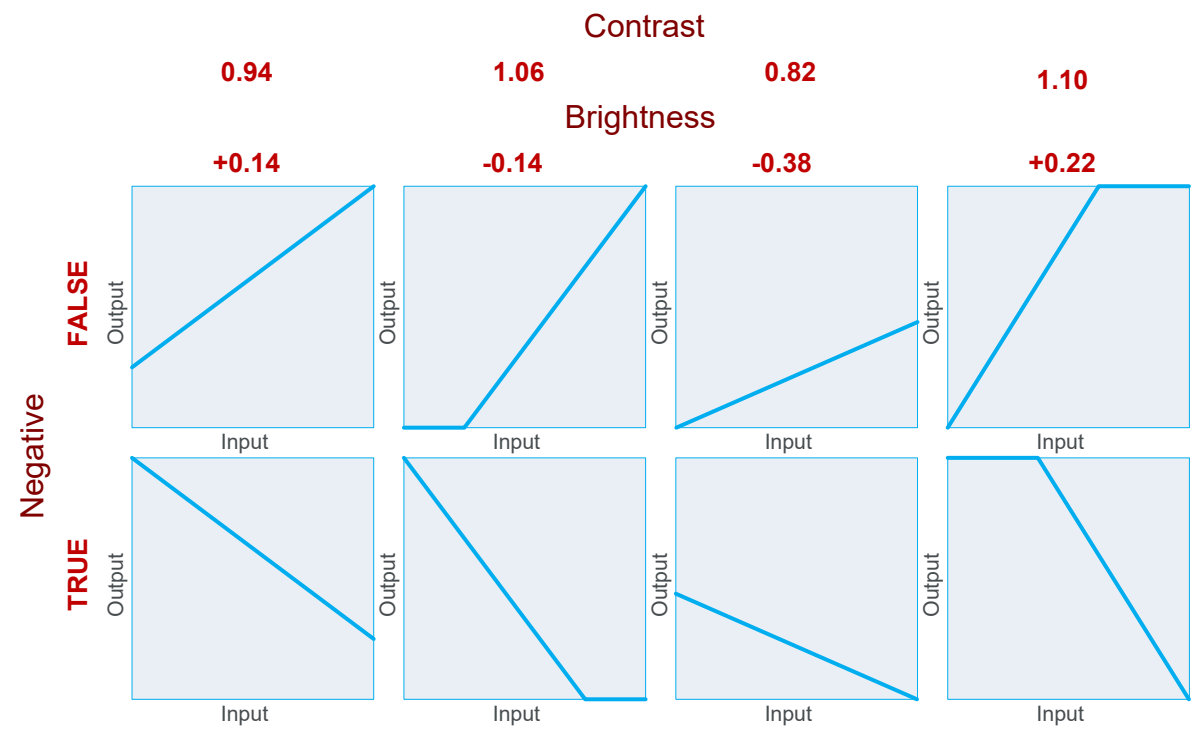
Effect of **Visibility** for typical values of **Contrast** and **Brightness** parameters assuming that Negative = FALSE

Negative

The **Negative** parameter applies to both the "Response Control" on page 136 and the "Emphasis" on page 136 lookup table definition methods.

This control allows transforming an image into its negative image, where the lightest areas of the image appear darkest and the darkest areas appear lightest.

Negative	Note
FALSE	Default value.
TRUE	The transformation table is mirrored around a vertical axis in the graphs. This swaps the black and white values, and gives rise to a photographic negative effect.



Effect of **Negative** for typical values of other controls

Emphasis

The **Emphasis** parameter exclusively applies to the "**Emphasis**" on page 136 lookup table definition method.

It allows transforming an image using a power-law expression:

$$\text{Output} = \text{Input}^\gamma$$

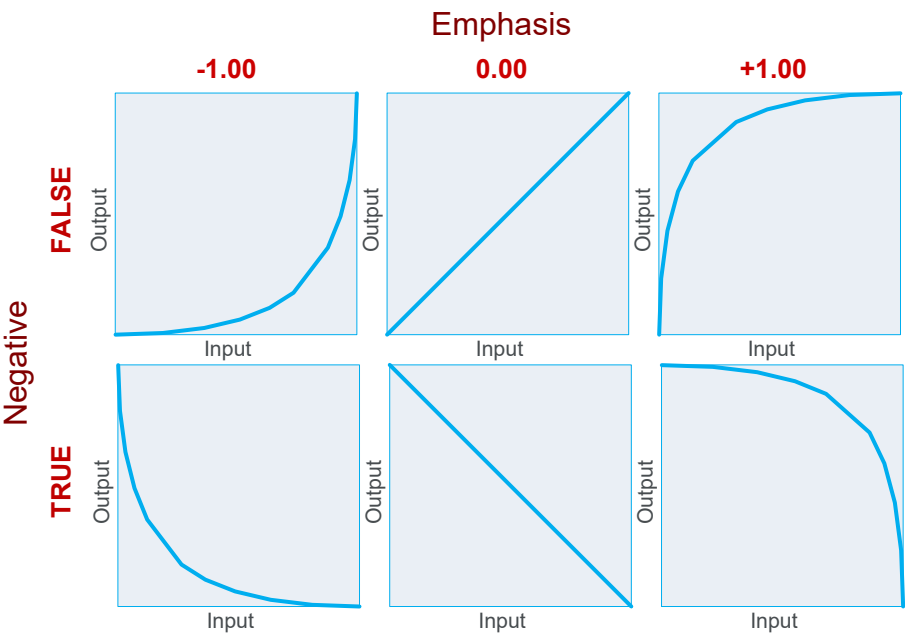
The γ – Gamma – exponent is mathematically linked to Emphasis by:

$$\gamma = 10^{-\text{Emphasis}}$$

Emphasis	Gamma	Note
1.00	0.1	Max. Emphasis value; smallest γ value
0.00	1	Default Emphasis value; linear law
-1.00	10	Min. Emphasis value; largest γ value

To achieve a required given γ , **Emphasis** should be set to:

$$\text{Emphasis} = -\log_{10}\gamma$$



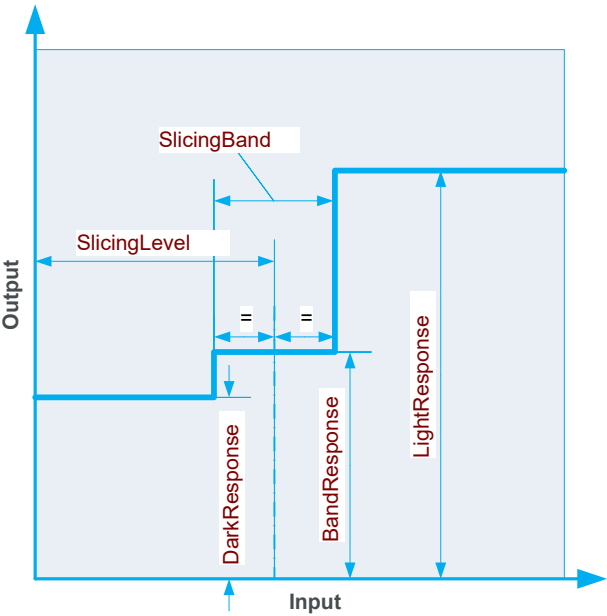
Emphasis effect for typical values of **Emphasis** and both values of **Negative**

SlicingLevel, SlicingBand, LightResponse, BandResponse and DarkResponse

SlicingLevel, SlicingBand, LightResponse, BandResponse and DarkResponse parameters exclusively apply to the "Threshold" on page 136 lookup table definition method.

As shown on the next figure, the parameters set defines a double threshold transformation law.

Parameter	Minimum Value	Default Value	Maximum Value
SlicingLevel	0.00	0.50	1.00
SlicingBand	0.00	0.50	1.00
LightResponse	0.00	0.75	1.00
BandResponse	0.00	0.50	1.00
DarkResponse	0.00	0.25	1.00



Double threshold transfer function



NOTE

SlicingLevel specifies the mean value of both thresholds in the input range.

LUT Setup Procedure

To setup the lookup table processing, proceed as follows:

1. Disable the lookup table
2. Define the lookup table configuration
3. Define the content of the lookup table
4. Upload the lookup table content into a specified lookup table data set
5. Enable the lookup table with a specified data set

Disabling the lookup table

To disable the lookup table:

- Set the **LUTEnable** feature to a **Off**.

Defining the lookup table configuration

To define the lookup table configuration, set the **LUTConfiguration** feature according to:

- The camera pixel type and bit depth
- The required output bit depth.

See also: "Monochrome Lookup Table Processing" on page 135 for configurations applicable to monochrome pixels.



NOTE

The lookup table configuration must be set prior to any other action.

Defining the lookup table content

See also: "LUT Content Definition" on page 136 for a description of the parametric and tabular methods used for defining a lookup table content.



NOTE

At least one lookup table set must be defined.

Upload a lookup table content

To upload a lookup table content in one operation:

- Select a lookup table data set to access by assigning the appropriate value to the **LUTSet** feature. For instance **Set1**.
- Set the **LUTIndex** feature to **0**.
- Write a string of **LUTLength** values to the **LUTValue** feature.



NOTE

The application may also selectively upload any individual lookup table entry or any block of consecutive lookup table entries.

Reading back a lookup table data set

To read back the lookup table data set in one operation:

- Select a lookup table data set to access by assigning the appropriate value to the LUTSet feature. For instance Set1.
- Set the LUTIndex feature to 0.
- Set the LUTReadBlockLength feature to the value returned by LUTLength.
- Get a string of LUTReadBlockLength values from the LUTValue feature.



NOTE

The application may also selectively read any lookup table entry individually or any block of consecutive entries.

Enabling the lookup table

To enable the lookup table:

- Set the LUTEnable feature to a value designating the lookup table data set to use.

Configuration Script Example

The following script is an example illustrating how to configure the lookup table for monochrome 8-bit to 8-bit operation and to define and upload 4 lookup table data sets using different lookup table definition methods.

```
function configure(g) {
  // Disable the lookup table
  g.StreamPort.set('LUTEnable', 'Off');
  // Configure the lookup table
  g.StreamPort.set('LUTConfiguration', 'M_8x8');

  // Build lookup table data set 1: response control
  g.StreamPort.set('LUTSet', 'Set1');
  require('coaxlink://lut/response-control')(g, { Contrast: 0.94
    , Brightness: 0.14
    , Visibility: 0.25
    , Negative: false });


  // Build lookup table data set 2: emphasis
  g.StreamPort.set('LUTSet', 'Set2');
  require('coaxlink://lut/emphasis')(g, { Emphasis: 0.5
    , Negative: true });


  // Build lookup table data set 3: threshold
  g.StreamPort.set('LUTSet', 'Set3');
  require('coaxlink://lut/threshold')(g, { SlicingLevel: 0.5
    , SlicingBand: 0.5
    , LightResponse: 0.75
    , BandResponse: 0.5
    , DarkResponse: 0.25 });

  // Build lookup table data set 4: table
  g.StreamPort.set('LUTSet', 'Set4');
  var i;
  for (i = 0; i < 256; ++i) {
    g.StreamPort.set('LUTIndex', i);
    g.StreamPort.set('LUTValue', String(255 - i));
  }
}
configure(grabbers[0]);
```

7.6. Bayer CFA Decoding

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12**
Value:

 (1-camera), (2-camera), (2-camera, bayer)

 (1-camera), (2-camera)

 (1-camera)

 (1-camera)

Bayer CFA Decoding Methods	147
Requirements and Performances	149
Using Bayer CFA Decoder	151
Advanced and Legacy Methods	153


Bayer CFA Decoding Methods

Various Bayer CFA decoding methods are defined to transform the raw Bayer CFA data stream issued by the camera into an RGB color data stream

CFA decoding method 1

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera)

 (1-camera), (2-camera)

 (1-camera)


The CFA decoding method 1 also known as "Legacy method" uses a 3 x 3 kernel and a linear interpolation method to compute the missing color components.

See also: "Advanced and Legacy Methods" on page 153 for an extensive description

CFA decoding method 2

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera)

 (1-camera), (2-camera)


 (1-camera)

The CFA decoding method 2 also known as "Advanced method" uses a 3 x 3 kernel and an advanced interpolation method to compute the missing color components.

See also: "Advanced and Legacy Methods" on page 153 for an extensive description

CFA decoding method 3

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3622-T Coaxlink Duo CXP-12**:

 (2-camera), (2-camera, bayer)

 (1-camera)

 (1-camera)

The CFA decoding method 3 uses a 5 x 5 kernel and a gradient-based interpolation method to compute the missing color components.



WARNING


Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**:

 (2-camera)

The Method 3 decoder is only available for Device0.

CFA decoding method 5

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3** and **PC3602-T Coaxlink Octo**:

 (2-camera), (2-camera, bayer)

 (2-camera)

The CFA decoding method 5 uses a 2x2 average-based interpolation method to compute the missing color components.



WARNING

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**:

 (2-camera)

The Method 5 decoder is only available for Device0.


Requirements and Performances

Scanning geometry

Bayer CFA decoding is only available for 1X_1Y progressive-scan geometry.

Maximum line length

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3** and **PC3622-T Coaxlink Duo CXP-12**:

 (2-camera), (2-camera, bayer)

 (1-camera)

The maximum line length is 8,192 pixels.

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera)

 (1-camera)

 (1-camera)

The maximum line length is 16,384 pixels.


Applies to the following firmware variants of **PC3602-T Coaxlink Octo**:

 (2-camera)

The maximum line length is 32,768 pixels.

Peak pixel processing rate

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**:

 (2-camera), (2-camera, bayer)

The peak pixel processing rate (per stream) is 1,000,000,000 pixels/second.

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3622-T Coaxlink Duo CXP-12**:

 (1-camera)

 (2-camera)

 (1-camera)

The peak pixel processing rate (per stream) is 1,108,000,000 pixels/second.

Applies to the following firmware variants of **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera)

 (1-camera)

The peak pixel processing rate (per stream) is 2,216,000,000 pixels/second.

PCIe performance

PCI Express Interface	Sustainable PCIe data rate	RGB8	RGB10, RGB12, RGB14, RGB16
"4-lane Rev 3.0 PCIe end-point" on page 334	3,350 MB/s typical	~1,117 Mpixels/s	~558 MPixels/s
"8-lane Rev 3.0 PCIe end-point" on page 335	6,700 MB/s typical	~2,238 Mpixels/s	~1,117 Mpixels/s



NOTE

When configured to deliver RGB8 pixels, the "PCI Express Interfaces" on page 333 is capable to sustain the highest CFA decoder pixel rate!

For 10-, 12-, 14- and 16-bit bit depths, the sustainable data output rate is further limited by the performances of the "PCI Express Interfaces" on page 333 on the Host PC.

Latency

The hardware CFA decoder performs on-the-fly conversion with a negligible latency when the data throughput is NOT limited by the available PCI Express bandwidth!

Using Bayer CFA Decoder

Using the Bayer CFA decoder of *Coaxlink frame grabbers*

Prerequisites

1. *Frame grabber*
Coaxlink frame grabber and firmware-variant with CFA decoder
2. *Camera*
 - a. Bayer CFA area-scan
 - b. Less than "Max. line length" pixels per line
 - c. Having the color registration (GR, RG, GB, or BG) of the first two first transmitted pixels of the first transmitted line and the pixel bit depth (8-bit, 10-bit, 12-bit, 14-bit or 16-bit) correctly specified in the PixelF field of the CoaXPress image header.



WARNING

When the fields Xoffs and/or Yoffs are greater than 0, the camera must report an adapted PixelF value corresponding to the transmitted data!

Bayer to RGB Pixel Processing Configurations

When the Bayer CFA decoder is enabled:

- the "[Pixel Unpacking and Alignment](#)" on [page 118](#) control is inoperative, the frame grabber unpacks 10-bit, 12-bit or 14-bit pixels to lsb.
- the "[Pixel Components Swapping](#)" on [page 163](#) feature allows to swap the Red and Blue components and deliver either BGR or RGB pixels.

Input Pixel Format	FFC	RedBlueSwap	Output Pixel Format
Bayer**8	off on	off	RGB8
		on	BGR8
Bayer**10pmsb	off on	off	RGB10
		on	BGR10
Bayer**12pmsb	off on	off	RGB12
		on	BGR12
Bayer**14pmsb	off on	off	RGB14
		on	BGR14
Bayer16	off on	off	RGB16
		on	BGR16

Enabling the Bayer CFA Decoder

In the Data Stream module, set the **BayerMethod** feature value to **Legacy**, **Advanced**, **Method3** or **Method5** according to the desired interpolation method.

Disabling the Bayer CFA Decoder

In the Data Stream module, set the **BayerMethod** feature value to **Disable**.

Advanced and Legacy Methods

This topic describes two Bayer CFA decoding methods respectively named **Legacy** and **Advanced**.

The two methods transform the raw Bayer CFA data stream issued by the camera into an RGB color data stream using a 3x3 kernel. The missing pixel components are reconstructed from the nearest components.

The **Legacy** interpolation method computes the missing color components by applying exclusively the Mean() function.

The **Advanced** interpolation method computes the missing color components using the Mean() and the Median() functions. It eliminates the aliasing effect on the highly contrasted sharp transitions in the image.

Functions Definitions

The min() function returns the lowest integer value from a set of 2 integer values.

The max() function returns the highest integer value from a set of 2 integer values.

The mean() function returns one integer value that represents the mean value of 2 integers. It is computed as follows:

Function	Mean(a,b)
<i>Formula</i>	$(a + b + 1) >> 1$

The median() function returns a set of two integer values that are the two median values of a set of four integers. It is computed as follows:

Function	Median(a,b,c,d)
<i>Value 1 formula</i>	$\text{Min} [\text{Max}(a,b); \text{Max}(c,d)]$
<i>Value 2 formula</i>	$\text{Max} [\text{Min}(a,b); \text{Min}(c,d)]$

Decoder operation

For each pixel of the source image, the CFA decoder computes two missing color components from surrounding pixels.

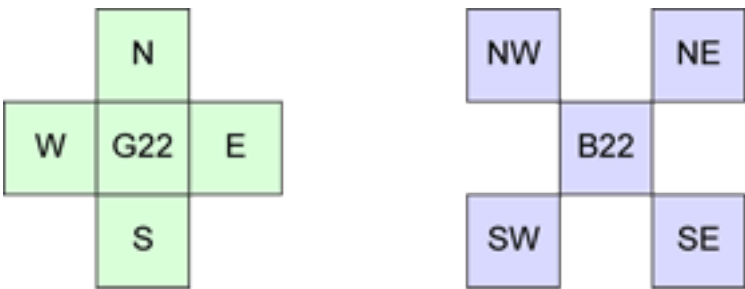
The text hereafter describes how the 4 central pixels located at positions 22, 32, 23 and 33 of a 4 x 4 Bayer CFA array are computed:

B11	G21	B31	G41
G12	R22	G32	R42
B13	G23	B33	G43
G14	R24	G34	R44

The relative positions of the surrounding pixels are identified by compass markings:

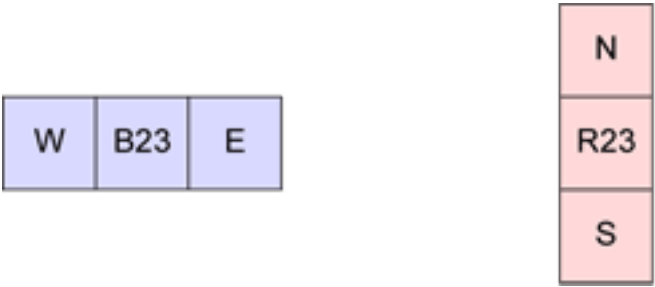
NW	N	NE
W		E
SW	S	SE

Formulas for position 22



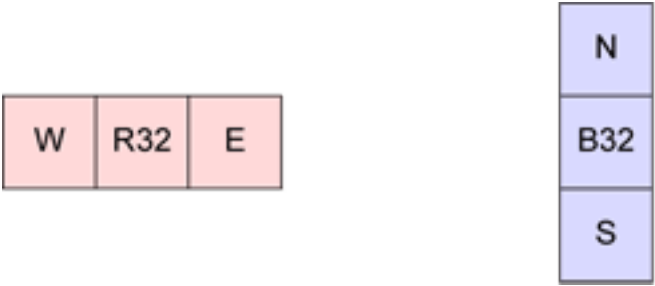
Component	Method	Formula
R22	Legacy, Advanced	R22
G22	Legacy	Mean[Mean(N,S), Mean(W,E)]
	Advanced	Mean[Median(N, S, E, W)]
B22	Legacy	Mean[Mean(NW, SW), Mean(NE, SE)]
	Advanced	Mean[Median(NW, SW, NE, SE)]

Formulas for position 23



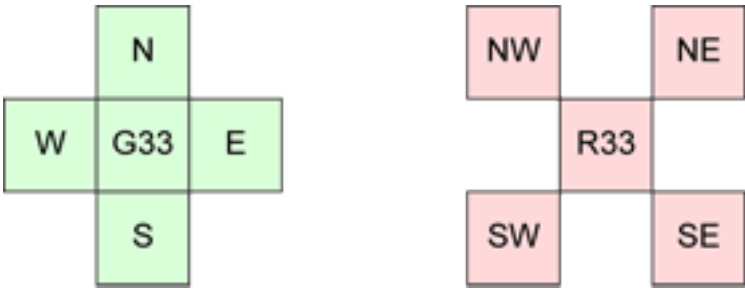
Component	Method	Formula
R23	Legacy, Advanced	Mean(N,S)
G23	Legacy, Advanced	G23
B23	Legacy, Advanced	Mean(W,E)

Formulas for position 32



Component	Method	Formula
R32	Legacy, Advanced	Mean(W,E)
G32	Legacy, Advanced	G32
B32	Legacy, Advanced	Mean(N,S)


Formulas for position 33



Component	Method	Formula
R33	Legacy	Mean[Mean(NW, SW), Mean(NE, SE)]
	Advanced	Mean[Median(NW, SW, NE, SE)]
G33	Legacy	Mean[Mean(N,S), Mean(W,E)]
	Advanced	Mean[Median(N, S, E, W)]
R33	Legacy, Advanced	B33

7.7. Pixel Binning

Applies to the following firmware variants of PC3623-T Coaxlink Quad CXP-12 Value:

 (1-camera), (1-camera, line-scan)

Binning Configurations	158
Specifications	160
Limitations	161

Binning Configurations

The binning processing unit combines a cluster of 2 x 2 or 4 x 4 pixels into a single pixel by summing or averaging their respective pixel values.

2x2 binning window

	Pixel 0	Pixel 1	Pixel 2	Pixel 3	Pixel 4	Pixel 5	Pixel 6	Pixel 7
Line 0	[00]	[01]	[02]	[03]	[04]	[05]	[06]	[07]
Line 1	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]
Line 2	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]
Line 3	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]
Line 4	[40]	[41]	[42]	[43]	[44]	[45]	[46]	[47]
Line 5	[50]	[51]	[52]	[53]	[54]	[55]	[56]	[57]
Line 6	[60]	[61]	[62]	[63]	[64]	[65]	[66]	[67]
Line 7	[70]	[71]	[72]	[73]	[74]	[75]	[76]	[77]

The pixel binning is performed on a cluster of 4 pixels. It divides both the image width and the image height by 2.

4x4 binning window

	Pixel 0	Pixel 1	Pixel 2	Pixel 3	Pixel 4	Pixel 5	Pixel 6	Pixel 7
Line 0	[00]	[01]	[02]	[03]	[04]	[05]	[06]	[07]
Line 1	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]
Line 2	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]
Line 3	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]
Line 4	[40]	[41]	[42]	[43]	[44]	[45]	[46]	[47]
Line 5	[50]	[51]	[52]	[53]	[54]	[55]	[56]	[57]
Line 6	[60]	[61]	[62]	[63]	[64]	[65]	[66]	[67]
Line 7	[70]	[71]	[72]	[73]	[74]	[75]	[76]	[77]

The pixel binning is performed on a cluster of 16 pixels. It divides both the image width and the image height by 4.

Binning methods

Bypass

The binning processor is bypassed.

Sum

The binning processor sums up the values of the pixels located in the specified *binning window*.

Mean

The binning processor computes the mean value of the pixels located in the specified *binning window*.



NOTE

The mean operation divides the summation result by 4 or by 16 depending on the selected *binning window*. The result is rounded down to the nearest unsigned integer.

Enabling /disabling binning

Binning is controlled through the **BinningMethod** feature of the Data Stream module:

Binning Method	Action on set
Disable	Disable binning. (Default settings)
Sum_2x2	Enable 2x2 binning using 'sum' method
Mean_2x2	Enable 2x2 binning using 'mean' method
Sum_4x4	Enable 4x4 binning using 'sum' method
Mean_4x4	Enable 4x4 binning using 'mean' method

Specifications

Supported camera pixel types

The binning processing unit can be operated with the following camera pixel types:

- 8-/10-/12-/14- and 16-bit single-component pixels (e.g. Monochrome cameras)
- 8-/10-/12-/14- and 16-bit 3-component pixels (e.g. color RGB cameras)
- 8-/10-/12-/14- and 16-bit 4-component pixels (e.g. color RGBa cameras)
- 8-/10-/12-/14- and 16-bit Bayer CFA cameras only when the Bayer CFA decoder is enabled

Supported input bit depth vs. binning configuration

Binning Method	Supported input bit depth
----------------	---------------------------

Disable	
---------	--

Mean_2x2	8-/10-/12-/14- and 16-bit
----------	---------------------------

Mean_4x4	
----------	--

Sum_4x4	8-/10-/12- and 14-bit
---------	-----------------------

Sum_4x4	8-/10- and 12-bit
---------	-------------------

Limitations

Image width

Width Increment Step

Depending on the binning configuration, the image width settings of the camera must be a multiple of the specified *Width Increment Step* value.

Bit Depth	Binning Method	Binning Window	Width Increment Step (pixels)
8-bit	Bypass	2x2	4
8-bit	Bypass	4x4	4
8-bit	Mean	2x2	8
8-bit	Mean	4x4	16
8-bit	Sum	2x2	4
8-bit	Sum	4x4	8
10-, 12-, 14- and 16-bit	Bypass	2x2	2
10-, 12-, 14- and 16-bit	Bypass	4x4	2
10-, 12-, 14- and 16-bit	Mean	2x2	4
10-, 12-, 14- and 16-bit	Mean	4x4	8
10-, 12-, 14- and 16-bit	Sum	2x2	4
10-, 12-, 14- and 16-bit	Sum	4x4	8

Maximum image width

The max. image width is limited by the FIFO inside the Binning processor. It depends on the pixel format:

Pixel Format	Max. Image Width (pixels)
Mono8	65,536
Mono10, Mono12, Mono14, Mono16	32,768
RGB8	21,840
RGB10, RGB12, RGB14, RGB16	10,920
RGBa8	16,384
RGBa10, RGBa12, RGBa14, RGBa16	8,192

Image Height

Height increment step (area-scan acquisition)

The image height must be a multiple of 2 when using Binning Window 2x2 and a multiple of 4 when using Binning Window 4x4.

Height increment step (line-scan acquisition)

The **ScanLength** settings must be a multiple of 2 when using Binning Window 2x2 and a multiple of 4 when using Binning Window 4x4.

Additional limitations

When the Pixel Binning is enabled, the "**Pixel Unpacking and Alignment**" on page 118 control is inoperative, the frame grabber unpacks 10-bit, 12-bit or 14-bit pixels to lsb.

7.8. Pixel Components Swapping

The image data stream pixel processor can be configured to swap the first and the third component data of 3-component pixels.

The swapping is controlled through the **RedBlueSwap** boolean GenApi feature:

- When set to **False** (default settings), the original component order is preserved
- When set to **True**, the first and the third components are swapped.

The function is available for image acquisition from:

- RGB color cameras delivering 3-component pixel data,
- RGBa color cameras delivering 4-component pixel data,
- BAYER CFA color cameras providing that the BAYER CFA decoding is enabled.

7.9. Endianness Conversion

The image data stream pixel processor delivers 16-bit pixel components using the little-endian convention.

The conversion is not performed when **UnpackingMode** is set to **Off**.

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]

7.10. Pixel Ordering

The 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.

8. Image Data Transfer

- 8.1. Buffer Filling Rules167
- 8.2. Image Width Increment Step169
- 8.3. Image Data Padding171
- 8.4. Horizontal Image Flipping173
- 8.5. Vertical Image Flipping174
- 8.6. Image Data Unscrambling175
- 8.7. Transmission Methods of 1X_2YE Images (Coaxlink series)180
- 8.8. Data Stream Statistics182
- 8.9. Data Transfer Rate Test Program184

8.1. Buffer Filling Rules

A DMA engine transfers the processed image data over the PCI Express bus to the allocated GenTL buffers according to rules that are different for line-scan and area-scan image acquisition.

Area-scan firmware variants

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

- The first acquired line data of a frame is, by default, stored at the beginning of a new buffer. When vertical image flipping is enabled by setting **StripeArrangement** to **Geometry_1X_1YE**, the first acquired line data of a frame is stored at the location of the last full line of a new buffer.
- When image transfer to host memory is done, the buffer, possibly partially filled, is made available to the application for processing.
- When the remaining space of a buffer is not sufficient to store a complete frame, the remaining data is handled according to the "**BufferFilledRule settings**" on page 167.

Line-scan firmware variants

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

- The first acquired line data of a scan is, by default, stored at the beginning of a new buffer. When vertical image flipping is enabled by setting **StripeArrangement** to **Geometry_1X_1YE**, the first acquired line data of a scan is stored at the location of the last full line 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 remaining data is handled according to the "**BufferFilledRule settings**" on page 167.
- When the last line data of a scan is acquired, the last buffer, possibly partially filled, is made available to the application for processing.

BufferFilledRule settings

Discard remaining data


When **BufferFilledRule** is set to **DiscardRemainingData**, the remaining data is discarded.




NOTE


- Default settings for area-scan acquisition.
- Only available for selected line-scan firmware variants.


Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo**, **PC3621-LH-T Coaxlink Mono CXP-12-LH**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 **QuadG3** ^{Prelim} (1-camera), (1-camera, 4-data-stream), (2-camera), (2-camera, bayer), (3-camera), (4-camera)

 **Octo** ^{Prelim} (1-camera), (1-camera, custom-logic), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera), (8-camera, line-scan)

 **Mono12LH** ^{Prelim} (1-camera)

 **Duo12** ^{Prelim} (1-camera), (2-camera)

 **Value12** ^{Prelim} (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan)

Continue in a next buffer (Default settings for line-scan acquisition)


When **BufferFilledRule** is set to **ContinueInNextBuffer**, the acquisition continues into a new buffer and the filled buffer is made available to the application for processing. This setting is also available on selected area-scan firmware-variants.





NOTE


- Default settings for line-scan acquisition.
- Only available for selected area-scan firmware variants.


Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo**, **PC3621-LH-T Coaxlink Mono CXP-12-LH**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 **QuadG3** ^{Prelim} (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)

 **Octo** ^{Prelim} (1-camera, line-scan), (2-camera, line-scan), (2-camera, line-scan, custom-logic), (4-camera, line-scan), (8-camera, line-scan)

 **Mono12LH** ^{Prelim} (1-camera, line-scan)

 **Duo12** ^{Prelim} (1-camera, line-scan), (2-camera, line-scan)

 **Value12** ^{Prelim} (1-camera, line-scan), (2-camera, line-scan), (4-camera, line-scan)

8.2. Image Width Increment Step



WARNING

The width of the image delivered by the camera must be a multiple of the *Image Width Increment Step*.

The *Image Width Increment Step* depends on:

- the camera *Pixel Format*
- the *LinePitch Alignment* - LPA - : the smallest number of bytes that the data stream chain can align








The following table shows the *Image Width Increment Step* vs. the camera *Pixel Format* for all possible LPA values:

Pixel Format	Bytes/pixel	Image Width Increment Step [Pixels]			
		LPA=4	LPA=8	LPA=16	LPA=32
Mono8 or Bayer8	1	4	8	16	32
Mono10-/12-/14-/16 or Bayer10-/12-/14-/16	2	2	4	8	16
RGB8	3	4	8	16	32
RGBA8	4	1	2	4	8
RGB10-/12-/14-/16	6	2	4	8	16
RGBA10-/12-/14-/16	8	1	1	2	4



LPA values

The LPA values depends on the selected product/firmware variant and the usage of the pixel processing:




Values of LPA when the FFC, CFA, JPEG and BIN processing elements are not used

LPA	Applicable product/firmware variants
4	<div>  (1-camera), (1-camera, 4-data-stream), (1-camera, line-scan), (2-camera), (2-camera, bayer), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan) </div> <div>  (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera), (8-camera, line-scan) </div> <div>  (1-camera), (1-camera, line-scan) </div> <div>  (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan) </div> <div>  (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan) </div>
8	
16	<div>  (2-camera, line-scan, custom-logic) </div>
32	<div>  (1-camera, custom-logic) </div>

Values of LPA when the FFC corrector is active

Step	Applicable product/firmware variants
16	<div>  (1-camera), (1-camera, line-scan) </div>
32	<div>  (2-camera) </div>


Values of LPA when the CFA decoder is active

LPA	Applicable product/firmware variants
16	<div>  (1-camera) </div> <div>  (1-camera) </div>
32	<div>  (1-camera), (2-camera) </div>

Values of LPA when the JPEG encoder is active

LPA	Applicable product/firmware variants
32	

Values of LPA for CustomLogic firmware variants


LPA	Applicable product/firmware variants
8	
32	 (1-camera, custom-logic)

Values of *Image Width Increment Step* when the BIN element is active

See also: "Limitations" on page 161

8.3. Image Data Padding

The DMA engine provides the capability to organize the data differently in the buffer by adding line padding or stripe padding.

**NOTE**

For driver versions prior to 6.2, the DMA engine was transferring the whole image data as a single 1D entity regardless the 2D structure: the lines of processed image data are concatenated into the destination buffer.

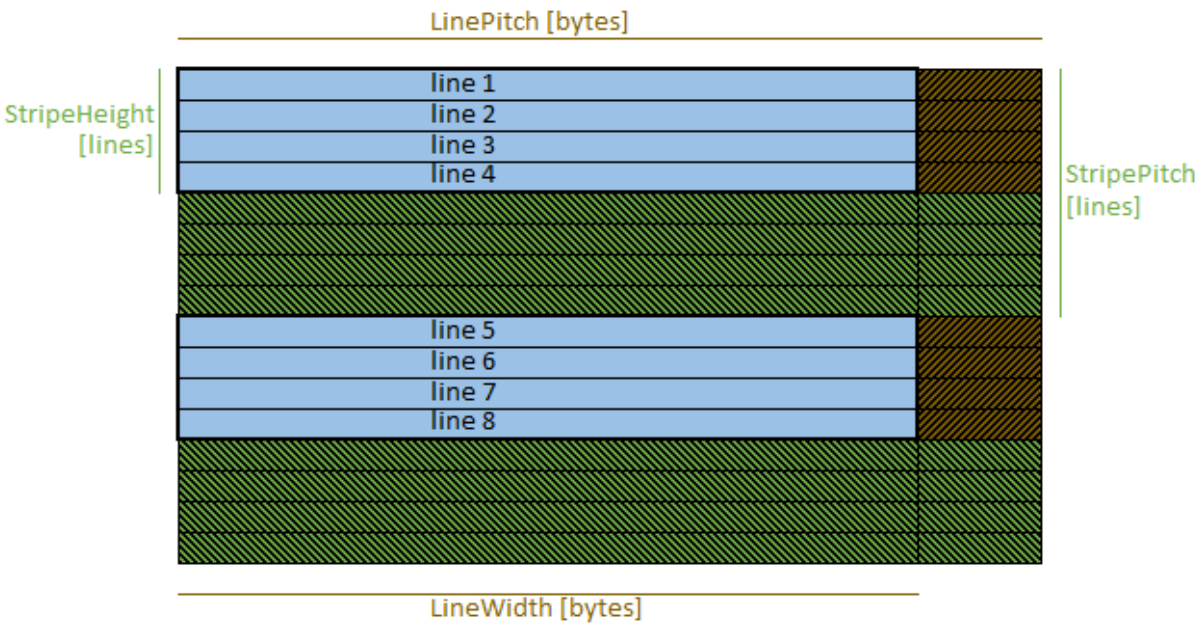


Image buffer padding model

Line Padding

The **LineWidth** and **LinePitch** features control the line padding.

When **LinePitch** > **LineWidth**, the line padding is enabled: the DMA engine inserts **LinePitch** - **LineWidth** bytes of padding at the end of each image line.

LinePitch can be set to **0** to disable padding after lines.

Stripe padding

Stripes are groups of adjacent lines. A stripe of height 1 is a line.


The **StripeHeight** and **StripePitch** features control the stripe padding.


When **StripePitch** > **StripeHeight**, the stripe padding is enabled: the DMA engine inserts **StripePitch** - **StripeHeight** lines of padding at the end of each stripe.

StripePitch can be set to **0** to disable padding after lines.

8.4. Horizontal Image Flipping

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3** and **PC3602-T Coaxlink Octo**:

 (1-camera, line-scan)

 (2-camera, line-scan), (2-camera, line-scan, custom-logic)

Description

Original image



Horizontally flipped image



Legend

 Image

 Buffer

Horizontal image flipping

The horizontal image flipping function, also known as horizontal mirroring, is a process in which an image is flipped along its vertical axis. This means that the left side of the image becomes the right side and vice versa.



NOTE

- Line padding is excluded from the operation.
- For RGB and RGBa, the component order remains unchanged

Data Stream setup

The **ReverseX** GenApi feature in the **ImageFormatControl Category** of the Coaxlink Data Stream module controls the horizontal image flipping.

- When left to its default value the image horizontal flipping is disabled.
- When set to **True**, the image horizontal flipping is enabled

Prerequisites

The horizontal image flipping function can be enabled only if following conditions area all satisfied

1. The product firmware variant provides the function.
2. The image width doesn't exceed 48 kilobytes, namely:
 - a. 49,152 monochrome or Bayer CFA pixels of 8-bit
 - b. 24,576 monochrome or Bayer CFA pixels of 10-/12-/14- or 16-bit

- c. 16,384 RGB pixels of 8-bit
- d. 8,192 RGB pixels of 10-/12-/14- or 16-bit
- e. 12,288 RGBa pixels of 8-bit
- f. 6,144 RGBa pixels of 10-/12-/14- or 16-bit

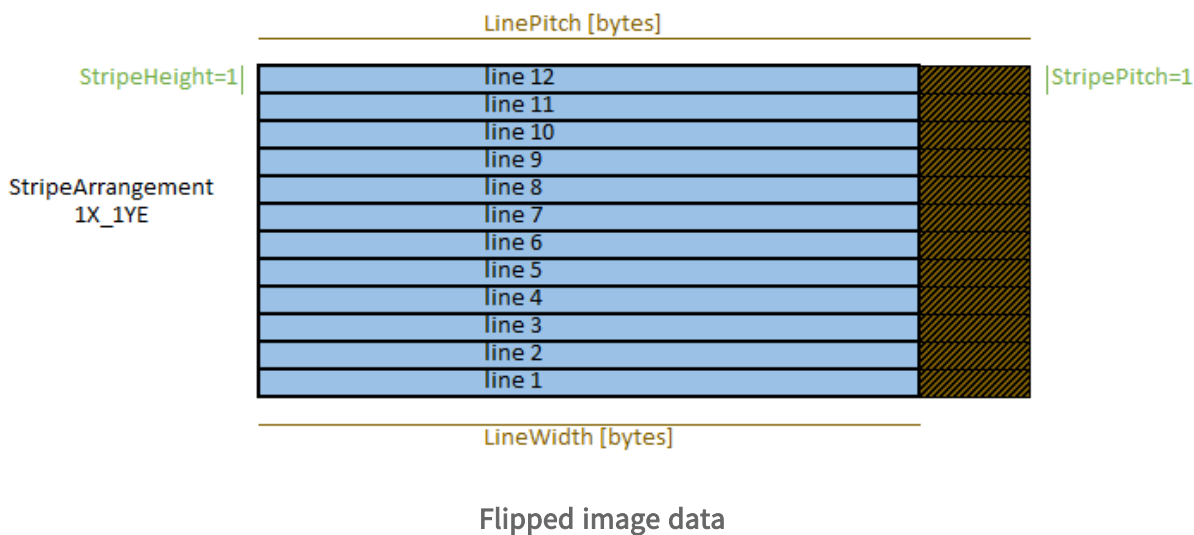
Limitations

When the horizontal image flipping function is enabled:

- 1. the Pixel Component Unpacking control is inoperative, the Coaxlink card unpacks 10-bit, 12-bit or 14-bit pixels to lsb
- 2. The Image width increment step is 4 Bytes
- 3. For Bayer CFA pixel formats , the color pattern changes as following:
 - a. BayerGB -> BayerBG
 - b. BayerBG -> BayerGB
 - c. BayerGR -> BayerRG
 - d. BayerRG -> BayerGR

8.5. Vertical Image Flipping

The DMA engine provides the capability to flip the image vertically.



The vertical image flip is controlled by the **StripeArrangement** feature of the data Stream module.

By default, **StripeArrangement** is set to **1X_1Y**: the vertical image flip is disabled.

When **StripeArrangement** is set to **1X_1YE**, the driver determines the position of the first image line in the buffer by using this formula:

$$\text{BufferBase} + (\text{ImageHeight} - 1) * \text{LinePitch}$$

As a result:

- if the buffer is too small, it is the **top** part of the image (as given by the camera) that will be lost;
- lines will start at **BufferBase + n * LinePitch**;
- only complete lines are transferred;
- if the buffer size is not a multiple of **LinePitch** bytes, some bytes at the end of the buffer will be left unchanged.



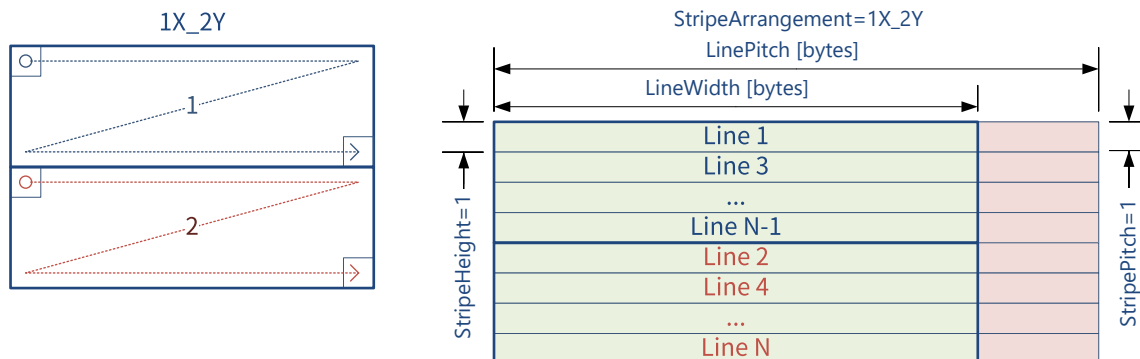
NOTE

When evaluating the above formula, if **LinePitch** is equal to 0, **LineWidth** will be used instead. Similarly, if **StripeHeight** is 0, 1 will be used instead.

8.6. Image Data Unscrambling

The DMA engine provides the capability to unscramble images having particular geometries along the Y-axis.

Unscrambling 1X_2Y images



When **StripeArrangement** is set to **Geometry_1X_2Y**, the driver determines the destination of the second line output by the camera (Line 2 in the above drawing) by using this formula:

$$\text{BufferBase} + (\text{BufferSize} + \text{LinePitch} - \text{LineWidth}) / (2 * \text{LinePitch}) * \text{LinePitch}$$

NOTE: this is the address of the first line in the second half of the buffer large enough to receive one complete line.

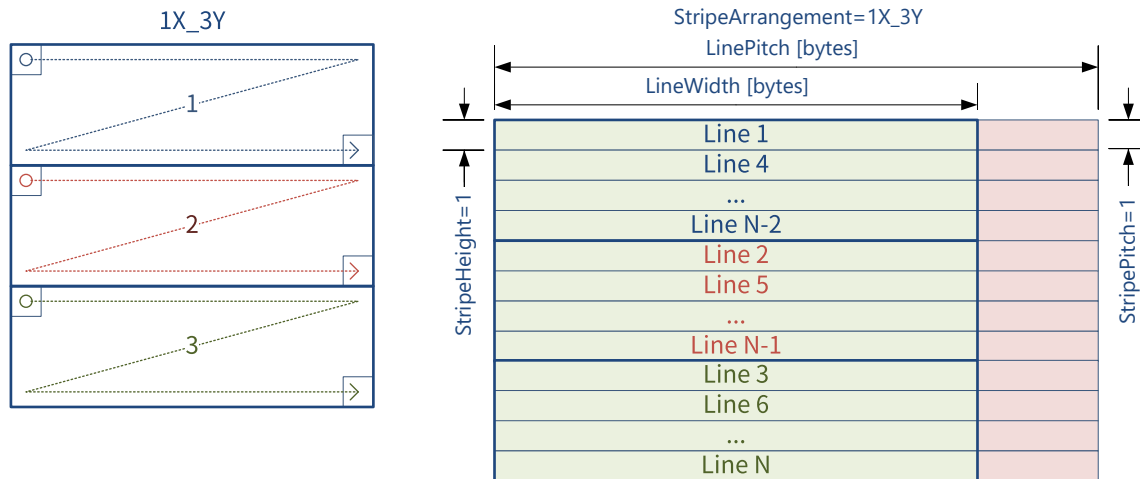
As a result:

- If the buffer is too small, the last lines output by the camera (i.e., the bottom part of each half of the image) will be lost; the application is responsible for avoiding this,
- Lines will start at

$$\text{BufferBase} + n * \text{LinePitch},$$

- Only complete lines are transferred. (When evaluating the above formula, if **LinePitch** is equal to 0, **LineWidth** will be used instead. Similarly, if **StripeHeight** is 0, 1 will be used instead.)
- StripeHeight** and **StripePitch** cannot be set to values greater than 1.

Unscrambling 1X_3Y images



When **StripeArrangement** is set to **Geometry_1X_3Y**, the driver determines the destination of the second line and third lines (Line 2 and Line 3 in the above drawing) by using these formulae:

$$\text{BufferBase} + (\text{BufferSize} + \text{LinePitch} - \text{LineWidth}) / (3 * \text{LinePitch}) * \text{LinePitch}$$

$$\text{BufferBase} + (\text{BufferSize} + \text{LinePitch} - \text{LineWidth}) / (3 * \text{LinePitch}) * \text{LinePitch} * 2$$

NOTE: these are the addresses of the first line in the second and third regions of the buffer large enough to receive one complete line.

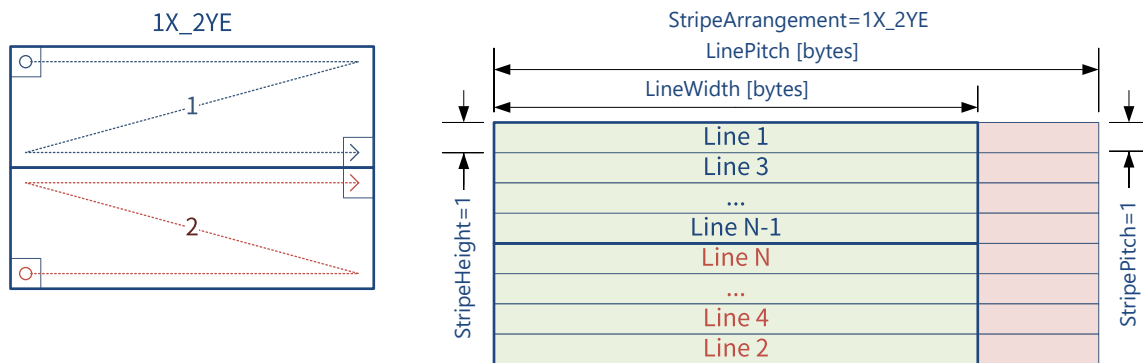
As a result:

- If the buffer is too small, the last lines output by the camera (i.e., the bottom part of each region of the image) will be lost; the application is responsible for avoiding this,
- Lines will start at

$$\text{BufferBase} + n * \text{LinePitch},$$

- Only complete lines are transferred. (When evaluating the above formula, if **LinePitch** is equal to 0, **LineWidth** will be used instead. Similarly, if **StripeHeight** is 0, 1 will be used instead.)
- StripeHeight** and **StripePitch** cannot be set to values greater than 1.

Unscrambling 1X_2YE images



When **StripeArrangement** is set to **Geometry_1X_2YE**, the driver determines the destination of the second line output by the camera (i.e., the position of last image line in the buffer) by using this formula:

$$\text{BufferBase} + (\text{ImageHeight} - 1) * \text{LinePitch}$$

As a result:

- If the buffer is too small, the last lines of the image will be lost; the application is responsible for avoiding this,
- Lines will start at

$$\text{BufferBase} + n * \text{LinePitch},$$

- Only complete lines are transferred. (When evaluating the above formula, if **LinePitch** is equal to 0, **LineWidth** will be used instead. Similarly, if **StripeHeight** is 0, 1 will be used instead.)
- **StripeHeight** and **StripePitch** cannot be set to values greater than 1.

See also: "Transmission Methods of 1X_2YE Images (Coaxlink series)" on page 180

Unscrambling 1X_2YM images



NOTE

- In the following figures, line numbers and block numbers sent by the *device* are in *white*, line numbers and block numbers received by the *host* are in *black*.
- The values of **StripePitch**, **StripeHeight** and **StripeOffset** must be multiple of **BlockHeight** value!

1X_2YM camera delivering lines by blocks of 2

StripeHeight=1

StripeArrangement=1X_2YM

BlockHeight=2

LinePitch [bytes]

line 11	block 6	line 11	block 6	<div>StripePitch=1</div>
line 9	block 5	line 9	block 5	
line 7	block 4	line 7	block 4	
line 5	block 3	line 5	block 3	
line 3	block 2	line 3	block 2	
line 1	block 1	line 1	block 1	
line 2	block 1	line 2	block 1	
line 4	block 2	line 4	block 2	
line 6	block 3	line 6	block 3	
line 8	block 4	line 8	block 4	
line 10	block 5	line 10	block 5	
line 12	block 6	line 12	block 6	

LineWidth [bytes]

1X_2YM camera delivering lines by blocks of 4

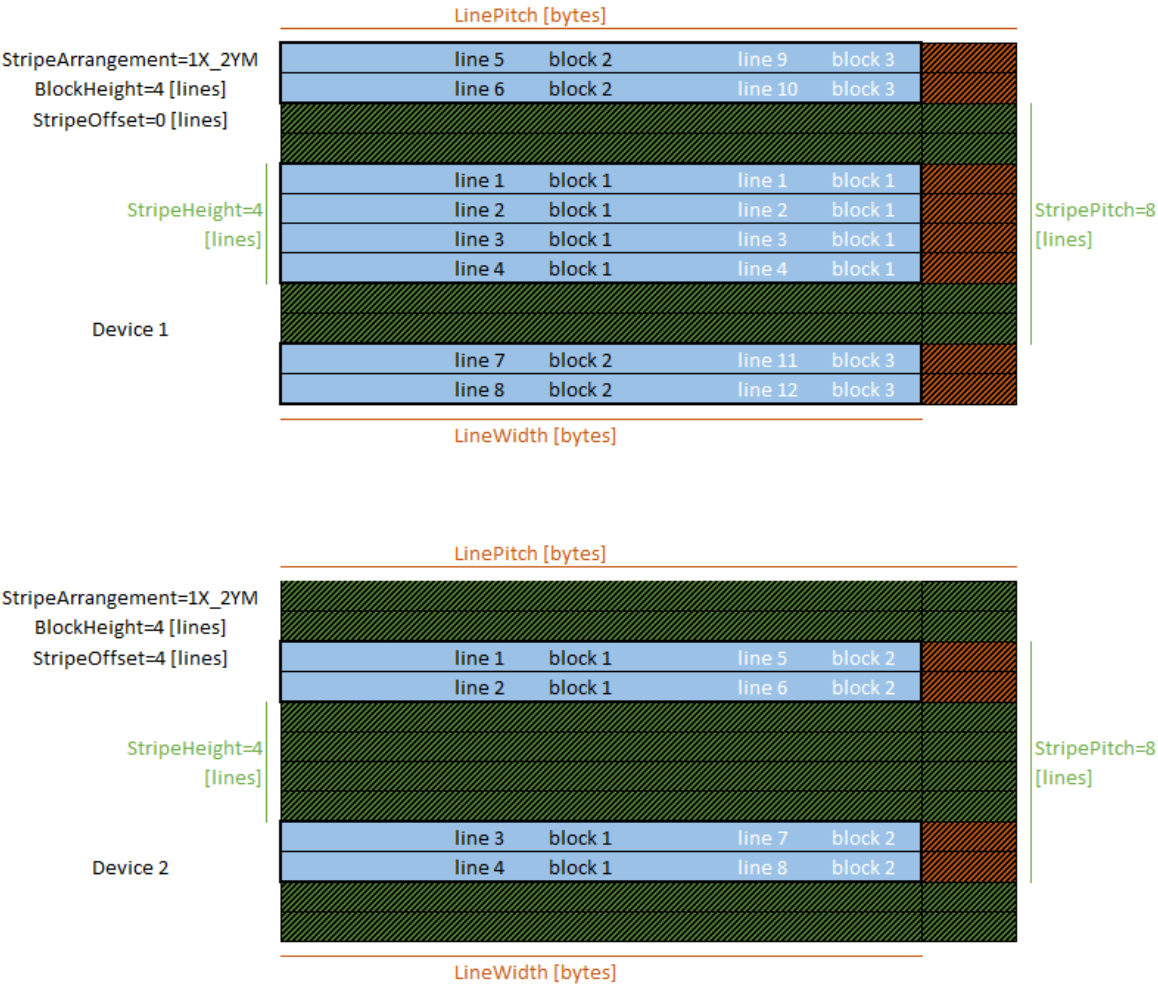
StripeHeight=1

StripeArrangement=1X_2YM

BlockHeight=4 [lines]

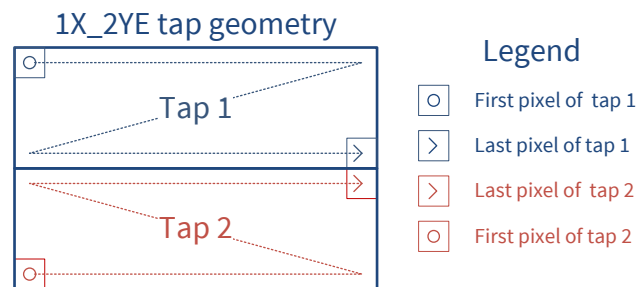
		LinePitch [bytes]					
line 9	block 3	line 9	block 3			StripePitch=1	
line 10	block 3	line 10	block 3				
line 5	block 2	line 5	block 2				
line 6	block 2	line 6	block 2				
line 1	block 1	line 1	block 1				
line 2	block 1	line 2	block 1				
line 3	block 1	line 3	block 1				
line 4	block 1	line 4	block 1				
line 7	block 2	line 7	block 2				
line 8	block 2	line 8	block 2				
line 11	block 3	line 11	block 3				
line 12	block 3	line 12	block 3				
		LineWidth [bytes]					

1X_2YM camera delivering lines by blocks of 4 to two hosts



8.7. Transmission Methods of 1X_2YE Images (Coaxlink series)

Geometry of 1X_2YE image sensors



An 1X_2YE image sensor has two taps:

- Tap 1 delivers the pixel data of the upper half of the image. It starts with the leftmost pixel of the top line and ends with the rightmost pixel of the bottom line of the upper half.
- Tap 2 delivers the pixel data of the lower half image lines. It starts with the leftmost pixel of the bottom line and ends with the rightmost pixel of the top line of the lower half.

Two methods are used by CoaXPress cameras to transmit images captured with 1X_2YE image sensors:

- "Method 1 - Transmission using a single CoaXPress data stream" on page 181
- "Method 2 - Transmission using a dedicated CoaXPress data stream for each tap" on page 182.

Method 1 - Transmission using a single CoaXPress data stream



NOTE

- This is NOT the method defined by the CoaXPress standard. However it is the most popular!
- The DMA engine of all area-scan firmware variants of all **Coaxlink frame grabbers** is capable to reorganize the lines and deliver an unscrambled image into the buffer.

The camera transmits the image data through a single CoaXPress data stream composed as follows:

Line 1 > Line H > Line 2 > Line H-1 > ... > Line H/2-1 > Line H/2 + 2 > Line H/2 > Line H/2 + 1

To deliver an unscrambled image into the buffer, configure the data stream as follows:

1. Keep default **DeviceTapGeometry** settings:
 - **DeviceTapGeometry=Geometry_1X_1Y**
2. Configure the DMA engine (as described in "[Unscrambling 1X_2YE images](#)" on page 177):
 - **StripeArrangement=Geometry_1X_2YE**
 - **LineWidth=<WIDTH>**
 - **LinePitch=<PITCH>** (only required when it is not equal to <WIDTH>)

Method 2 - Transmission using a dedicated CoaXPress data stream for each tap

Applies to the following firmware variants of PC3623-T Coaxlink Quad CXP-12 Value:

 (1-camera)



NOTE

- This is the method defined by the CoaXPress standard.
- Since this method requires more FPGA resources it is only supported by above-listed firmware variants.

The camera transmits the image data through two CoaXPress data streams:

Stream 1: Line 1 > Line 2 > ... > Line H/2 - 1 > Line H/2

Stream 2: Line H > Line H - 1 > ... > Line H/2 + 2 > Line H/2 + 1

To deliver an unscrambled image into the buffer, configure the data stream as follows:

1. Merge the 2 streams (as described in "Multi-tap CoaXPress Cameras" on page 77):
 - DeviceTapGeometry=Geometry_1X_2YE
 - Image1StreamID=... (depends on the camera, typically 0)
 - Image2StreamID=... (depends on the camera, typically 1)
2. Configure the DMA engine (as described in "Unscrambling 1X_2YE images" on page 177):
 - StripeArrangement=Geometry_1X_2YE
 - LineWidth=<WIDTH>
 - LinePitch=<PITCH> (only required when it is not equal to <WIDTH>)



NOTE

If an EGrabber object is created from EGrabberDiscovery::cameras, and, if a "1X_2YE (Method 2)" camera is detected, eGrabber automatically configures the data streams as described earlier. For instance, this is the case when, in **eGrabber Studio**, the user creates a grabber using the [Cameras](#) view.

In any other case, the configuration must be performed by the user.

See also: "Discovering grabbers and camera's" section in the D204ET-eGrabber Programmer Guide-eGrabber-25.07.1.2202.pdf document

8.8. Data Stream Statistics

The stream statistics tool monitors the image data stream at the card output and provides the application with averaged frame-, line- and data-rate.

Stream Statistics Sampling Methods

The `StatisticsSamplingSelector` determines the *averaging interval*. It can be any of the following:

- `LastSecond` or `LastTenSeconds`: The last completed *time* slot of 1 or 10 seconds.
- `Last2Buffers`, `Last10Buffers`, `Last100Buffers`, `Last1000Buffers`: The last 2, 10, 100, or 1000 *acquired buffers*
- `LastAcquisition`: The last acquisition activity period. Namely since the last `DStartAcquisition()` function call until now, if the acquisition is still active otherwise until the last `DStopAcquisition()` function call.
- `LastAcquisition`: Time interval between `StatisticsStartSampling` and `StatisticsStopSampling` commands.

The default sampling method is `LastSecond`.

Statistical Data

The statistical data is effectively computed when getting any of the following feature:

- `StatisticsFrameRate` reports the averaged frame rate expressed in frames/second (area-scan).
- `StatisticsLineRate` reports the average line rate expressed in lines/second (line-scan).
- `StatisticsDataRate` reports the average data rate expressed in megabytes/second

For every GenTL buffer filled during the averaging interval, the tool counts:

- The number of filled GenTL buffers and the corresponding number of frames (area-scan) or lines (line-scan)
- The number of transferred bytes of image data.

The related GenApi features are gathered into the Stream Statistics Category of the GenTL Data Stream Module.

8.9. Data Transfer Rate Test Program

Introduction

The *Data Transfer Rate Test Program* (DTR) can be used to measure the effective PCI Express data transfer rate in real conditions.

Host PC requirements

- The Host PC must be equipped with at least one **eGrabber**-driven frame grabber.
- Driver version 12.4 or higher must be installed on the Host PC.

Camera requirements

- The camera must be configured to deliver continuously image data.

Installation

The DTR is included in gentl.exe, a command-line tool that is delivered with the **eGrabber** driver. No further installation is required.

Measurement principle

The DTR measures the data transfer rate by completely filling the internal frame store and only then transferring images to the host computer:

1. All buffers are unqueued (the data stream cannot use them)
2. The data stream and remote device are started
3. When the frame store is full, the remote device is stopped
4. Current timestamp is retrieved (t0)
5. All buffers are queued to the data stream and transfers start
6. Buffers are popped from the data stream
7. When the frame store is empty and all buffers have been retrieved, the data stream is stopped

The DTR program computes the data transfer rate as follows:

```
- byte count = sum of each buffer's BUFFER_INFO_SIZE_FILLED  
- t1 = last buffer's BUFFER_INFO_CUSTOM_EVENT_TIMESTAMP  
- duration = t1 - t0  
- data transfer rate = byte count / duration
```

gentl --help

GentL Explorer

gentl [COMMAND] ... [OPTIONS]

Commands:

info	Show detailed information about the transport layer system
report	Generate a GentL report archive (for Euresys tech support)
xml	Download GenApi files (XML register descriptions)
play	Open a data stream and acquire images (no display)
view	Open a data stream and display images
grab	Grab N images
genapi	Enter the GenApi command-line interface or perform a GenApi operation
read	Read data from a GentL port
write	Write data to a GentL port
event	Wait for events and display information about them
script	Execute script
run	Run an action
dtr	Measure PCIe data transfer rate
ber	Measure bit error rate confidence level (a.k.a. link validation tool)

Common flags:

	--cti=LIBPATH	Path to GentL producer library. Default: use EURESYS_COAXLINK_GENTL64_CTI and GENICAM_GENTL64_PATH environment variables to locate the library.
-j=N		Limit the number of CPU cores to use to N (default: 2)
-h	--help	Display help message
-V	--version	Print version information
	--numeric-version	Print just the version number
-v	--verbose	Loud verbosity
-q	--quiet	Quiet verbosity

gentl dtr --help

```

GenTL Explorer

gentl dtr [OPTIONS]

Flags:
    --if=ID                Interface ID
    --dev=ID               Device ID
    --ds=ID                Data stream ID
    --device-access=ACCESS Access flags used to open the device (GenTL standard
access flags:
                                DEVICE_ACCESS_READONLY, DEVICE_ACCESS_CONTROL,
                                DEVICE_ACCESS_EXCLUSIVE;
                                Coaxlink custom access flags:
                                DEVICE_ACCESS_CUSTOM_READONLY_DEVICE_READONLY_
STREAM) (default:
                                DEVICE_ACCESS_CONTROL)
    --ro                  Open the device as read-only (shorthand for
                                --device-access=DEVICE_ACCESS_READONLY)
    --buffers=INT         Buffer count (default: 4)
    --buffersize=INT      Buffer size
    --width=WIDTH         Buffer width
    --height=HEIGHT       Buffer height
    --pixelformat=ITEM    PFNC Pixel format
    --bayer=BAYERDECODINGMETHOD Bayer method (Legacy, Advanced) (default: Advanced)
    --set=SETTINGS        GenApi settings, such as Module.Feature=INT
    --setup=FILE           Path to script to execute before starting stream
    --run=FILE             Path to script to execute concurrently
    --timeout=INT         Acquisition timeout, in milliseconds (default:
infinite)
    --zero                Zero memory when queuing buffers (default: memory is
only zeroed when buffers
                                are allocated)
    --remotexml=FILE      Use FILE as register description (default: register
description is read from
                                remote device)
    -n --repeat[=N]       Measure data transfer rate N times (default: 1)
Common flags:
    --cti=LIBPATH         Path to GenTL producer library. Default: use
EURESYS_COAXLINK_GENTL64_CTI and
                                GENICAM_GENTL64_PATH environment variables to locate
the library.
    -j=N                 Limit the number of CPU cores to use to N (default:
2)
    -h --help            Display help message
    -V --version          Print version information
    --numeric-version     Print just the version number
    -v --verbose         Loud verbosity
    -q --quiet           Quiet verbosity

```



TIP

For a better measurement accuracy, use the `gentl dtr -n` option to execute multiple measurements repeatedly. The DTR program will average the results.

9. Camera and Illumination Control

9.1. Camera Control Principles	188
Camera Cycle	189
Camera Cycle Concatenation Rules	190
Camera Control Methods	192
9.2. Illumination Control Principles	195
Illumination Devices	196
Aligning Camera and Illumination Cycles	197
9.3. Camera and Illumination Controller	198
Camera and Illumination Controller Overview	198
Cycle Timing Machine	199
Multiple Cycle Timings	201
Cycle Manager	202
Cycle Trigger Manager	203
Sequence Manager	205
CIC Output Signals Routing	207
9.4. CIC Timing Diagrams	208
Single Cycle	209
Overlapping Cycles - Single timing	211
Multiple Timings Example	216
Cycle Sequence Timing Diagrams	218

9.1. Camera Control Principles

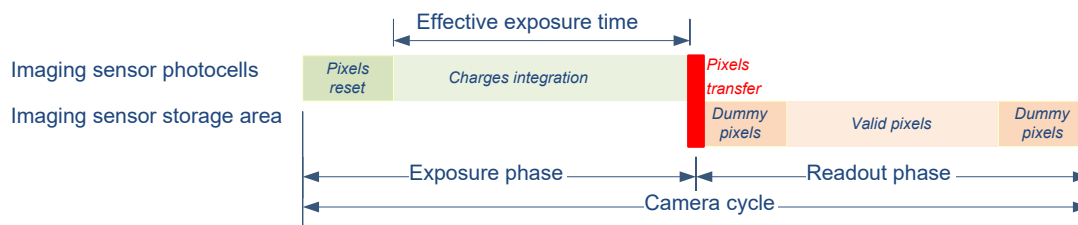
Camera Cycle 189

Camera Cycle Concatenation Rules 190

Camera Control Methods 192

Camera Cycle

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



Camera cycle

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 “pixels 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).

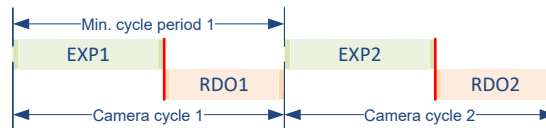
Some sensors provide the capability to select one or more region of interest (ROI) speeding up the readout since less data needs to be transferred.

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.



Shortest possible cycle period achievable by cameras NOT allowing the cycle overlapping

$$\text{Min. cycle period}_n = \text{EXP}_n + \text{RDO}_n$$

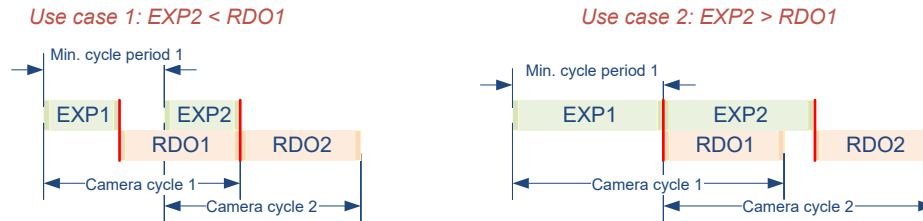


NOTE

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

Rules for cameras allowing overlapping

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



Shortest possible cycle period achievable by cameras allowing the cycle overlapping

In the first case, the duration of the exposure phase of the second 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

Camera control methods of Coaxlink frame grabbers

Coaxlink frame grabbers provide five camera control methods selected via the **CameraControlMethod** feature of the GenTL Device module. They are named **NC**, **RC**, **RG**, **INTERNAL** and **EXTERNAL**.

NC camera control method

The **NC** camera control method 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.



WARNING

The Camera and Illumination Controller (CIC) is not used!

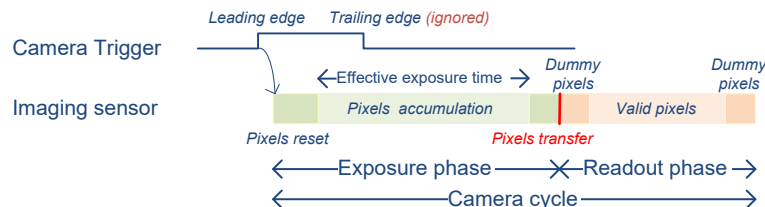
- There is no **Camera Trigger** signal produced by the CIC. The frame grabber do NOT control the camera cycles.
- There is no **Strobe** signal produced by the CIC. The frame grabber do NOT control the illumination.

The external controller is entirely responsible for the camera cycle timings!

RC camera control method

The **RC** camera control method 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 the frame grabber.



Grabber controlled camera cycle using the RC camera control method

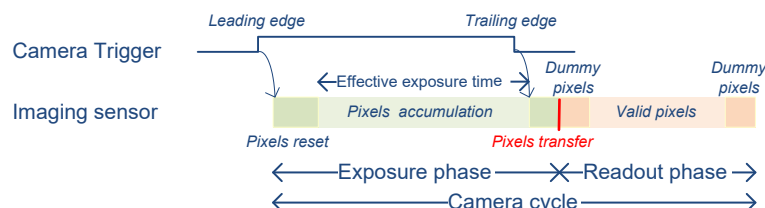
The CIC produces one single Camera Trigger pulse every camera cycle. The Camera Trigger leading edge triggers a new camera cycle and initiates a new exposure period. The Camera Trigger trailing edge is ignored by the camera.

On **PC1628 Grablink Duo**, the pulse width is configurable through the **ClTriggerDuration** of the Device Module.

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 the frame grabber.



Grabber controlled camera cycle using the RG camera control method

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 trailing edge terminates the exposure period and triggers the readout.

INTERNAL camera control method

The **INTERNAL** camera control method targets asynchronous reset cameras that are controlled by a hardware signal applied by I/O Toolbox events mapped to CoaXPress LinkTrigger messages.

See also: "Host to Device Trigger Source" on page 56 to select a GPIO input port as trigger source.



NOTE

There is a CameraTrigger signal that is generated upon external trigger signal. The CameraTrigger signal is used to initiate a data packet, even though the signal timing is not directly controlled by the CIC.



WARNING

The Camera and Illumination Controller (CIC) is not used!

- There is no Camera Trigger signal produced by the CIC. The frame grabber do NOT control the camera cycles.
- There is no Strobe signal produced by the CIC. The frame grabber do NOT control the illumination.

The external controller is entirely responsible for the camera cycle timings!

EXTERNAL camera control method

The **EXTERNAL** camera control method targets asynchronous reset cameras that are controlled by a hardware signal applied by an external controller to any GPIO input port of the grabber.

See also: "Host to Device Trigger Source" on page 56 to select a GPIO input port as trigger source.



NOTE

There is a CameraTrigger signal that is generated upon external trigger signal. The CameraTrigger signal is used to initiate a data packet, even though the signal timing is not directly controlled by the CIC.



WARNING

The Camera and Illumination Controller (CIC) is not used!

- There is no Camera Trigger signal produced by the CIC. The frame grabber do NOT control the camera cycles.
- There is no Strobe signal produced by the CIC. The frame grabber do NOT control the illumination.

The external controller is entirely responsible for the camera cycle timings!

9.2. Illumination Control Principles

Illumination Devices196

Aligning Camera and Illumination Cycles197

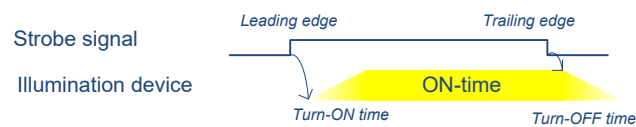
Illumination Devices

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.



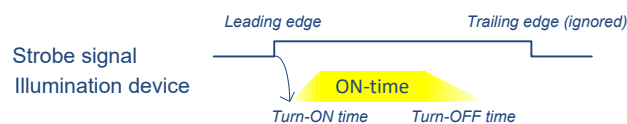
Timing diagram for intermittent illumination devices

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.



Timing diagram for strobed illumination devices

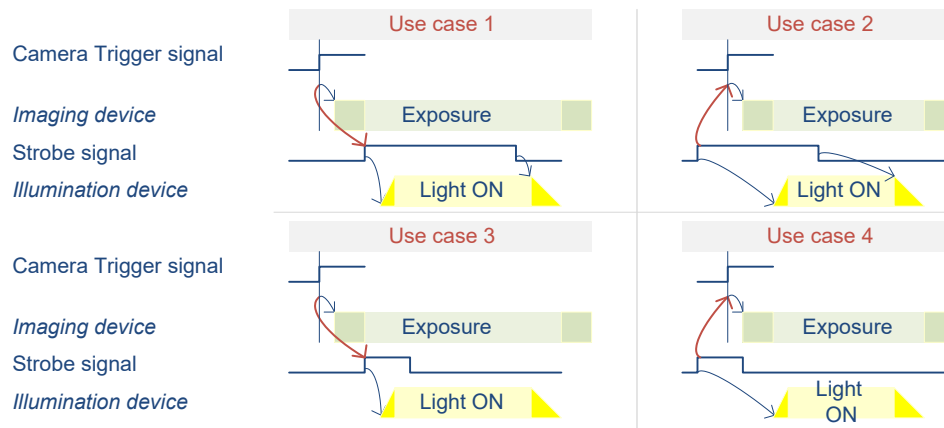
The on-time duration is either uncontrolled or controlled by the illumination device itself.

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.



4 typical use cases of Camera Trigger vs. Strobe alignment

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.

9.3. Camera and Illumination Controller

Camera and Illumination Controller Overview198

Cycle Timing Machine199

Multiple Cycle Timings201

Cycle Manager202

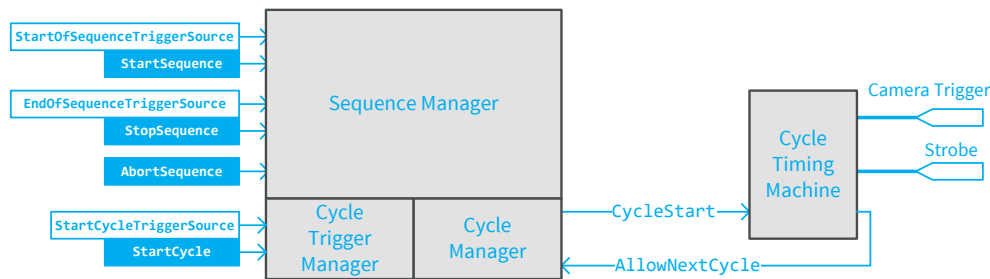
Cycle Trigger Manager203

Sequence Manager205

CIC Output Signals Routing207

Camera and Illumination Controller Overview

The Camera and Illumination Controller (abbreviated as CIC) controls one camera and its associated illumination.



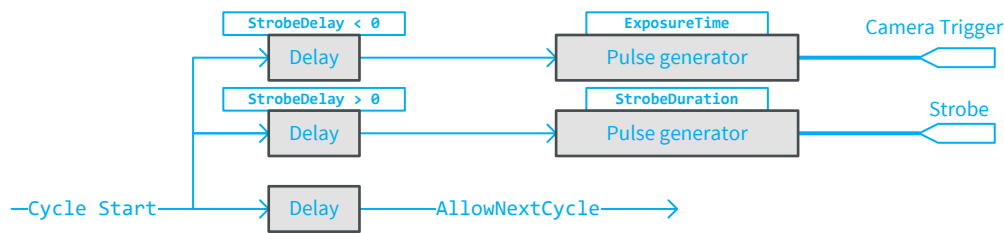
Camera and Illumination Controller block diagram

The CIC is composed of 4 main interconnected blocks:

- The **Cycle Timing Machine** is responsible for the generation of accurately timed events and signals structuring one camera and illumination controller cycle (CIC Cycle), namely: **Camera Trigger** and **Strobe**.
- The **Cycle Manager** is responsible for the generation of the **CycleStart** event. It prevents initiating a new cycle while the *start cycle conditions* are not all satisfied and while the cycle timing machine does not allow a new cycle to begin.
- The **Cycle Trigger Manager** is responsible, in collaboration with the *Cycle Manager* and the *Sequence Manager*, to elaborate the effective **CycleStart** event that initiates one cycle of the *Cycle Timing Machine*.
- The **Sequence Manager** manages sequences of CIC cycles according to user-defined start sequence and stop sequence conditions.

The camera is controlled with the **Camera Trigger** signal and the illumination device is controlled with the **Strobe** signal. **Several routing options** are available.

Cycle Timing Machine



CIC Cycle Timing Machine block diagram

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.

Intra-cycle timing

Three GenApi features of the Device module are used to configure the timing of the output signals within a cycle:

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

See also: "Single Cycle" on page 209 for more explanations and timing diagrams

Cycle-to-cycle timing

The `AllowNextCycle` event is 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 driver according to the following user settings:

- The `ExposureReadoutOverlap` feature of the Camera Model category defines if the camera supports or not the exposure/readout overlapping. If overlapping is allowed, the `AllowNextCycle` event is issued earlier and faster cycle rates are obtained.
- The `ExposureRecoveryTime` feature of the Camera Model category defines the minimum time gap required by the camera between two exposures. This feature is relevant when `ExposureReadoutOverlap = TRUE` and the duration of the exposure phase becomes larger than the duration of the readout phase.
- The `CycleMinimumPeriod` of the Cycle Control category defines the minimum cycle period. This value may not be smaller than the time required by the camera to perform the image readout!



NOTE


Some cameras have a data store in the image data path. This enables capturing bursts of images at a higher cycle-to-cycle rate than the camera-to-frame grabber data link can sustain. In that case, `CycleMinimumPeriod` declares the smallest cycle-to-cycle time that the image sensor can achieve!


See also: "Overlapping Cycles - Single timing" on page 211 for more explanations and timing diagrams.

Multiple Cycle Timings

Applies to the following firmware variants of **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo** and **PC3623-T Coaxlink Quad CXP-12 Value**:

 (1-camera)

 (1-camera), (1-camera, custom-logic), (2-camera), (3-camera), (4-camera), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera)

 (1-camera), (1-camera, line-scan), (2-camera), (3-camera), (4-camera)

In the Cycle Timing Category of the Device Module, the **CycleTimingSelector** parameter acts as a selector for **ExposureTime**, **StrobeDelay** and **StrobeDuration**.

The CIC of above listed firmware variants allows to define up to 16 different timing sets!

By default, **CycleTimingSelector** = 1.

To activate the *multiple cycle timings* feature, set the **CycleTimingCount** parameter to any value between 2 and 16 according to the number of cycle timings to be executed.

The CIC uses successively and repeatedly the different timing definitions starting with index 0 up to the value of **CycleTimingCount** -1.

See also: "Multiple Timings Example" on page 216 for more explanations and timing diagrams.

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* listed hereafter are not satisfied:

Sequence Active Condition

The Sequence Manager must be in the ACTIVE state: the sequence has started and the sequence stop condition has not yet been reached.

This condition always applies.

Next Cycle Allowed Condition

The Cycle Timing Machine has already issued the `Allow Next Cycle` event of the previous cycle.

This condition always applies.

Free Memory Condition

There is enough free memory on board to acquire the image data of the next cycle.

This condition always applies.

Cycle Trigger Event Condition

A new Cycle Trigger event is required to initiate a new cycle.

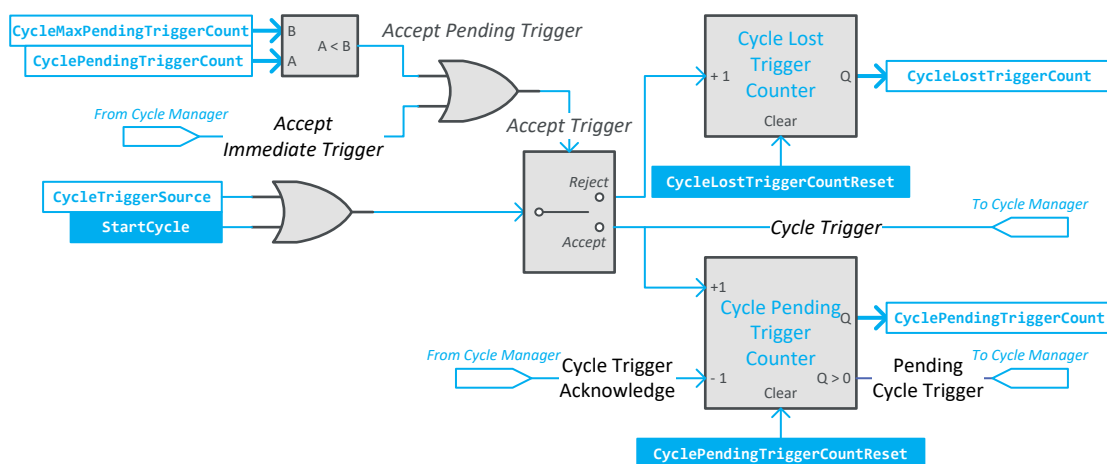
This condition applies only when `Cycle TriggerSource` \neq `Immediate` AND `CycleMaxPendingTriggerCount` = 0.

Pending Trigger Condition

There is at least one pending trigger (possibly a new one) waiting for service.

This condition applies only when `Cycle TriggerSource` \neq `Immediate` AND `CycleMaxPendingTriggerCount` > 0.

Cycle Trigger Manager



CIC Cycle Trigger Manager block diagram

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 Sources

The source of Cycle Trigger is defined by `CycleTriggerSource`.

When set to **Immediate**, the Cycle Trigger Manager is self-triggered. It generates a Cycle Trigger immediately after the start of the sequence and then repeatedly every `CycleMinimumPeriod` period.

When set to **StartCycle** or to *<any I/O toolbox event source>* the Cycle Trigger Manager doesn't start immediately after the start of the sequence, instead it waits for the execution of a **StartCycle** command or the occurrence of an event on the selected I/O toolbox event source.

A wide set of Cycle Trigger event sources is available. It includes all the I/O toolbox events, namely: LIN, QDC, MDV, DIV, DEL, EIN and User Events.

Cycle Trigger Latch Mechanism

The Cycle Manager is fitted with a trigger latch mechanism capable of latching cycle triggers that cannot be served immediately. Such triggers are named "pending triggers" since their execution is simply postponed until the corresponding CIC cycle is initiated.

The maximum number of pending triggers that can be recorded is defined by `CycleMaxPendingTriggerCount`. When `CycleMaxPendingTriggerCount = 0`, the trigger latching mechanism is disabled. This is the default value. To enable the trigger latching mechanism, set `CycleMaxPendingTriggerCount` to any integer value in range 1 to 7.

The number of pending triggers is reported by `CyclePendingTriggerCount`.

Cycle Trigger Events Sorting

When `CycleTriggerSource` is set to `StartCycle` or to *<any I/O toolbox event source>*, every Cycle Trigger event is evaluated against the *trigger acceptance criteria* and sorted according to the result.

The rejected Cycle Trigger events increment the *Cycle Lost Trigger Counter*.

The accepted Cycle Trigger events increment the *Cycle Pending Trigger Counter* if the pending trigger cannot be served immediately.

Trigger Acceptance Criteria

Cycle Trigger events are *accepted and executed immediately* when both conditions are satisfied:

- Cycle Sequence is active
- Cycle Manager is currently waiting for an immediate trigger event to start a new cycle (Accept Immediate Trigger)

Cycle Trigger events are *accepted and executed later* when following conditions are satisfied:

- Cycle Sequence is active.
- The number of pending triggers, `CycleMaxPendingTriggerCount`, is less than `CycleMaxPendingTriggerCount`.
- The Cycle Manager is not (yet) ready to initiate new cycle.

Sequence Manager

The *Sequence Manager* is the top-level manager of the CIC: It controls the Cycle Trigger Manager and the Cycle Manager.

It defines sequences of identical CIC cycles according to user-defined start sequence and stop sequence conditions.

Starting a Sequence

The conditions for starting a sequence are defined by **StartOfSequenceTriggerSource**.

When **StartOfSequenceTriggerSource** is set to **Immediate** (default setting), the Sequence Manager doesn't require any further action to allow the Cycle Manager and the Cycle Trigger Manager to proceed with the first cycle.

Depending on the **CycleTriggerSource** settings of the Cycle Manager the first cycle will be executed:

- Immediately when **CycleTriggerSource** is set to **Immediate**
- On execution of the **StartCycle** command when **CycleTriggerSource** is set to **StartCycle** or
- On execution of the **StartCycle** command or when an event occurs on the I/O toolbox event source designated by **CycleTriggerSource**.

When **StartOfSequenceTriggerSource** is set to **StartSequence**, the Sequence Manager waits for the execution of a **StartSequence** command before allowing the Cycle Manager and the Cycle Trigger Manager to proceed with the first cycle.

When **StartOfSequenceTriggerSource** is set to **<any-event-source>**, the Sequence Manager waits for the execution of a **StartSequence** command or the occurrence of an I/O toolbox event on the designated event source before allowing the Cycle Manager and the Cycle Trigger Manager to proceed with the first cycle.

Stopping a sequence

The conditions for stopping a sequence are defined by **EndOfSequenceTriggerSource**.

When **EndOfSequenceTriggerSource** is set to **StopSequence** (default setting), the Sequence Manager stops the sequence at the next cycle boundary after the execution of a **StopSequence** command.

When **EndOfSequenceTriggerSource** is set to **SequenceLength**, the Sequence Manager stops automatically the sequence after having executed a number of camera cycles specified by **SequenceLength**. The sequence can be stopped anticipatively on execution of the **StopSequence** command. The default **SequenceLength** value is 1; any value up to 16,777,215 is allowed.

When **EndOfSequenceTriggerSource** is set to **<any-event-source>**, the Sequence Manager waits for the execution of a **StopSequence** command or the occurrence of an I/O toolbox event on the designated event source before stopping the sequence at the next cycle boundary.

NOTE: Any combination of **StartOfSequenceTriggerSource** and **EndOfSequenceTriggerSource** settings is allowed.

Changing sequence length while camera is grabbing

Starting with release 10.5, if **SequenceLength** is changed between start-of-sequence and end-of-sequence events, the new value will be effective for the subsequent sequence.

NOTE: The value of **SequenceLength** is latched at the start-of-sequence event.

CIC Output Signals Routing

Camera Trigger signal

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 frame grabber 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

See also: "CoaXPress Host To Device Trigger" on page 56 for detailed information about the Host to Device Trigger transmitter (Coaxlink series only).

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

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.

9.4. CIC Timing Diagrams

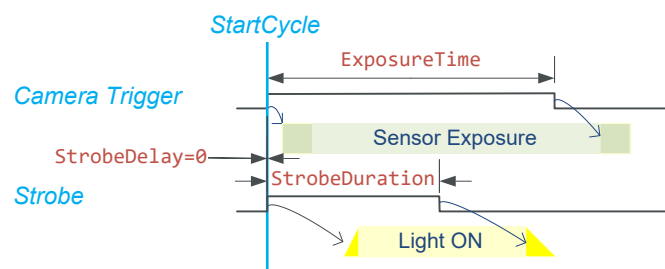
- Single Cycle209
- Overlapping Cycles - Single timing211
- Multiple Timings Example216
- Cycle Sequence Timing Diagrams218

Single Cycle

Timing diagrams of single CIC cycles

This topic shows timing diagrams of individual CIC cycles and illustrates the **ExposureTime**, **StrobeDuration** and **StrobeDelay** features for 3 use cases corresponding to positive, zero and negative values of **StrobeDelay**.

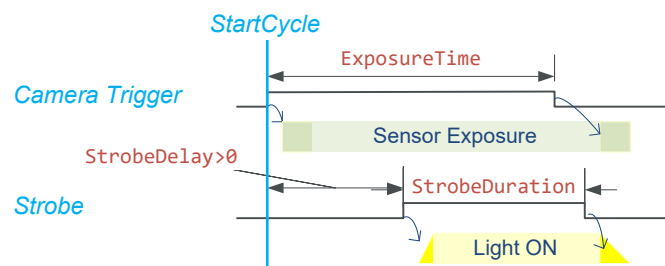
In-phase Strobe



In-phase Strobe signal

The **Camera Trigger** and the **Strobe** signals go high immediately after the **StartCycle** event.

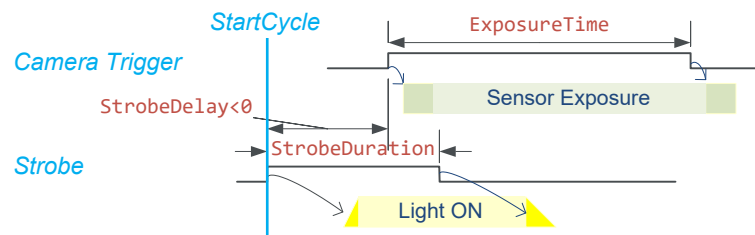
Late Strobe



Late Strobe signal

The **Camera Trigger** signal goes high immediately after the **StartCycle** event and the **Strobe** signal goes high after **StrobeDelay** microseconds.

Early Strobe



Early Strobe signal

The **Strobe** signal goes high immediately after the **StartCycle** event and the **Camera Trigger** signal goes high after a time delay equal to the opposite value of **StrobeDelay** microseconds.

Overlapping Cycles - Single timing

Timing diagrams of overlapping CIC cycles with identical cycle timing definition

This topic describes the behavior of the CIC when it is configured to generate one sequence of 4 cycles, each with identical timings settings.

In the following timing diagrams, user-defined values are shown in red:

- **c** is the minimum cycle period defined by **CycleMinimumPeriod**,
- **d** is the **Strobe** delay defined by **StrobeDelay**,
- **e** is the **Camera Trigger** pulse width defined by **ExposureTime**,
- **r** is the minimum time interval between consecutive exposure defined by **ExposureRecoveryTime**,
- **s** is the **Strobe** pulse width defined **StrobeDuration**.

In the following timing diagrams, values calculated by the driver are shown in blue:

- **a** is the CIC cycle duration,
- **f** is the time interval between consecutive **Camera Trigger** pulses,
 - the **Strobe** pulse width. This is the value of **StrobeDuration** set by the user.

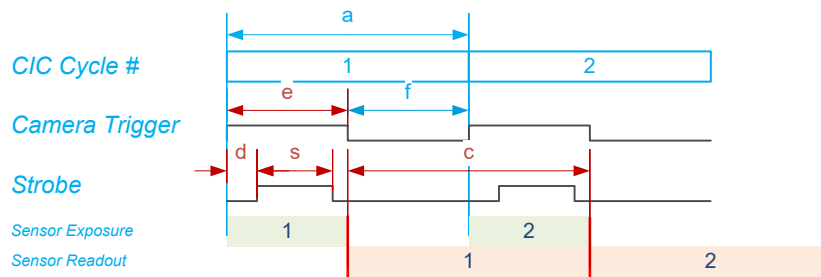
The driver calculates the duration of the CIC Cycle (**a** value) from the user-defined settings **ExposureTime**, **ExposureRecoveryTime** and **CycleMinimumPeriod** by searching the smallest value satisfying the following conditions:

- **Condition 1:** The time interval between consecutive **Camera Trigger** pulses (**f** value) must be greater than or equal to the **ExposureRecoveryTime** settings (**r** value). 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** value) 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 (**a** value) 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.

Case 1: Readout-limited - Late Strobe



The camera cycle rate is only limited by the camera readout time

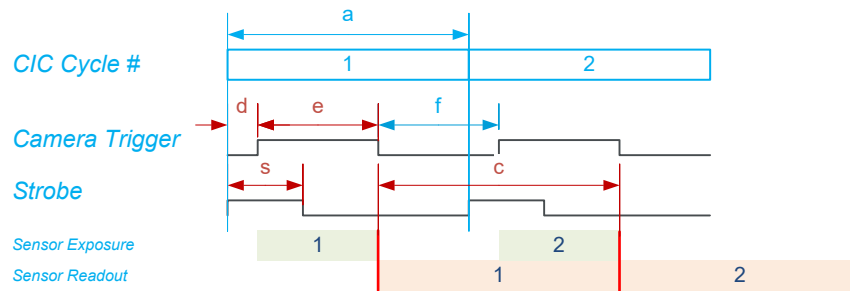
This situation occurs when the exposure time (**e** value) is significantly smaller than the readout duration (**c** value). In that situation:

- **f** is likely larger than **ExposureRecoveryTime**: *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 driver to calculate the cycle duration.

The optimal duration of the CIC Cycle is equal to the effective duration of the sensor readout phase. This is obtained when the user sets **CycleMinimumPeriod** 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



The camera cycle rate is only limited by the camera readout time (despite the early strobe)

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

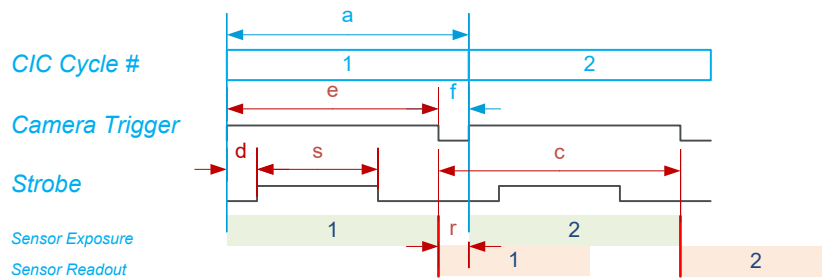
This situation occurs when the exposure time (**e** value) is significantly smaller than the readout duration (**c** value). In that situation:

- **f** is likely larger than **ExposureRecoveryTime**: *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 $(d + e < c)$.
- The *Condition 2* is the only condition used by the driver to calculate the cycle duration.

The optimal duration of the CIC Cycle is equal to the effective duration of the sensor readout phase. This is obtained when the user sets **CycleMinimumPeriod** 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



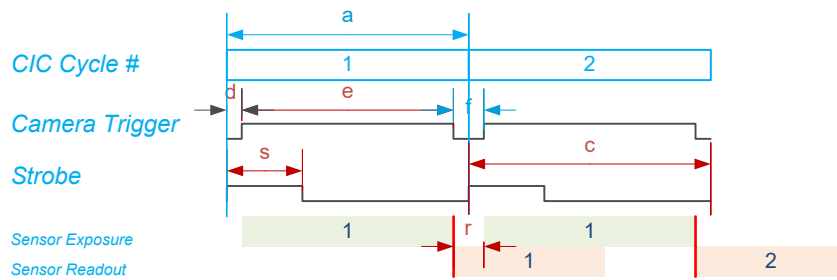
The camera cycle rate is limited by the exposure time settings

This situation occurs when the exposure time (**e** value) is significantly larger than the readout duration (**c** value). 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 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 **ExposureRecoveryTime** to a value corresponding to the minimal time interval allowed by the camera between consecutive Camera Trigger pulses.

Case 4: Exposure-limited- Early Strobe



The camera cycle rate is limited by the exposure time settings (despite the 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 in case of small negative **StrobeDelay** values.

This situation occurs when the exposure time (**e** value) is significantly larger than the readout duration (**c** value). 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 driver to calculate the cycle duration.

The user must set **ExposureRecoveryTime** to a value corresponding to the largest of the following two values:


- Minimal time interval allowed by the camera between consecutive **Camera Trigger** pulses.
- Opposite value of **StrobeDelay**.


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

Multiple Timings Example

Applies to the following firmware variants of PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value:

 (1-camera)

 (1-camera), (1-camera, custom-logic), (2-camera), (3-camera), (4-camera), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera)

 (1-camera), (1-camera, line-scan), (2-camera), (3-camera), (4-camera)

This topic describes the behavior of the CIC when it is configured to generate one sequence of 4 cycles, each with different timings settings.

Camera model parameters

- CameraControlMethod=RG;
- ExposeReadoutOverlap=TRUE;
- ExposureRecoveryTime set to *r* time interval in the drawing
- CycleMinimumPeriod set to *c* time interval in the drawing

Cycle timing parameters

CycleTimingCount=4

CycleTimingSelector	0	1	2	3
ExposureTime	e1	e2	e3	e4
StrobeDelay	0	0	0	0
StrobeDuration	e1	e2	e3	e4



NOTE

1. In this example, the Strobe signal is identical to the Camera Trigger signal
2. Exposure times (e1, e2, e3, e4) are all different, starting with the smallest and ending with the largest.

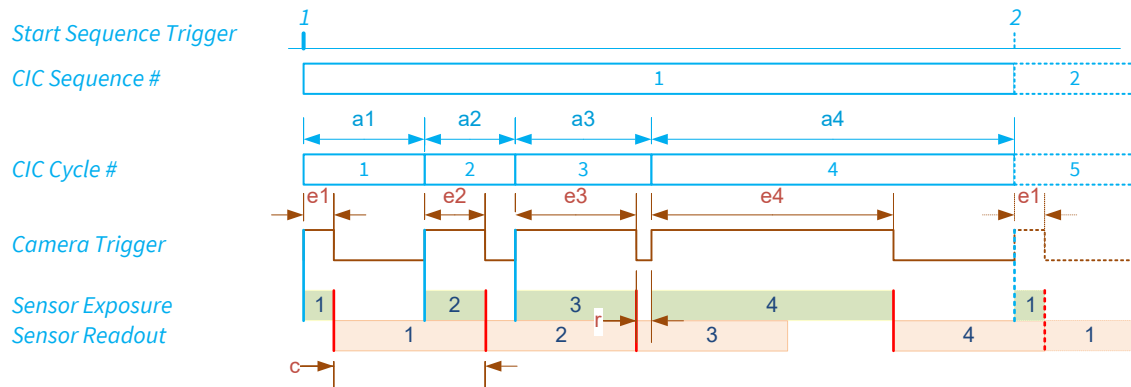
Cycle control parameters

- CycleTriggerSource=Immediate;

Sequence Control parameters

- StartOfSequenceTriggerSource≠Immediate;
- EndOfSequenceTriggerSource=SequenceLength;
- SequenceLength=4;

Description



Optimal sequence of 4 cycles with different timing settings

The sequence starts with the Start Sequence Trigger event (#1 in the drawing) and stops automatically after 4 cycles.

The next sequence may only start after the completion of the first sequence.

The four cycles of the sequence are overlapped with the minimum time interval between cycles according to "Camera Cycle Concatenation Rules" on page 190.

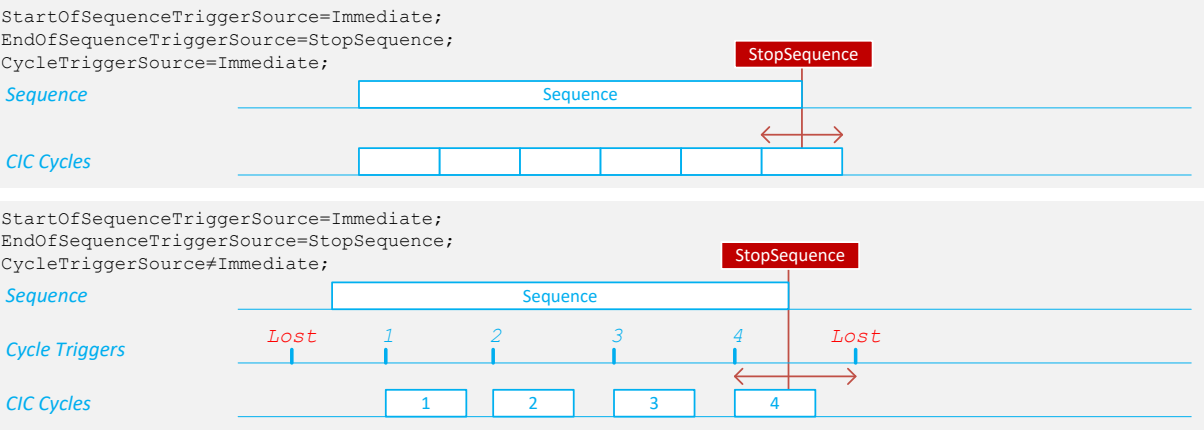
The first three cycles are "Readout Limited"; the fourth cycle is "Exposure Limited".

After having executed 4 cycles, the CIC controller falls back to the first timing definition.

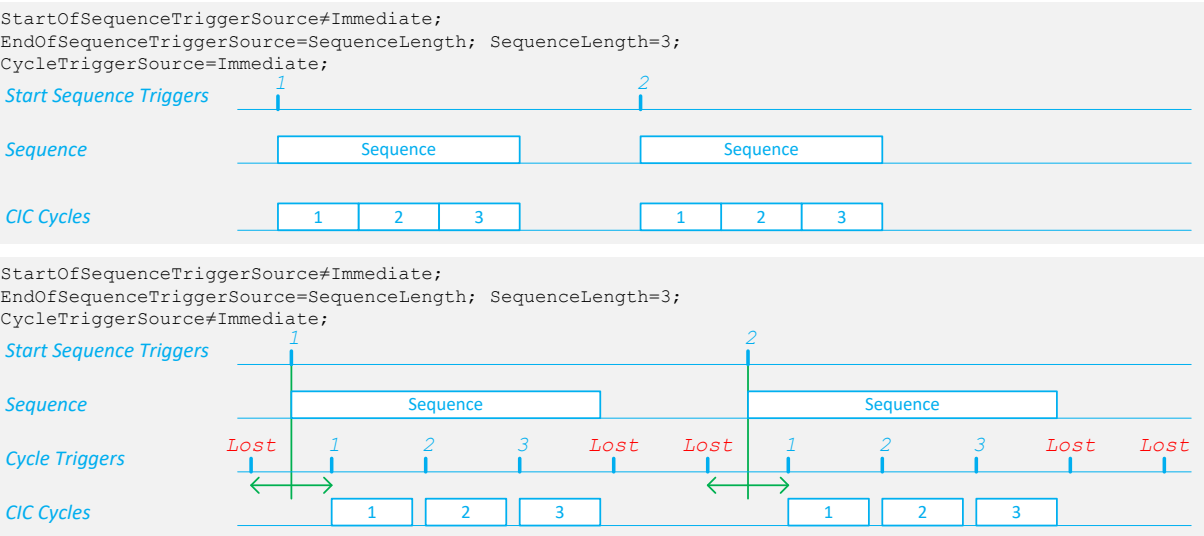
The sequence terminates slightly (*e1 time*) before the end of the fourth readout allowing the next sequence to begin with the optimal timing.

Cycle Sequence Timing Diagrams

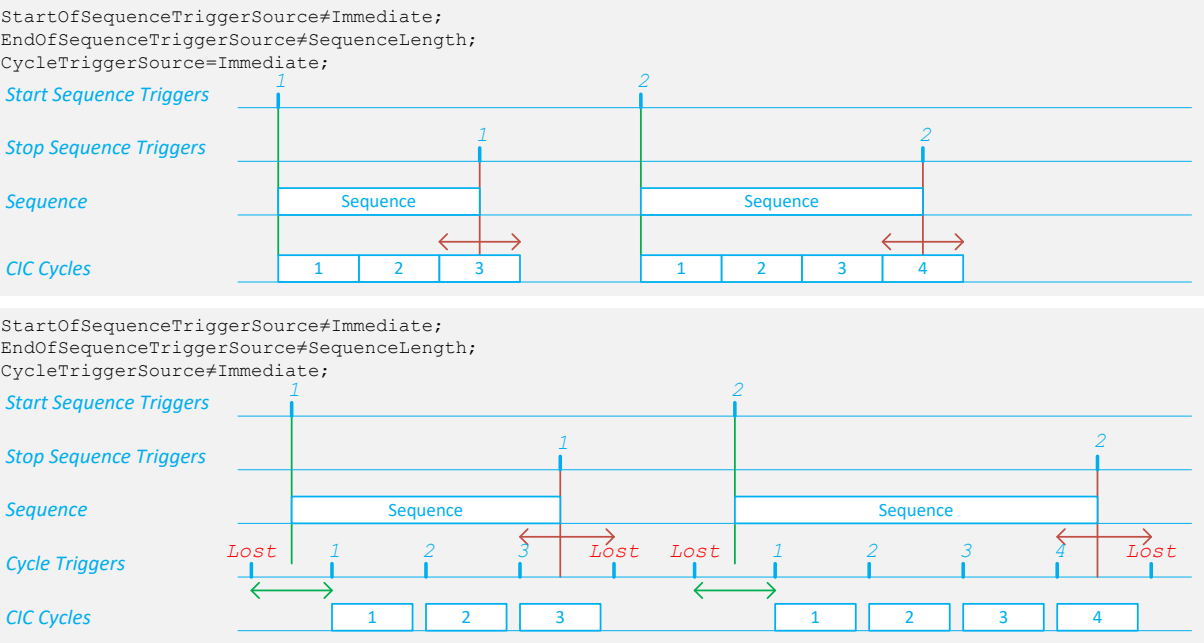
Timing diagrams of sequences of CIC cycles



Cycle sequences with immediate start



Cycle sequences with triggered start and fixed length



Cycle sequences with triggered start and triggered end

10. General Purpose I/O

10.1. I/O Lines Overview	221
10.2. I/O Lines Usage	224
10.3. I/O Control Blocks	225
10.4. Line Format and Line Mode Controls	227
10.5. Line Polarity Control	230
10.6. Line Filter Control	231
10.7. Line Source Selection	232
10.8. Line Source Divider	233
10.9. Logical I/O Line State	234
10.10. Physical I/O Line State	235
10.11. Line Driver Physical Output States	236
10.12. Initial States	237

10.1. I/O Lines Overview

General Purpose I/O lines overview in eGrabber-driven frame grabbers

GPIO lines in GenApi

The **LineSelector** feature of the **eGrabber-driven frame grabbers** Interface Modules exposes:

- 2 **Standard I/O sets** of **10** I/O lines named **DIN****, **IIN****, **IOUT**** and **TTLIO****
- 1 **Module I/O set** of **40** I/O lines named **MIO****

Standard I/O set

Each *Standard I/O Set* is composed of 10 I/O lines:

I/O Line Type	I/O Line Names	Count
Isolated input	IINx1, IINx2, IINx3, IINx4	4
Isolated output	IOUTx1, IOUTx2	2
RS-422 input	DINx1, DINx2	2
TTL I/O	TTLIOx1, TTLIOx2	2
Total		10

NOTE: x is the instance number: **1** for the first instance; **2** for the second instance.

Module I/O set

I/O Line Type	I/O Line Names	Count
See note (1)	MIO1 ... MIO40	40



NOTE

(1) The mix of I/O line types is defined by the attached extension.

GPIO lines per product

The number of effectively available GPIO lines is defined by the frame grabber configuration:

- A frame grabber includes 0, 1 or 2 "**Standard I/O set**" on page 221.
- A selection of frame grabbers with less than 2 "**Standard I/O set**" on page 221 accepts one extension module with an additional Standard I/O set.
- Frame grabbers with an I/O extension connector may accept one I/O Extension module.

The following table summarizes the capabilities:

Product	Standard I/O sets		Module I/O set
	#1	#2	
PC1633-T Coaxlink Quad G3	On-board	On-board	N/A
PC3602-T Coaxlink Octo	On-board	See note (2)	See note (3)
PC3621-LH-T Coaxlink Mono CXP-12-LH	On-board	See note (2)	See note (3)
PC3622-T Coaxlink Duo CXP-12	On-board	See note (2)	See note (3)
PC3623-T Coaxlink Quad CXP-12 Value	On-board	On-board	See note (3)

- (2) Attach one of the following I/O extension modules to the **I/O extension connector**:
 - **PC3614 HD26F I/O Extension Module - Standard I/O Set** to obtain the second "Standard I/O set" on page 221 instance.
 - **PC3618 HD26F I/O Extension Module - Fast I/O** to obtain the second "Standard I/O set" on page 221 instance with faster IIN** isolated input lines.

See also: 3618 I/O Extension Module in the hardware manual for more information.

- (3) Attach one of the following I/O extension modules to the **I/O extension connector**:
 - **PC3610 HD26F I/O Extension Module - TTL-RS422** to obtain up to a configurable mix of TTL and RS422 I/O lines: the "3610/3612 Module I/O set" on page 223.
 - **PC3612 HD26F I/O Extension Module - TTL-CMOS5V-RS422** to obtain up to a configurable mix of TTL/CMOS5V and RS422 I/O lines: the "3610/3612 Module I/O set" on page 223.

See also: 3610/3612 I/O Extension Modules in the hardware manual for more information.



WARNING

Only one module can be attached to the **I/O extension connector**!

GPIO lines electrical style

See also: "I/O Interfaces" on page 351 in the hardware manual for more information about the electrical styles

3610/3612 Module I/O set

Compatible with PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

Octo

Prelim

Mono12LH

Prelim

Duo12

Prelim

Value12

Prelim

PC3610 HD26F I/O Extension Module - TTL-RS422 and PC3612 HD26F I/O Extension Module - TTL-CMOS5V-RS422 provide the "3610/3612 Module" I/O set:

I/O Line Type	I/O Line Names	Count
RS-422 input	MIO1, MIO3...MIO19	0 to 10
RS-422 output	MIO1, MIO3...MIO19	0 to 10
TTL input	MIO1, MIO2...MIO20	0 to 20
TTL output (3610) CMOS output (3612)	MIO1, MIO2...MIO20	0 to 20
Total		10 RS-422 0 TTL/CMOS
		...
		0 RS-422 20 TTL/CMOS

10.2. I/O Lines Usage

Input lines

The input-capable I/O lines (DIN, TTLIO, IN, MIO) 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

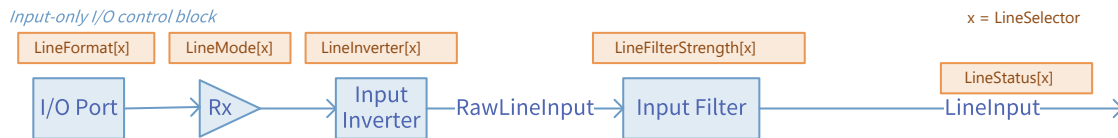
The output-capable I/O lines (TTLIO, IOUT, MIO) can be used as:

- 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 frame grabber and used to trigger the camera.

10.3. I/O Control Blocks

Every I/O line is controlled through an I/O control block. There are 3 types of control blocks:

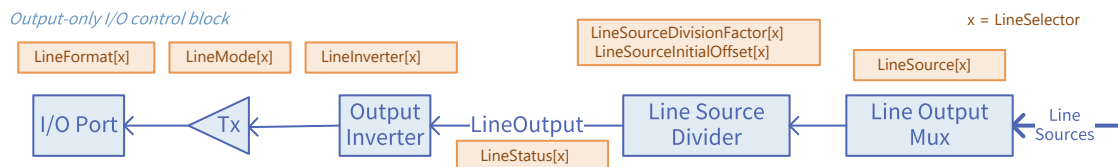
Input-only I/O control block



Input-only I/O control blocks and "Bidirectional I/O control block" on page 226 share a common *input path structure* composed with:

- The *I/O port* and *Rx* blocks represent the I/O pin on the **GPIO connector** and the line receiver circuit. The electrical style is defined by **LineFormat**: **ISO**, **DIFF** or **TTL**. The direction is defined by **LineMode**=**Input**. For details, refer to "Line Format and Line Mode Controls" on page 227.
- The *Input Inverter* block represents the user-configurable logic inverter. The inverter is controlled by **LineInverter**. For details, refer to "Line Polarity Control" on page 230.
- The *Input Filter* block represents the user-configurable glitch-removal filter. The filter is controlled by **LineFilterStrength**. For details, refer to "Line Filter Control" on page 231.
- **LineStatus** reports the logical state of the **LineInput** signal.

Output-only I/O control block

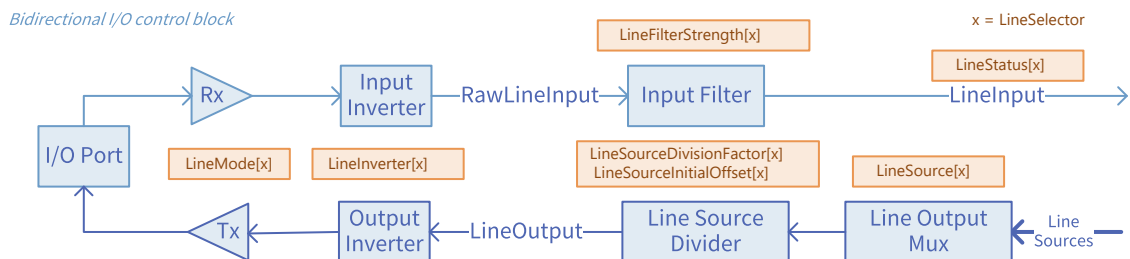


Output-only I/O control blocks and "Bidirectional I/O control block" on page 226 share a common *output path structure* including:

- The *I/O port* and *Tx* blocks represent the I/O pin on the **GPIO connector** and the line driver circuit. The electrical style is defined by **LineFormat**: **ISO**, **DIFF** or **TTL**. The direction and the driver configuration (if any) is defined by **LineMode**. For details, refer to "Line Format and Line Mode Controls" on page 227.
- The *Output Inverter* block represents the user-configurable logic inverter. The inverter is controlled by **LineInverter**. For details, refer to "Line Polarity Control" on page 230.
- The *Line Source Divider* block represents the user-configurable pulse train divider. The division is controlled by **LineSourceInitialOffset** and **LineSourceDivisionFactor**. For details, refer to "Line Source Divider" on page 233.

- The *Line Output Mux* block represents the user-configurable line source multiplexer. The multiplexer is controlled by **LineSource**. For details, refer to "[Line Source Selection](#)" on page 232.
- **LineStatus** reports the logical state of the **LineOutput** signal.

Bidirectional I/O control block



Bidirectional I/O control blocks implement both an input and an output paths. The input path structure is common with the "[Input-only I/O control block](#)" on page 225 and the output path structure is common with the "[Output-only I/O control block](#)" on page 225.

However, take care of the following facts:

- The **LineStatus** value reports the logical state of the **LineInput** signal. It is not possible to read back the logical state of the **LineOutput** signal!
- The **LineInverter** settings applies equally to the signal input path and the signal output path. It is not possible to have different polarity settings!

Legend

In the above figures:

- 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 (rectangular) shape represents a GenApi 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**)

10.4. Line Format and Line Mode Controls

Introduction

The following tables summarize 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**

Standard I/O sets

Standard I/O set #1

LineSelector	Bit#	LineFormat	LineMode
DIN11	0	DIFF	Input
DIN12	1	DIFF	Input
IIN11	4	ISO	Input
IIN12	5	ISO	Input
IIN13	6	ISO	Input
IIN14	7	ISO	Input
IOUT11	12	ISO	Output
IOUT12	13	ISO	Output
TTLIO11	16	TTL	Input, Output, DriveLow or DriveHigh
TTLIO12	17	TTL	Input, Output, DriveLow or DriveHigh

Standard I/O set #2

LineSelector	Bit#	LineFormat	LineMode
DIN21	1	DIFF	Input
DIN22	3	DIFF	Input
IIN21	8	ISO	Input
IIN22	9	ISO	Input
IIN23	1	ISO	Input
IIN24	11	ISO	Input
IOUT21	14	ISO	Output
IOUT22	15	ISO	Output
TTLIO21	18	TTL	Input, Output, DriveLow or DriveHigh
TTLIO22	19	TTL	Input, Output, DriveLow or DriveHigh

TTLIO ports mode control

The **LineMode** feature controls the direction and the line driver mode of each individual TTLIO port. Four modes can be selected at any time:

- **Input**: input only, totem-pole driver disabled (default power-up settings),
- **Output**: totem-pole driver capable of driving low and high,
- **DriveLow**: open-collector driver capable of driving low only,
- **Drivehigh**: open-emitter driver capable of driving high only.



NOTE

The two latest configurations allow wired-AND configurations. The line state can be read back through the input port.

I/O modules

3610/3612 I/O Extension modules

LineSelector	Bit#	LineFormat	LineMode
MIO1	20	TTL or DIFF	Input or Output
MIO2	21	TTL	Input or Output
:	:	:	:
MIO19	38	TTL or DIFF	Input or Output
MIO20	39	TTL	Input or Output

MIO format and mode controls

The **LineFormat** feature controls the electrical style of the MIO ports. Possible values:

- **TTL**: single-ended (TTL or CMOS)
- **DIFF**: differential (RS422)

The **LineMode** feature controls the direction of the MIO ports. Possible values::

- **Input**: input only,
- **Output**: totem-pole driver capable of driving low an high.



NOTE

The controls can only be changed during the module configuration .

See also: [3610/3612 I/O Extension Modules](#) for an extensive description of the configuration.

10.5. 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 Inputs* signals of the input path and the *Line Sources* signals of the output path are always using positive logic.

10.6. Line 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:

- When set to **Custom**, the filter time constant is configurable by setting **LineFilterDelay** to the desired value.
- When set to another value (**Lowest**, ... **Highest**):
 - the filter time constant is preset according to the selected strength and the I/O input line type (as shown hereafter)
 - the actual filter time constant is reported by **LineFilterDelay**

The default settings is **Lowest**.

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



TIP

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



NOTE

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.

10.7. 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 line source

To select an internal signal to feed the line driver of an output capable I/O line:

1. Select an I/O line by assigning the appropriate value to the **LineSelector** feature
2. Assign the appropriate value to the **LineSource** feature:

LineSource feature value	Source signal
UserOutput<0:7>	Any bit of the User Output register
Device<0:7>Strobe	Strobe output of any device
Device<0:7>CameraTrigger	Camera trigger of any device
Device<0:7>Stream<0:7>StartOfCameraReadout	StartOfCameraReadout event* ¹ on any stream of any device
CustomLogicOutput<0:31>	Any bit of the custom_logic_output_ctrl register* ² (CustomLogic General Purpose I/O Interface)
Low	Steady low
High	Steady high



NOTE

1. The StartOfCameraReadout event corresponds to a pulse of 344 ns.
2. The custom_logic_output_ctrl register is only available in CustomLogic variants.

10.8. Line Source Divider

Output ports pulse train divider

The signal output path of "Output-only I/O control block" on page 225 and "Bidirectional I/O control block" on page 226 includes a *Line Source Divider* block.

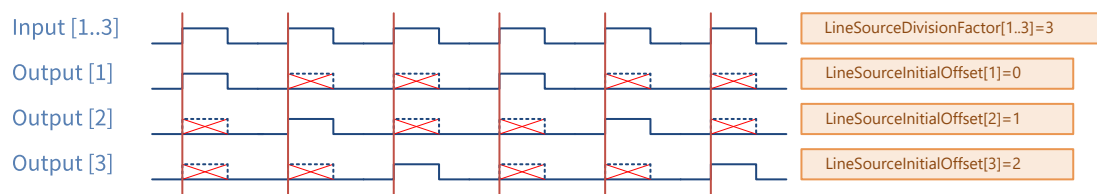
The *Line Source Divider* decimates the incoming pulse train, keeping only 1 pulse every N pulses. N is user-configurable by setting **LineSourceDivisionFactor** to any value in the [1..8] range. The default value is 1 (no decimation).

The decimation happens on the rising edge of the incoming pulse. This is not configurable.

The decimation can be offset by up to 7 pulses by setting **LineSourceInitialOffset** to any value in the [0..7] range. The default value is 0 (no offset).

Typical use case

The *Line Source Divider* of 3 outputs is configured to decimate the same input signal by 3 with respective offsets of 0, 1 and 2.



Decimation by 3 with respective offsets of 0, 1 and 2

GenApi settings example

The pulse train of the **Strobe** signal of Device0 is routed to 3 outputs: TTLIO11, TTLIO12 and TTLIO21 and, then, decimated by 3 with respective offsets of 0, 1 and 2.

```
LineMode[TTLIO11] = Output
LineSource[TTLIO11] = Device0Strobe0
LineSourceDivisionFactor[TTLIO11] = 3
LineSourceInitialOffset[TTLIO11] = 0
```

```
LineMode[TTLIO21] = Output
LineSource[TTLIO21] = Device0Strobe0
LineSourceDivisionFactor[TTLIO21] = 3
LineSourceInitialOffset[TTLIO21] = 2
```

```
LineMode[TTLIO12] = Output
LineSource[TTLIO12] = Device0Strobe0
LineSourceDivisionFactor[TTLIO12] = 3
LineSourceInitialOffset[TTLIO12] = 1
```

10.9. 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

1. *Step 1:* Select an I/O line by assigning the appropriate value to the **LineSelector** feature
2. *Step 2:* 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.

10.10. 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. <i>Line may be left unconnected or connected with the wrong polarity.</i>
		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.
		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.

10.11. 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 CMOS	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.

10.12. 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**.

11. I/O Toolbox

11.1. Introducing the I/O Toolbox	239
11.2. I/O Toolbox Composition Tables	241
11.3. Line Input Tool	244
11.4. Quadrature Decoder Tool	245
11.5. Device Link Trigger Tool	248
11.6. User Actions Tool	249
11.7. C2C-Link Synchronization Tool	254
11.8. Delay Tool	256
11.9. Divider Tool	258
11.10. Multiplier/Divider Tool	259

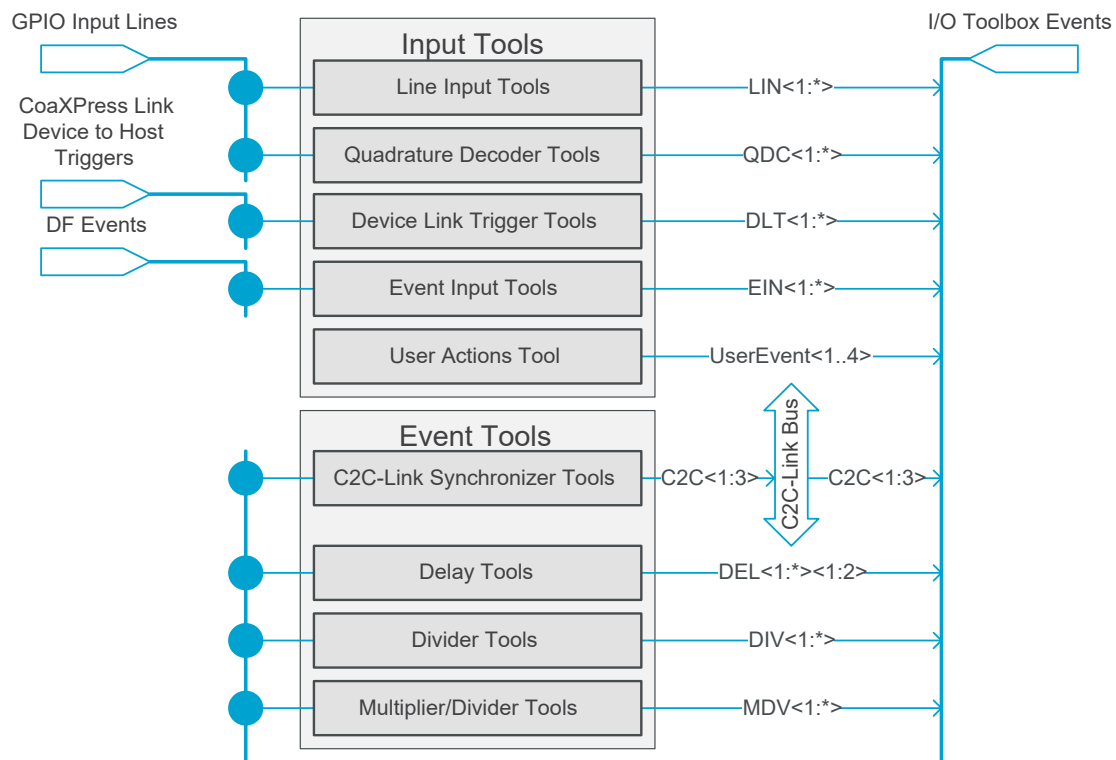
11.1. Introducing the I/O Toolbox

The *I/O Toolbox* is a configurable array of interconnected tools belonging to the GenTL Interface module.

Each tool generates one (or more) event stream. The stream name is composed of a prefix designating the tool (LIN, QDC, DLT...) followed by a 1-based index.

The *I/O Toolbox Events stream* are shared by various consumers on the card:

- The Camera and Illumination controller of any device belonging to the card for camera cycle triggering
- The Acquisition controller of any device belonging to the card for starting or stopping line-scan acquisition sequences
- Event counters



I/O Toolbox structure diagram

I/O Toolbox input tools

The *Input Tools* generate event streams from external sources.

1. The "[Line Input Tool](#)" on [page 244](#) for use with sensors and detectors attached to a single GPIO input line.
2. The "[Quadrature Decoder Tool](#)" on [page 245](#) for use with quadrature motion encoders attached to a pair of GPIO input lines.
3. The "[Device Link Trigger Tool](#)" on [page 248](#) for use with triggers issued by CoaXPress 2.0 cameras.
4. The [Event Input Tool](#) for use in line-scan data-forwarding applications.
5. The "[User Actions Tool](#)" on [page 249](#) for use by the application software to generate user events.

A fully populated interconnection matrix allows:

- any *Line Input Tool* to be fed by any GPIO input line.
- any *Quadrature Decoder Tool* to be fed by a selection of GPIO input line pairs.
- any *Device Link Trigger Tool* to be fed by any CoaXPress Link Device to Host Trigger of any camera attached to the card.

I/O Toolbox event tools

The *Event Tools* process internal events generated by any *I/O Toolbox tool*

1. The "[C2C-Link Synchronization Tool](#)" on [page 254](#) delivers one event stream to the C2C-Link Bus driver.
2. The "[Delay Tool](#)" on [page 256](#) delays the events of one (or two) stream(s) by a configurable number of clock tick events.
3. The "[Divider Tool](#)" on [page 258](#) divides the event rate by an integer factor D.
4. The "[Multiplier/Divider Tool](#)" on [page 259](#) converts the event rate by a rational factor M/D.

A fully populated interconnection matrix allows any *Event Tool* to be fed by any *I/O Toolbox event stream*

I/O Toolbox tools cascading

Tools can be cascaded to form a tool chain:

- A tool chain always begins with an Input Tool.
- A tool may drive 0, 1 or several Event Tools.

11.2. I/O Toolbox Composition Tables

Introduction

Each of the following tables show the composition of the I/O toolbox for all variants of the designated product.

Column Name	Description
Input Tools	I/O Toolbox input tools: - #LIN: Number of line input tools - #QDC: Number of quadrature decoder tools - #DLT: Number of device link trigger tools - #EIN: Number of event input tools - #UAS: Number of user actions scheduler tools
Event Tools	I/O Toolbox event tools: - #DEL: Number of delay tools - #DIV: Number of divider tools - #MDV: Number of multiplier/divider tools - #C2C: Number of C2C-Link synchronization tools



NOTE

An empty cell indicates that the corresponding tool type is not available.

PC1633-T Coaxlink Quad G3

Firmware Variant	Input tools					Event tools			
	#LIN	#QDC	#DLT	#EIN	#UAS	#DEL	#DIV	#MDV	#C2C
1-camera	8	1			1	2	1	1	2
1-camera, 4-data-stream	8	1			1	2	1	1	2
1-camera, line-scan	8	1			1	2	1	1	3
1-slm-camera	8	1			1	2	1	1	2
1-sls-camera	8	1			1	2	1	1	3
2-camera	8	2			1	2	2	2	2
2-camera, bayer	8	1			1				
2-camera, line-scan	8	2			1	2	2	2	3
3-camera	8	2			1	2	2	2	2
4-camera	8	4			1	4	4	4	2
4-camera, line-scan	8	4			1	4	4	4	3

PC3602-T Coaxlink Octo

Firmware Variant	Input tools					Event tools			
	#LIN	#QDC	#DLT	#EIN	#UAS	#DEL	#DIV	#MDV	#C2C
1-camera	8	1	2		1	2	1	1	2
1-camera, custom-logic	8	1	2		1	2	1	1	2
1-camera, line-scan	8	1	2		1	2	1	1	3
2-camera	8	1	4		1	2	1	1	2
2-camera, line-scan	8	2	4		2	2	2	2	3
2-camera, line-scan, custom-logic	8	2	4		1	2	2	2	3
3-camera	8	1	6		1	2	1	1	2
4-camera	8	1	8		1	2	1	1	2
4-camera, line-scan	8	4	8		1	4	4	4	3
5-camera	8	1	10		1	2	1	1	2
5-camera, 5D22211	8	1	10		1	2	1	1	2
6-camera	8	1	12		1	2	1	1	2
8-camera	8	1	16		1	2	1	1	2
8-camera, line-scan	8	4	16		1	4	4	4	3

PC3621-LH-T Coaxlink Mono CXP-12-LH

Firmware Variant	Input tools					Event tools			
	#LIN	#QDC	#DLT	#EIN	#UAS	#DEL	#DIV	#MDV	#C2C
1-camera	8	1	2		1	2	1	1	2
1-camera, line-scan	8	1	2		1	2	1	1	3

PC3622-T Coaxlink Duo CXP-12

Firmware Variant	Input tools					Event tools			
	#LIN	#QDC	#DLT	#EIN	#UAS	#DEL	#DIV	#MDV	#C2C
1-camera	8	1	2		1	2	1	1	2
1-camera, line-scan	8	1	2		1	2	1	1	3
2-camera	8	2	4		1	2	2	2	2
2-camera, line-scan	8	2	4		1	2	2	2	3

PC3623-T Coaxlink Quad CXP-12 Value

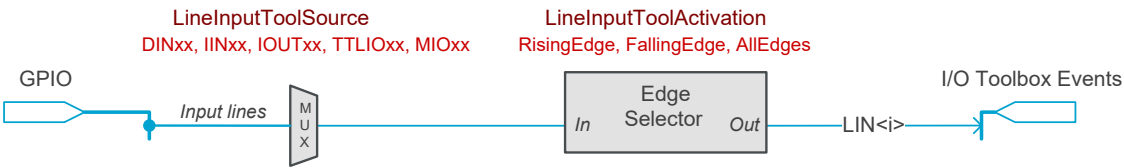
Firmware Variant	Input tools					Event tools			
	#LIN	#QDC	#DLT	#EIN	#UAS	#DEL	#DIV	#MDV	#C2C
1-camera	8	1	2		1	2	1	1	2
1-camera, line-scan	8	1	2		1	2	1	1	3
2-camera	8	2	4		1	2	2	2	2
2-camera, line-scan	8	2	4		1	2	2	2	3
3-camera	8	3	6		1	3	3	3	2
4-camera	8	4	8		1	4	4	4	2
4-camera, line-scan	8	4	8		1	4	4	4	3

11.3. Line Input Tool

Applies to all firmware variants of eGrabber-driven frame grabbers

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

Diagram



LIN tool functional and wiring diagram

Any input-capable GPIO line can be selected as the input source.
The tool feeds one I/O Toolbox event stream named LIN<i>.

Operation

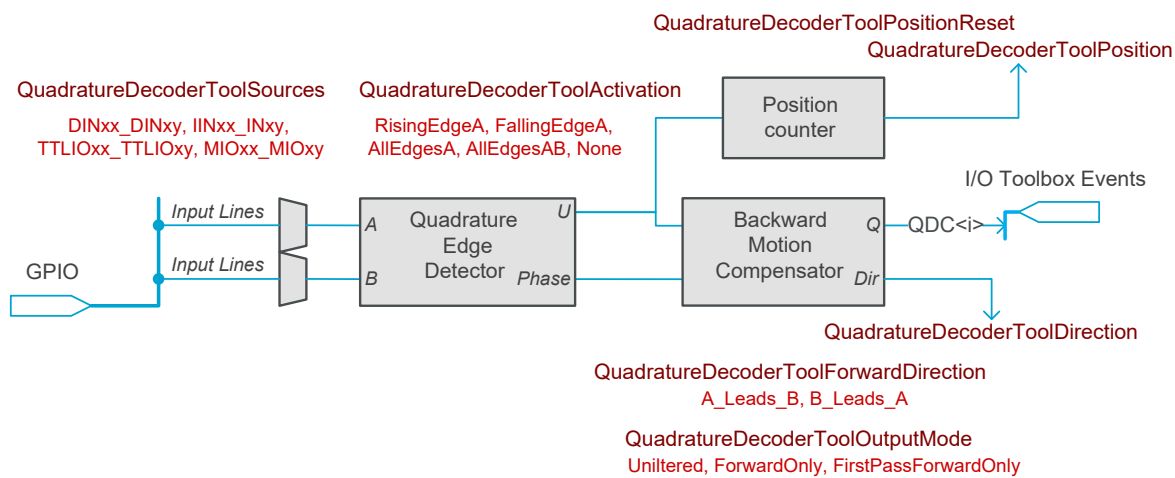
The Line Input tool detects the rising or the falling edge of the LineInput signal delivered by the I/O Control block selected by LineInputToolSource.
The Line Input tool delivers one event at every rising or falling edge or both according to the LineInputToolActivation settings.

11.4. Quadrature Decoder Tool

Applies to all firmware variants of eGrabber-driven frame grabbers

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

Diagram



QDC tool functional and wiring diagram

The quadrature edge detector is fed by a pair of signals named A and B delivered by a *phase-quadrature motion encoder* device.

The source selectors allows selected pairs of adjacent input-capable GPIO input lines to be selected as A/B input sources.

See also: "QuadratureDecoderToolSources" on page 1

The tool includes the following function blocks:

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

The tool delivers:

- One I/O Toolbox event stream named QDC<i>.
- A *direction* status bit indicating the direction of the motion.
- A *position* status word indicating the position offset.

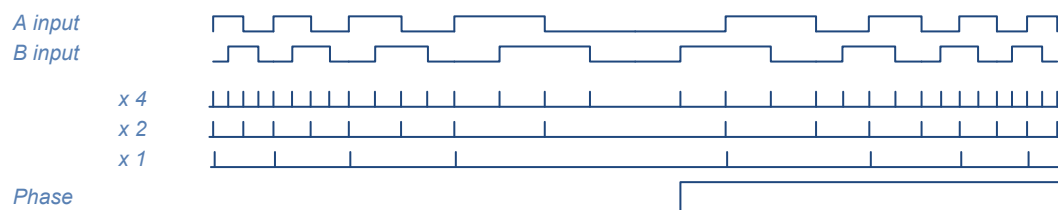
Operation

The Quadrature Decoder Tool decodes the A/B signals and delivers 1, 2, or 4 events every A/B cycle, possibly filtered by the backward motion compensator.

Quadrature edge detector

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 to the `QuadratureDecoderToolActivationsettings`
- An identification of the phase between A and B (A leads B or vice-versa)



Quadrature edge detector waveforms

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

The phase indication may be inverted according to the `QuadratureDecoderToolForwardDirection` settings before being delivered to the *Direction* output.

Backward motion compensator

The *backward motion compensator* (BMC) filters the U stream according to the `QuadratureDecoderToolOutputModesetting`.

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



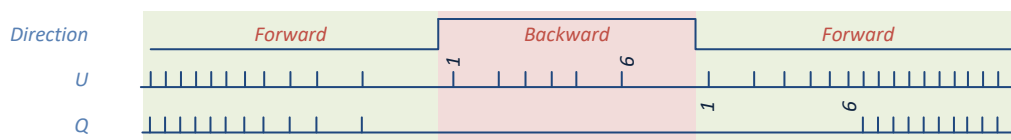
BMC waveforms - Unfiltered

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



BMC waveforms - Forward Only

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



BMC waveforms - First Pass Forward Only

Position Counter

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.

11.5. Device Link Trigger Tool

Applies to the following firmware variants of **PC3602-T Coaxlink Octo**, **PC3621-LH-T Coaxlink Mono CXP-12-LH**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12 Value**:

Octo ^{Prelim} (1-camera), (1-camera, custom-logic), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (2-camera, line-scan, custom-logic), (3-camera), (4-camera), (4-camera, line-scan), (5-camera), (5-camera, 5D22211), (6-camera), (8-camera), (8-camera, line-scan)

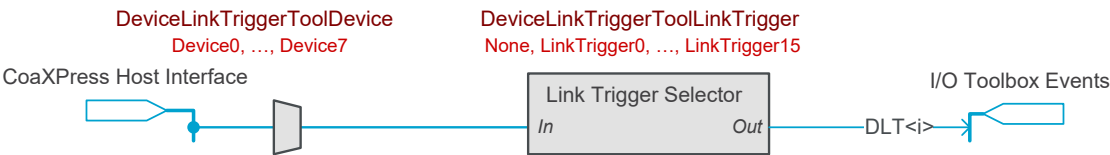
Mono12LH ^{Prelim} (1-camera), (1-camera, line-scan)

Duo12 ^{Prelim} (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan)

Value12 ^{Prelim} (1-camera), (1-camera, line-scan), (2-camera), (2-camera, line-scan), (3-camera), (4-camera), (4-camera, line-scan)

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Device Link Trigger Tool	DLT	1 Event	1 event stream: DLT<i>

Diagram



DLT tool functional and wiring diagram

Any remote device attached to the CoaXPress Host interface can be selected as the source device for CoaXPress 2.0 Device-to-Host triggers.

Any of the 16 LineTrigger of the selected remote device can be selected as the input source.

The tool feeds one I/O Toolbox event stream named DLT<i>.

Operation

The Device Link Trigger tool generates one DLT event when a valid high-speed connection trigger packet message corresponding to the **DeviceLinkTriggerToolLinkTrigger** is received from a CoaXPress 2.0 remote device designated by **DeviceLinkTriggerToolDevice**.

11.6. User Actions Tool

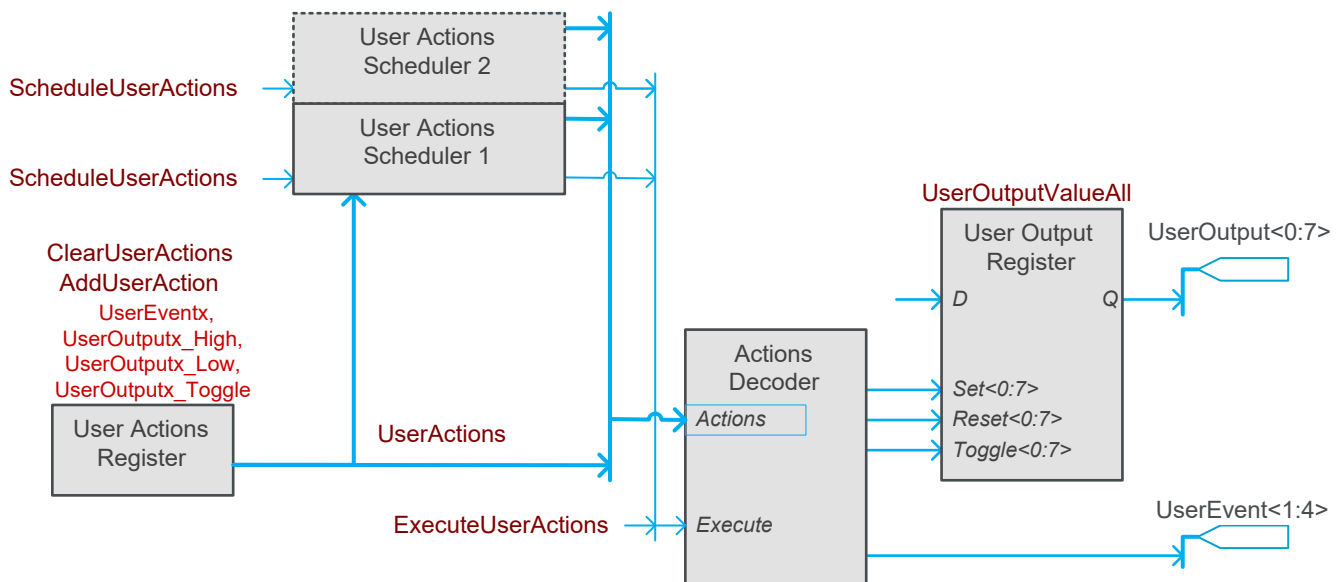
Applies to all firmware variants of **eGrabber-driven frame grabbers**

The *User Actions Tool* allow the application software to perform the following *User Actions*:

- Setting high, setting low or toggling any bit of the **UserOutputRegister**,
- Generate any one or more **UserEvent**.

To generate actions, the user application has to proceed in two steps:

1. Define a set of one or more User Actions in the **UserActionsRegister**.
2. Schedule the execution:
 - a. For an *Immediate* execution, use the **ExecuteUserActions** GenApi feature.
 - b. For a *Deferred* execution, use the **ScheduleUserActions** GenApi feature of any available **UserActionsScheduler**.



User Actions Tool functional block diagram

User output register

eGrabber-driven frame grabbers provide an 8-bit¹ *User Output Register* where bits are named UserOutput0 ... UserOutput7.

Any *User Output Register* bit can drive any one or more output-capable GPIO lines.

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 *User Output Register* bits.

User events

eGrabber-driven frame grabbers provide a generator for 4 user-defined events named UserEvent1 ... UserEvent4.

User-defined events can be used by various consumers:

- To trigger a camera cycle using **CycleTriggerSource** feature
- To trigger the start or the end of an acquisition sequence using **StartOfSequenceTriggerSource** and **EndOfSequenceTriggerSource** features
- To trigger the start or the end of a line-scan acquisition using **StartOfScanTriggerSource** and **EndOfScanTriggerSource** features
- As an event source for the Divider, Multiplier/Divider and the Delay tool using **DividerToolSource**, **DelayToolSource<1:2>** and **MultiplierDividerToolSource** features
- As trigger source on the C2C-Link using the **C2CLinkSynchronizationToolSource** feature
- As notification context using **EventNotificationContext<1:3>** features

User actions register

The User Actions Register is a 32-bit register that defines a set of User Actions that will be executed simultaneously.

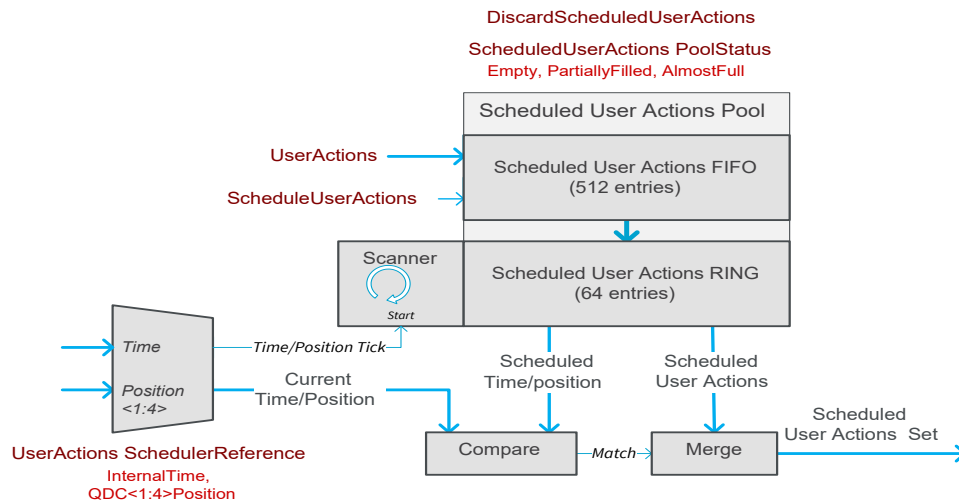
The **ClearUserActions** feature allows to clear the register.

The **AddUserActions** feature allows the application to compose the actions set one by one:

- Assert a user event using the **UserEvent<1:4>** values
- Set any user output bit high using **UserOutput<0:7>_High**
- Set any user output bit low using **UserOutput<0:7>_Low**
- Toggle user output bit high using **UserOutput<0:7>_Toggle**

¹ **PC1630 Coaxlink Mono** implements only the 4 lowest bits!

User actions scheduler



User actions scheduler functional diagram

The *User Actions Scheduler* (UAS) function block allows an application software to postpone the execution of the actions at predefined time or position.

Prior to schedule any user actions, the user application has to setup the UAS:

1. Define the **SchedulerReference**
2. Initialize the **ScheduledActionsPool**

Scheduler reference

The scheduler reference is a 32-bit value that can be a *time* or a *position*. It is defined by setting **UserActionsSchedulerReference** as follows:

- **InternalTime** selects the *frame grabber 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.
- **QDC<1:4>Position** select the *Position Counter* of the Quadrature Decoder tools QDC<1:4> respectively. In that case,



NOTE

For correct operation of the UAS with a position reference:

- The position counter must increment monotonically and not faster than every microsecond.
- 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 `QuadratureDecoderToolOutputMode` to `ForwardOnly` or to `FirstPassForwardOnly`.

Scheduled user actions pool

The *Scheduled User Actions Pool* is a memory area where the *Scheduled User Actions Sets* are stored by the user application.

The pool is sub-divided into two sections:

1. A 512-locations FIFO
2. A 64-locations RING

FIFO operation

New Scheduled User Actions are first written to the FIFO before being automatically transferred to the RING when it contains at least one free location.

The first (oldest) entry is transferred first. The entries are not reordered!

The `ScheduledUserActionsPoolStatus` reports an `AlmostFull` value when the FIFO is almost full and is unable to accept a new entry.

Adding new Scheduled User Actions

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` GenApi 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` GenApi feature.

Removing Scheduled User Actions

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.

NOTE: It is not possible to remove a specific entry!

RING operation

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.

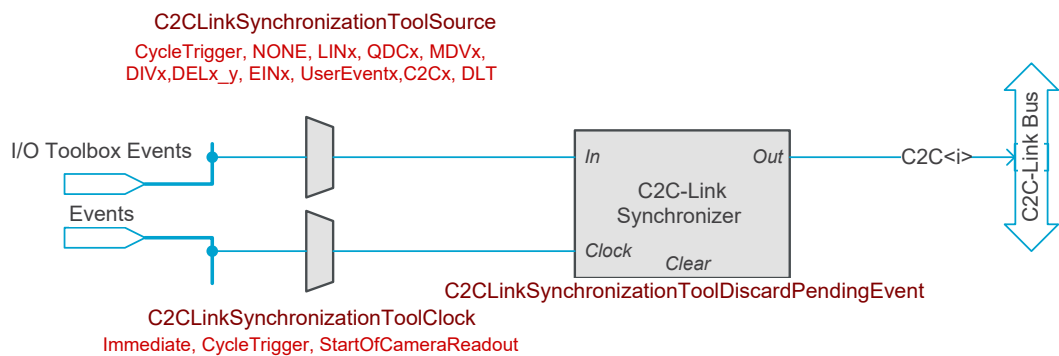
At the end of the scan, the merged elected actions are executed simultaneously. The time delay from the reference tick up to the execution of the elected actions is very small (sub-microsecond) and constant.

11.7. C2C-Link Synchronization Tool

Applies to all firmware variants of eGrabber-driven frame grabbers

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
C2C-Link Synchronization Tool	C2C	1 or 2 event streams	1 event stream: C2C<i>

Diagram



C2C tool functional and wiring diagram

The C2C-Link Synchronization tool (C2C) tool delivers one event stream to the C2C-Link Bus driver. It includes the following blocks:

- A source selector
- A clock source selector
- An event synchronizer with clear control

Operation

Source selector

The source selector selects the event stream applied to the tool input (In). It provides following options:

- On C2C1 instance only: `Cycle Trigger` event stream driven by the Camera and Illumination controller.
- On C2C2 and C2C3 instances only: any I/O toolbox event.

Synchronizer control

The clock source selector controls the event stream synchronization:

- When `C2CLinkSynchronizationToolClock` is set to `Immediate`, the event stream applied to the input (In) is sent immediately to the output.
- On C2C2 and C2C3 instances only: when `C2CLinkSynchronizationToolClock` is set to `CycleTrigger`, the event is latched and delayed until the following `Cycle Trigger` event.
- On C2C2 and C2C3 instances only: when `C2CLinkSynchronizationToolClock` is set to `StartOfCameraReadout`, the event is latched and delayed until the following `Start of Camera Readout` event.

The `C2CLinkSynchronizationToolDiscardPendingEvent` command discards an event that has been received but that has not been forwarded.



NOTE

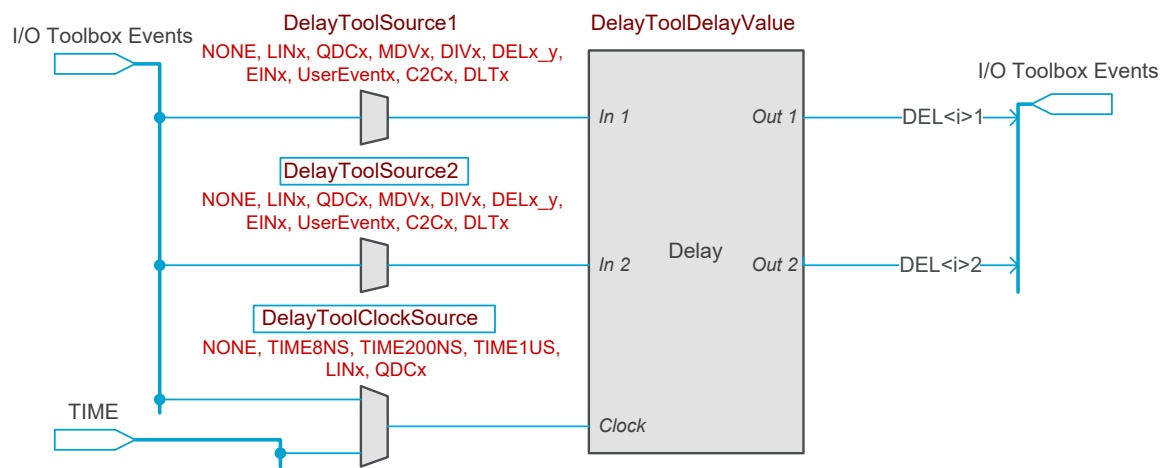
Area-scan firmware variants provide 2 instances of the C2C tool; line-scan firmware variants provide 3 instances!

11.8. Delay Tool

Applies to all firmware variants of eGrabber-driven frame grabbers

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

Diagram



DEL tool functional and 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.

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 sources are selected by **DelayToolSource1** and **DelayToolSource2** respectively.

The same delay applies to both channels. The common delay is defined by **DelayToolDelayValue**.

The same clock source applies to both channels. The clock source is defined by **DelayToolClockSource**. It can be a *time base*, a *Line Tool event stream*, or a *Quadrature Decoder Tool event stream*.

Selecting a *time base* implements a time delay function. The available time bases are:

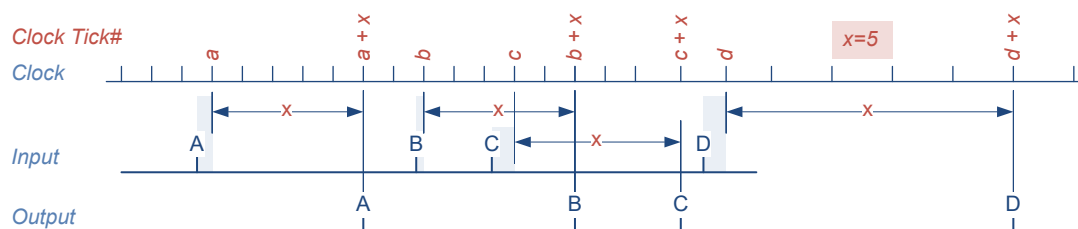
- **8NS**: A 125 MHz high accuracy regular time base allowing delays from *40 nanoseconds* up to *134 milliseconds* by steps of 8 nanoseconds
- **200NS**: A 5 MHz high accuracy regular time base allowing delays from *200 nanoseconds* up to *3.35 seconds* by steps of 200 nanoseconds
- **1US**: A 1 MHz high accuracy regular time base allowing delays from *1 microsecond* up to *16.7 seconds* by steps of 1 microsecond

Selecting a *line tool event stream* implements a position offset function when the line tool is fed by a motion encoder device. Any available Line Input tool or Quadrature Decoder tool can be used as delay clock source. The delay range is *1* up to *16,777,215 events*.



WARNING

The Delay tool operates as a delay line. The tool may accept a new event while the previous one is not yet delivered! The Delay tool is capable of recording, globally for all channels, up to 16 distinct events.



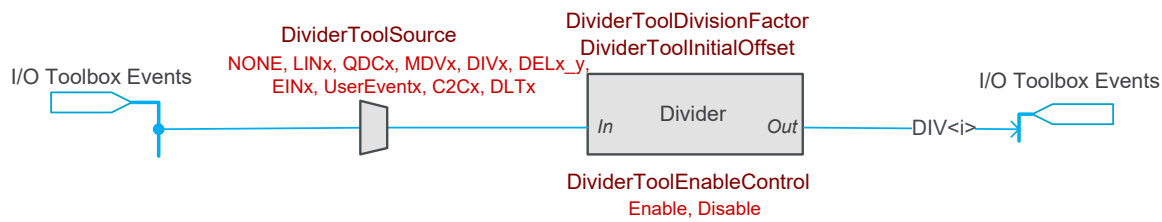
DEL tool waveforms

11.9. Divider Tool

Applies to all firmware variants of eGrabber-driven frame grabbers

Tool Name	Short Name	Inputs Count/Type	Outputs Count/Type/Name
Divider Tool	DIV	1 Toolbox event stream	1 event stream: DIV<i>

Diagram

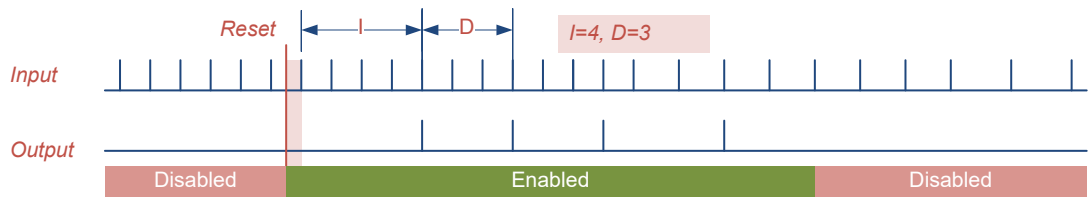


DIV tool functional and wiring diagram

Any I/O Toolbox event stream can be selected as the input source.

The tool feeds one I/O Toolbox event stream named DIV<i>.

Operation



DIV tool waveforms

Once enabled, the Divider tool skips the first – I – input events before delivering an event every D input events.

The division factor – D – is defined by DividerToolDivisionFactor. The default value is 2 and the value range is 1 … 65535.

The initial offset – I – is defined by DividerToolInitialOffset. The default value is 0 and the value range is 0 … 65535.

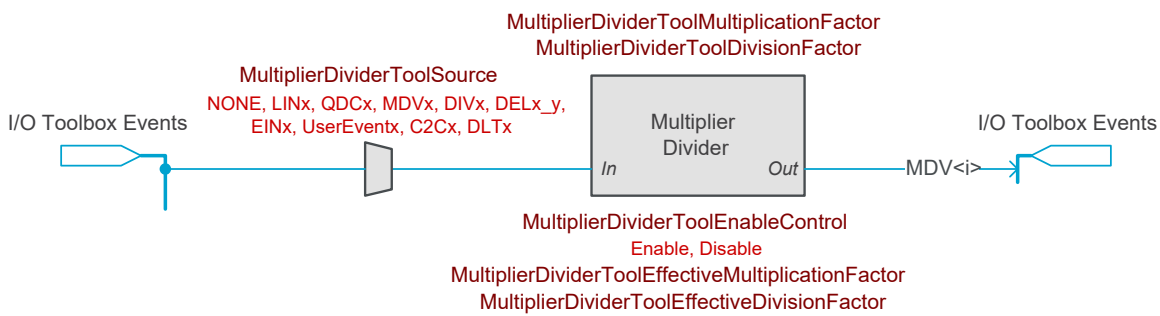
The operation state is defined by DividerToolEnableControl. The default value is Disable.

11.10. Multiplier/Divider Tool

Applies to all firmware variants of eGrabber-driven frame grabbers

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>

Diagram



MDV tool functional and wiring diagram

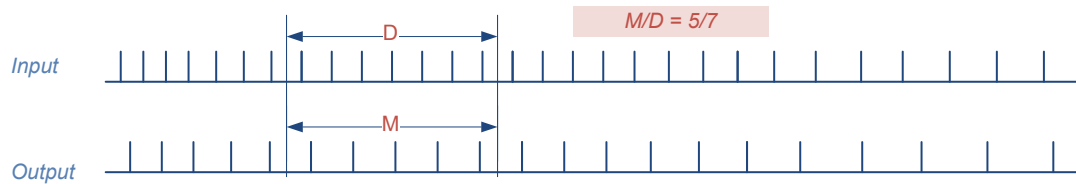
Any I/O Toolbox event stream can be selected as the input source.
The tool feeds one I/O Toolbox event stream named MDV<i>.

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.



MDV tool waveforms

The Rate Conversion Ratio is configured as the ratio of two float numbers:

- The *M* value is defined by `MultiplierDividerToolMultiplicationFactor`. The default value is 1.0 and the value range is 0.001 to 1000.0.
- The *D* value is defined by `MultiplierDividerToolDivisionFactor`. The default value is 1.0 and the value range is 0.001 to 1000.0.

The effective multiplication and division factors are respectively reported by `MultiplierDividerToolMultiplicationFactor` and `MultiplierDividerToolDivisionFactor`.



NOTE

The effective values may slightly differ from the specified values. However, the RCR relative error remains negligible (less than 1/1000).

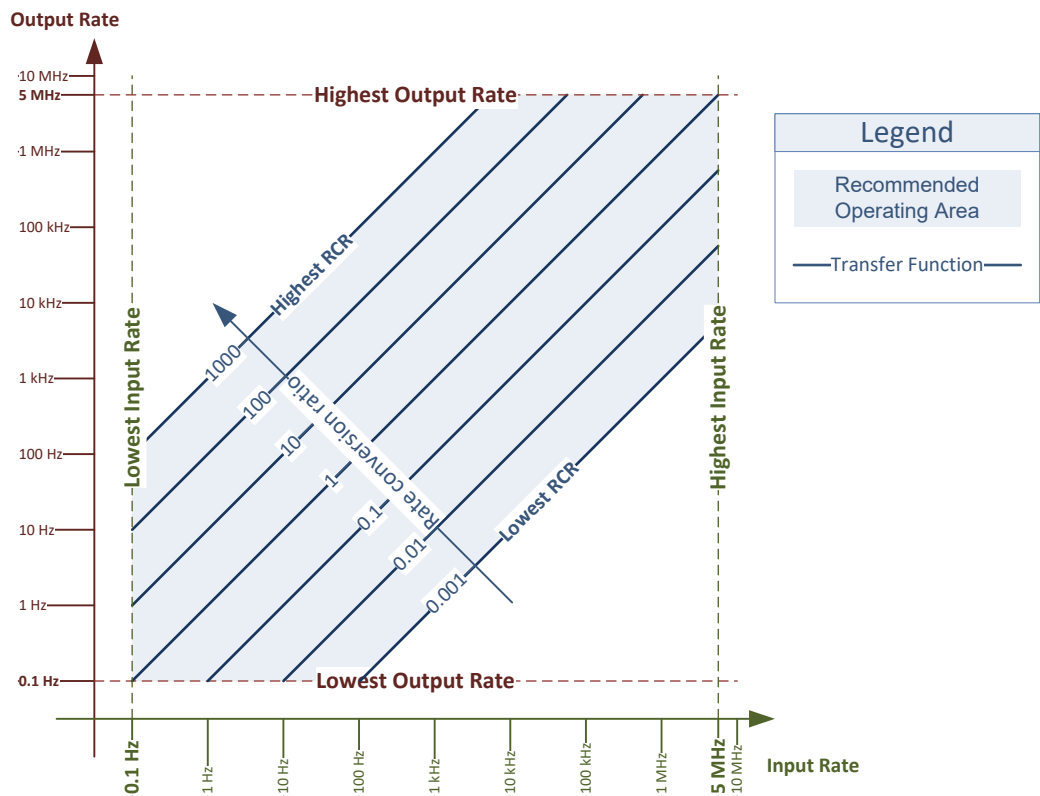


NOTE

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 accumulated phase error increases when the input frequency increases. It decreases when the input frequency decreases.
The Multiplier/Divider is *not phase accurate*.

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



MDV tool operating limits diagram

12. Event Signaling And Counting

Extensive user-configurable event-reporting and event-counting mechanism

12.1. Introduction	263
12.2. Custom Event Sources	266
EVENT_CUSTOM_CIC	267
EVENT_CUSTOM_CXP_DEVICE	268
EVENT_CUSTOM_CXP_INTERFACE	268
EVENT_CUSTOM_DATASTREAM	270
EVENT_CUSTOM_DEVICE_ERROR	271
EVENT_CUSTOM_IO_TOOLBOX	272
12.3. Event Specific Context Data	274
12.4. About GenTL Signaling	276

12.1. Introduction

Short description

eGrabber-driven frame grabbers feature a powerful event management that allows the application to be notified of the occurrence of various events.

In addition to the GenTL EVENT_NEW_BUFFER and EVENT_REMOTE_DEVICE standard event types, the Coaxlink, Grablink and Gigelink GenTL producers 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 be enabled or disabled.
- The content of each user-defined context data.

Event data are temporarily stored in the Event Queue Buffer. The Coaxlink GenTL producer is notified, using an interruption mechanism, of the availability of one or more event entries in the Event Queue Buffer.

The Coaxlink, Grablink and Gigelink GenTL producers implement 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.

Event types

GenTL identifies the events according to their Type and their relevant object Module.

Standard event types

The Coaxlink, Grablink and Gigelink GenTL producers implement 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.
EVENT_REMOTE_DEVICE	Device	Notification if the GenTL Producer library wants to manually set a feature in the GenICam GenApi instance using the module.

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.

The Coaxlink GenTL producer implements the following custom event types for registration by the GenTL Consumer application:

Event Type	Module	Description
EVENT_CUSTOM_CIC	Device	Notification of Camera and Illumination Control events
EVENT_CUSTOM_CXP_DEVICE	Device	Notification of CoaXPress device events
EVENT_CUSTOM_CXP_INTERFACE	Interface (Device)	Notification of CoaXPress Host Interface events
EVENT_CUSTOM_DATASTREAM	Data Stream	Notification of CoaXPress data stream events
EVENT_CUSTOM_DEVICE_ERROR	Device	Notification of device error events
EVENT_CUSTOM_IO_TOOLBOX	Interface (Device)	Notification of I/O Toolbox 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.

See also: "Custom Event Sources" on page 266 for an exhaustive list.

Custom events counter

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:

- Read the count value of a selected event source.
- Reset the counter of a selected event source.
- Reset the counters of all the event sources of the module.

The count-value can also be used as user-defined context data by any event source.

Custom events configuration

The event source is configurable.

At any time, the user application can:

- Enable or disable the notification of a selected event source.
- Enable or disable the notification of all the event sources of the module.
- Define the content of each user-defined context data of a selected event source.

Notification

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.

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 I/O lines sampled at the event occurrence time
- Count value of any event counter.
- Count value of any Quadrature Decoder (QDC) position counter.

Some event sources provide additional options.

See also: "Event Specific Context Data" on page 274

12.2. Custom Event Sources

EVENT_CUSTOM_CIC	267
EVENT_CUSTOM_CXP_DEVICE	268
EVENT_CUSTOM_CXP_INTERFACE	268
EVENT_CUSTOM_DATASTREAM	270
EVENT_CUSTOM_DEVICE_ERROR	271
EVENT_CUSTOM_IO_TOOLBOX	272

EVENT_CUSTOM_CIC

Camera and Illumination Controller custom event sources (Device module)

Coaxlink GenTL Producer

Event Source	Description
_CAMERA_TRIGGER_RISING_EDGE	Rising edge of the Camera Trigger output signal (Start of Exposure in RC and RG camera control methods)
_CAMERA_TRIGGER_FALLING_EDGE	Falling edge of the Camera Trigger output signal (End of Exposure in RG camera control method)
_STROBE_RISING_EDGE	Rising edge of the Strobe output signal
_STROBE_FALLING_EDGE	Falling edge of the Strobe output signal
_ALLOW_NEXT_CYCLE	A new camera cycle is allowed to start immediately.
_DISCARDED_CIC_TRIGGER	A CIC cycle trigger is discarded.
_PENDING_CIC_TRIGGER	A CIC cycle trigger is recorded, but its execution is delayed until CIC is ready.
_CXP_TRIGGER_ACK	A positive acknowledgment is received in response to a CoaXPress Host to Device Trigger Message.
_CXP_TRIGGER_RESEND	A resent of the CoaXPress Host to Device Trigger Message is executed.
_TRIGGER	CIC trigger



NOTE

There is one Camera and Illumination Controller instance per GenTL Device Module. The number of GenTL Device Modules per frame grabber is defined by the firmware-variant.

EVENT_CUSTOM_CXP_DEVICE

CoaXPress Host Interface custom event sources (Device Module)

Event Source	Description
_LINK_TRIGGER	LinkTrigger<N> received from CoaXPress device

EVENT_CUSTOM_CXP_INTERFACE

CoaXPress Host Interface custom event sources (Interface module)

Event Source	Description
_CRC_ERROR_CXP_A	A CRC error is detected on the Connection A of the CoaXPress Host Interface
_CRC_ERROR_CXP_B	A CRC error is detected on the Connection B of the CoaXPress Host Interface
_CRC_ERROR_CXP_C	A CRC error is detected on the Connection C of the CoaXPress Host Interface
_CRC_ERROR_CXP_D	A CRC error is detected on the Connection D of the CoaXPress Host Interface
_CRC_ERROR_CXP_E	A CRC error is detected on the Connection E of the CoaXPress Host Interface
_CRC_ERROR_CXP_F	A CRC error is detected on the Connection F of the CoaXPress Host Interface
_CRC_ERROR_CXP_G	A CRC error is detected on the Connection G of the CoaXPress Host Interface
_CRC_ERROR_CXP_H	A CRC error is detected on the Connection H of the CoaXPress Host Interface
_CONNECTION_DETECTED_CXP_A	Low level connection lock achieved on CXP connector A
_CONNECTION_DETECTED_CXP_B	Low level connection lock achieved on CXP connector B
_CONNECTION_DETECTED_CXP_C	Low level connection lock achieved on CXP connector C
_CONNECTION_DETECTED_CXP_D	Low level connection lock achieved on CXP connector D
_CONNECTION_DETECTED_CXP_E	Low level connection lock achieved on CXP connector E
_CONNECTION_DETECTED_CXP_F	Low level connection lock achieved on CXP connector F
_CONNECTION_DETECTED_CXP_G	Low level connection lock achieved on CXP connector G
_CONNECTION_DETECTED_CXP_H	Low level connection lock achieved on CXP connector H
_CONNECTION_UNDETECTED_CXP_A	Low level connection lock lost on CXP connector A
_CONNECTION_UNDETECTED_CXP_B	Low level connection lock lost on CXP connector B

Event Source	Description
_CONNECTION_UNDETECTED_CXP_C	Low level connection lock lost on CXP connector C
_CONNECTION_UNDETECTED_CXP_D	Low level connection lock lost on CXP connector D
_CONNECTION_UNDETECTED_CXP_E	Low level connection lock lost on CXP connector E
_CONNECTION_UNDETECTED_CXP_F	Low level connection lock lost on CXP connector F
_CONNECTION_UNDETECTED_CXP_G	Low level connection lock lost on CXP connector G
_CONNECTION_UNDETECTED_CXP_H	Low level connection lock lost on CXP connector H
_DEVICE_0_READY	CoaXPress link configuration done for Device 0
_DEVICE_1_READY	CoaXPress link configuration done for Device 1
_DEVICE_2_READY	CoaXPress link configuration done for Device 2
_DEVICE_3_READY	CoaXPress link configuration done for Device 3
_DEVICE_4_READY	CoaXPress link configuration done for Device 4
_DEVICE_5_READY	CoaXPress link configuration done for Device 5
_DEVICE_6_READY	CoaXPress link configuration done for Device 6
_DEVICE_7_READY	CoaXPress link configuration done for Device 7
_DEVICE_0_LOST	Device 0 disconnected
_DEVICE_1_LOST	Device 1 disconnected
_DEVICE_2_LOST	Device 2 disconnected
_DEVICE_3_LOST	Device 3 disconnected
_DEVICE_4_LOST	Device 4 disconnected
_DEVICE_5_LOST	Device 5 disconnected
_DEVICE_6_LOST	Device 6 disconnected
_DEVICE_7_LOST	Device 7 disconnected
_DEVICE_0_CONFIGURING	CoaXPress link configuration in progress for Device 0
_DEVICE_1_CONFIGURING	CoaXPress link configuration in progress for Device 1
_DEVICE_2_CONFIGURING	CoaXPress link configuration in progress for Device 2
_DEVICE_3_CONFIGURING	CoaXPress link configuration in progress for Device 3
_DEVICE_4_CONFIGURING	CoaXPress link configuration in progress for Device 4
_DEVICE_5_CONFIGURING	CoaXPress link configuration in progress for Device 5
_DEVICE_6_CONFIGURING	CoaXPress link configuration in progress for Device 6
_DEVICE_7_CONFIGURING	CoaXPress link configuration in progress for Device 7

EVENT_CUSTOM_DATASTREAM

Data Stream custom event sources (Data Stream Module)

Coaxlink and Grablink GenTL Producers

Event Source	Description
_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.
_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.
_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.
_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.
_REJECTED_FRAME	An image frame is rejected (On-board FIFO Buffer is full). Applies to area-scan firmware variants only.
_REJECTED_SCAN	An image scan is rejected (On-board FIFO Buffer is full). Applies to line-scan firmware variants only.
_TRIGGER_TO_CAMERA_READOUT_TIMEOUT	Trigger to camera readout timeout.
_CAMERA_READOUT_TIMEOUT	Camera readout timeout.
_BROKEN_FRAME	Broken frame due to frame store overflow. Applies to area-scan firmware variants only.

EVENT_CUSTOM_DEVICE_ERROR

Device Error custom event sources (Device Module)

Coaxlink GenTL Producer

Event Source	Description
_STREAM_PACKET_SIZE_ERROR	Stream packet size error
_STREAM_PACKET_FIFO_OVERFLOW	Stream packet FIFO overflow
_CAMERA_TRIGGER_OVERRUN	New trigger sent to remote device even though readout of previous frame has not started yet
_DID_NOT_RECEIVE_TRIGGER_ACK	Trigger ignored because ACK to previous trigger has not been received yet
_TRIGGER_PACKET_RETRY_ERROR	Trigger packet resend not successful
_INPUT_STREAM_FIFO_HALF_FULL	Input stream FIFO half full
_INPUT_STREAM_FIFO_FULL	Input stream FIFO full
_IMAGE_HEADER_ERROR	Image header error
_MIG_AXI_WRITE_ERROR	MIG AXI write error
_MIG_AXI_READ_ERROR	MIG AXI read error
_PACKET_WITH_UNEXPECTED_TAG	Received a CXP packet with unexpected tag
_FILL_LEVEL_ABOVE_IL_SOS_REJECTED	Start of scan skipped (caused by internal exception: frame store almost full)
_FILL_LEVEL_ABOVE_AF_EARLY_EOS	End of scan (caused by internal exception: frame store almost full)
_EXTERNAL_TRIGGER_REQS_TOO_CLOSE	External trigger requests too close together

EVENT_CUSTOM_IO_TOOLBOX

I/O Toolbox custom event sources (Interface module)

Coaxlink GenTL Producer

Event Source	Description
_LIN1	Line Input Tool 1 – Event output
_LIN2	Line Input Tool 2 – Event output
_LIN3	Line Input Tool 3 – Event output
_LIN4	Line Input Tool 4 – Event output
_LIN5	Line Input Tool 5 – Event output
_LIN6	Line Input Tool 6 – Event output
_LIN7	Line Input Tool 7 – Event output
_LIN8	Line Input Tool 8 – Event output
_QDC1	Quadrature Decoder Tool 1 – Event output
_QDC1_DIR	Quadrature Decoder Tool 1 – Changed direction
_QDC2	Quadrature Decoder Tool 2 – Event output
_QDC2_DIR	Quadrature Decoder Tool 2 – Changed Direction
_QDC3	Quadrature Decoder Tool 3 – Event output
_QDC3_DIR	Quadrature Decoder Tool 3 – Changed direction
_QDC4	Quadrature Decoder Tool 4 – Event output
_QDC4_DIR	Quadrature Decoder Tool 4 – Changed direction
_DIV1	Divider Tool 1 – Event output
_DIV2	Divider Tool 2 – Event output
_DIV3	Divider Tool 3 – Event output
_DIV4	Divider Tool 4 – Event output
_MDV1	Multiplier/Divider Tool 1 – Event output
_MDV2	Multiplier/Divider Tool 2 – Event output
_MDV3	Multiplier/Divider Tool 3 – Event output
_MDV4	Multiplier/Divider Tool 4 – Event output
_DEL1_1	Delay Tool 1 Output 1 – Event output
_DEL1_2	Delay Tool 1 Output 2 – Event output
_DEL2_1	Delay Tool 2 Output 1 – Event output
_DEL2_2	Delay Tool 2 Output 2 – Event output
_DEL3_1	Delay Tool 3 Output 1 – Event output
_DEL3_2	Delay Tool 3 Output 2 – Event output

Event Source	Description
_DEL4_1	Delay Tool 4 Output 1 – Event output
_DEL4_2	Delay Tool 4 Output 2 – Event output
_USER_EVENT_1	User Event 1
_USER_EVENT_2	User Event 2
_USER_EVENT_3	User Event 3
_USER_EVENT_4	User Event 4
_C2C1	C2C Synchronization Tool Output 1 – Event output
_C2C2	C2C Synchronization Tool Output 2 – Event output
_C2C3	C2C Synchronization Tool Output 3 – Event output
_EIN1	Event Input Tool 1 – Event output
_EIN2	Event Input Tool 2 – Event output
_DLT1	Device Link Trigger Tool 1 – Event output
_DLT2	Device Link Trigger Tool 2 – Event output
_DLT3	Device Link Trigger Tool 3 – Event output
_DLT4	Device Link Trigger Tool 4 – Event output
_DLT5	Device Link Trigger Tool 5 – Event output
_DLT6	Device Link Trigger Tool 6 – Event output
_DLT7	Device Link Trigger Tool 7 – Event output
_DLT8	Device Link Trigger Tool 8 – Event output
_DLT9	Device Link Trigger Tool 9 – Event output
_DLT10	Device Link Trigger Tool 10 – Event output
_DLT11	Device Link Trigger Tool 11 – Event output
_DLT12	Device Link Trigger Tool 12 – Event output
_DLT13	Device Link Trigger Tool 13 – Event output
_DLT14	Device Link Trigger Tool 14 – Event output
_DLT15	Device Link Trigger Tool 15 – Event output
_DLT16	Device Link Trigger Tool 16 – Event output



NOTE

Check the "[I/O Toolbox Composition Tables](#)" on page 241 for applicable values

12.3. Event Specific Context Data

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: frame store 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: frame store 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: reached scan length
4	Cause: DSStopAcquisition
5	Cause: internal exception (frame store 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: frame store is full.
1	Cause: data stream is not active
2	Cause: frame store underwent overflow

EVENT_DATA_NUMID_CXP_DEVICE_LINK_TRIGGER

Value of *EventSpecific* for GenTL::EuresysCustomGenTL::EVENT_DATA_NUMID_CXP_DEVICE_LINK_TRIGGER is a bitfield that can be interpreted according to the following definition:

Bit#	Description
7:0	LinkTriggerN (Word 2 of the CoaXPress 2.0 high speed connection trigger packet)
15:8	Delay (Word 1 of the CoaXPress 2.0 high speed connection trigger packet)
31:16	Reserved



WARNING

Undocumented bits must be ignored.

12.4. About GenTL Signaling

The **eGrabber** driver implements the Signaling mechanism of a GenTL Producer.

This mechanism is briefly described hereafter.

See also: section 4.2 starting on page 34 of the [GenICam GenTL Standard Version 1.4](#) for an extensive description.

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.



WARNING

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.

13. Advanced Features

- 13.1. C2C-Link 278**
 - C2C-Link Interconnections 279
 - C2C-Link Electrical Specification 280
 - Trigger Propagation Delays 282
 - Cycle Trigger Synchronization 284
 - C2C-Link Setup Procedure 287
- 13.2. OEM Safety Key 289**
 - Introducing OEM Safety Key 290
 - Using OEMSafetyKey 291

13.1. C2C-Link

C2C-Link Interconnections279

C2C-Link Electrical Specification280

Trigger Propagation Delays282

Cycle Trigger Synchronization284

C2C-Link Setup Procedure287

C2C-Link Interconnections

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

In area-scan applications, the C2C-Link Interconnect:

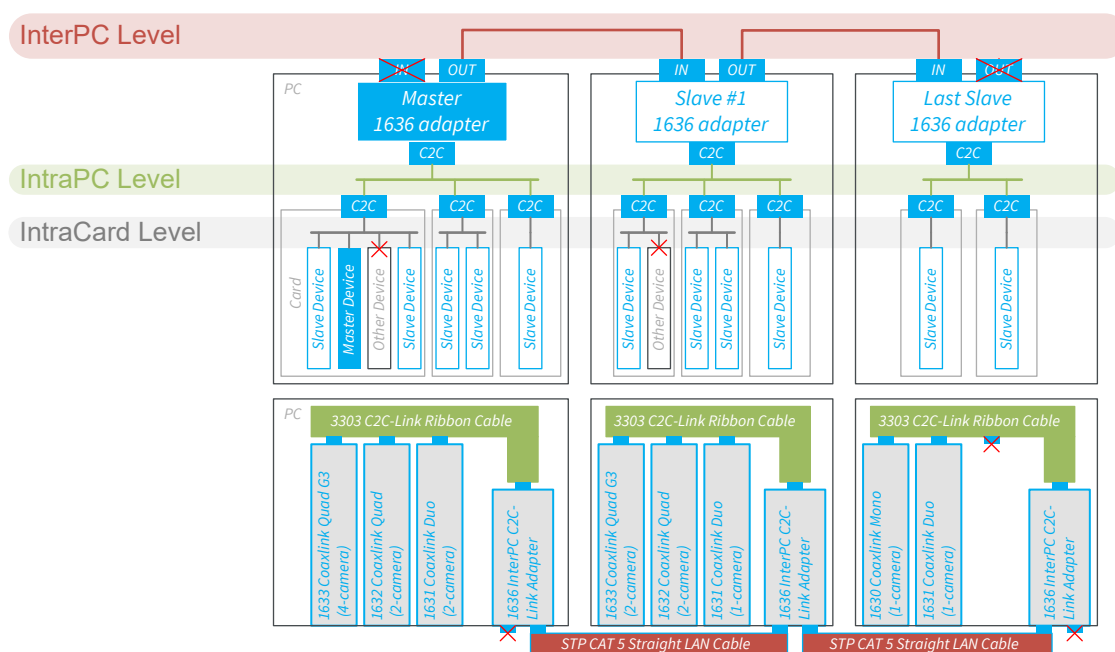
- allows to share up to 2 triggers: a CIC Cycle Trigger (mandatory) and one I/O Toolbox event (optional).
- allows any C2C-Link device to indicate to the C2C-Link master device that it is not able to accept a trigger.

In line-scan applications, the C2C-Link Interconnect allows to share up to 3 triggers: a CIC Cycle Trigger (mandatory) and two I/O Toolbox events (optional).

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

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 **PC3303 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 **PC1636 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 configuration example using InterPC, IntraPC and IntraCard levels

C2C-Link Electrical Specification

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: 5 ns/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		8	-
Cable length	-		1.2	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

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) PC1636 InterPC C2C-Link Adapter – IntraPC-to-InterPC delay		15		ns
(3) PC1636 InterPC C2C-Link Adapter – InterPC-to-IntraPC delay		20		ns
(4) PC1636 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
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:

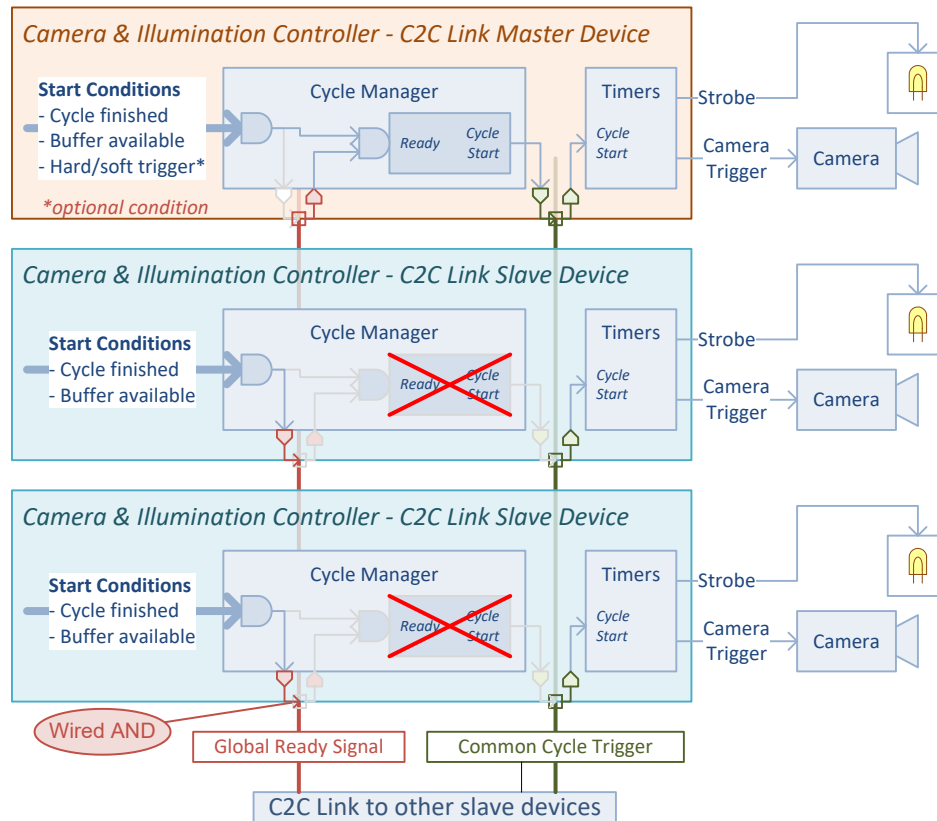
Parameter	Min.	Typ.	Max.	Units
(1) Master IntraPC interconnection	0	5	10	ns
(2) Master PC1636 InterPC C2C-Link Adapter – IntraPC-to-InterPC delay		15		ns
(5) Master to slave1 InterPC LAN cable – 20 m		100		ns
(3) Slave1 PC1636 InterPC C2C-Link Adapter – InterPC-to-IntraPC delay		20		ns
(1) Slave1 IntraPC 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 PC1636 InterPC C2C-Link Adapter – IntraPC-to-InterPC delay		15		ns
(5) Master to slave1 InterPC LAN cable – 20 m		100		ns
(4) Slave1 PC1636 InterPC C2C-Link Adapter – InterPC-to-InterPC delay		0		ns
(5) Slave1 to slave2 InterPC LAN cable – 20 m		100		ns
(3) Slave2 PC1636 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

Cycle Trigger Synchronization

Principle



One C2C-Link master and two C2C-Link slave devices (With Global Ready)

Common cycle trigger

The CIC synchronization is achieved by sharing a common **Cycle Trigger** event between all involved CIC 's using the C2C-Link interconnections. The C2C-Link interconnects two or more devices. One device is named *C2C-Link Master Device*, the others are named *C2C-Link Slave Device*.

The **Cycle Trigger** is generated by the Cycle Manager of the "*C2C-Link Master Device*" and broadcasted on the C2C-Link via the "**C2C-Link Synchronization Tool**" on page 254 of the I/O toolbox. All the participating C2C-Link devices (1 Master and one or more slaves) uses the shared C2C1 event stream as event source for the camera cycle trigger.

The CIC Cycles of all participating devices start simultaneously. However, the Cycle Timing parameters (exposure time, strobe pulse width and strobe delay) can be configured individually.

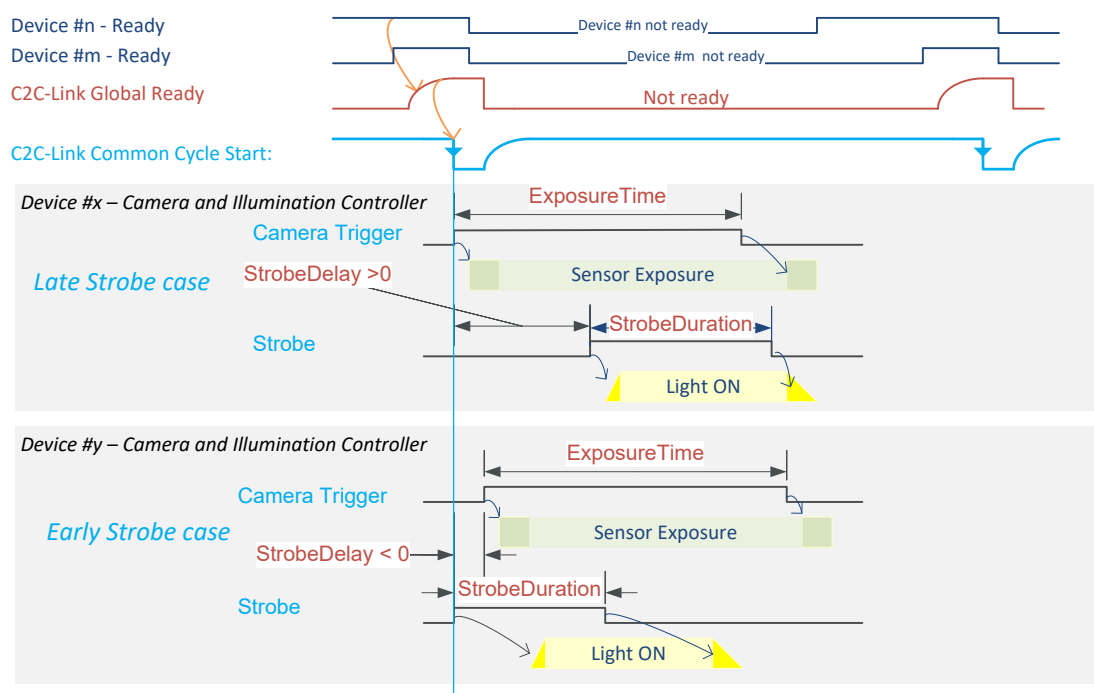
Global Ready signal

Global Ready applies only to area-scan firmware variants!

The Global Ready signal is elaborated by all C2C-Link slaves using a wired AND connection logic. A C2C-Link Slave forces the Global Ready signal to 'false' until its start conditions are all satisfied.

The C2C-Link Master device further delays the assertion a Cycle Trigger event while the Global Ready signal is false.

Timing Diagram



CIC Synchronization through C2C-Link timing diagram (With Global Ready)

The above diagram shows the timing diagram of two consecutive common `CycleTrigger` events with Global Ready :

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.

C2C-Link Setup Procedure

Hardware setup

This step is specific to each C2C-Link Configuration:

IntraCard C2C-Link configuration

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

IntraPC C2C-Link configuration

This configuration requires one accessory cable such as the **PC3303 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 frame grabber.

InterPC C2C-Link configuration

This configuration requires one **PC1636 InterPC C2C-Link Adapter** for each PC and one RJ 45 CAT 5 STP straight LAN cable for each adapter but the last one.

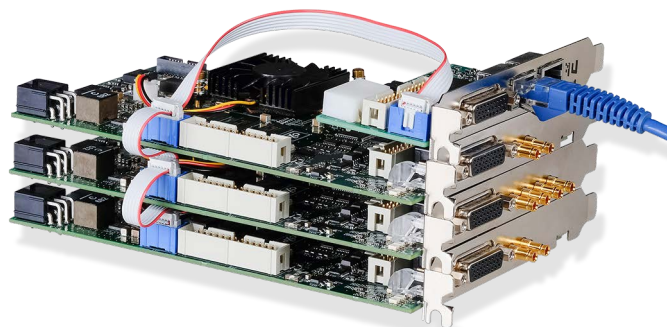
In each participating PC:

1. Install **PC1636 InterPC C2C-Link Adapter** into a free slot and secure the bracket.
2. Connect the adapter to a power source.

See also: "Adapter Powering" topic in the 1636 section of the hardware manual.

3. Using one **PC3303 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.

See also: "InterPC Interconnect" topic in the 1636 section of the hardware manual.



IntraPC segment of an InterPC configuration

GenApi setup

Master C2C-Link device

- Assign value **Master** to **C2CLinkConfiguration** of the GenTL Device module
- Configure the I/O Toolbox C2C1 tool of the GenTL Interface module to share Cycle Trigger:
 - Assign value **CycleTrigger** to **C2CLinkSynchronizationToolSource**
 - Assign value **Immediate** to **C2CLinkSynchronizationToolClock**
- Optional, configure the I/O Toolbox C2C2 and C2C3 tools of the GenTL Interface module according to the application requirements

Slave C2C-Link devices

Assign value **Slave** to **C2CLinkConfiguration** of the GenTL Device module

13.2. OEM Safety Key

Introducing OEM Safety Key	290
Using OEMSafetyKey	291

Introducing OEM Safety Key

The OEM Safety Key capability allows the application to:

- Program an "OEM safety key" in the non-volatile memory of the frame grabber.
- 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** GenApi feature with the OEM Safety Key value, the **eGrabber** driver computes an encrypted version of the OEM Safety Key and stores it in the non-volatile memory of the frame grabber.

The encrypted value can be retrieved by getting the value of **EncryptedOemSafetyKey** immediately after having set **ProgramOemSafetyKey**.



WARNING

Only the same application process having set **ProgramOemSafetyKey** is allowed to retrieve the encrypted value. This is only allowed until any other GenApi 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 frame grabber, 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.

AVT 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.

Using OEMSafetyKey

Programming Step – Option A

Using **GenICam Browser (Deprecated)**:

- 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 frame grabbers. 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 frame grabber 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::string encryptedKey=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.

PART III
HARDWARE MANUAL

1. Mechanical Specification

Mechanical specifications of the product(s) including: product pictures, physical dimensions, connectors description and pin assignments, LEDs description, switches description, etc.

- 1.1. Product Layouts294
- 1.2. Camera and Data Output Connectors 300
- 1.3. GPIO Connectors306
- 1.4. Other Connectors314
- 1.5. LEDs and Switches320
- 1.6. Physical Characteristics326

1.1. Product Layouts

PC1633-T Coaxlink Quad G3 295

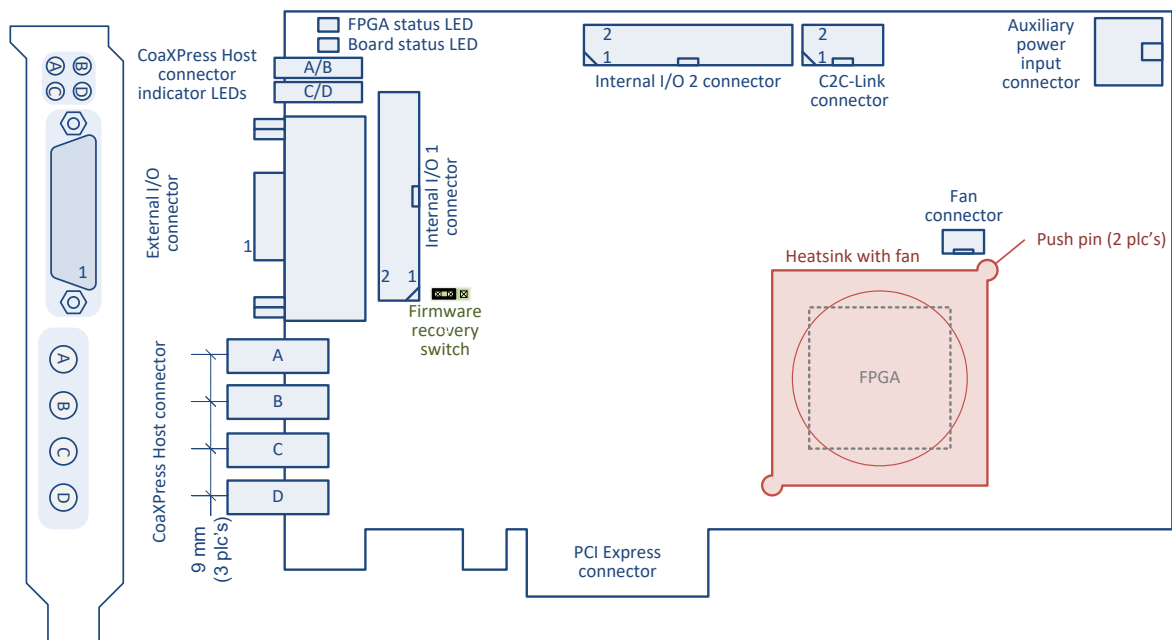
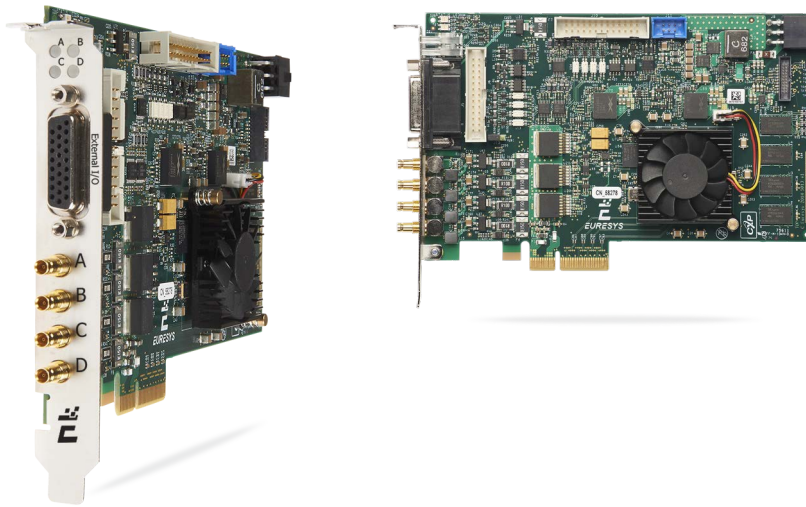
PC3602-T Coaxlink Octo 296

PC3621-LH-T Coaxlink Mono CXP-12-LH 297

PC3622-T Coaxlink Duo CXP-12 298

PC3623-T Coaxlink Quad CXP-12 Value 299

PC1633-T Coaxlink Quad G3

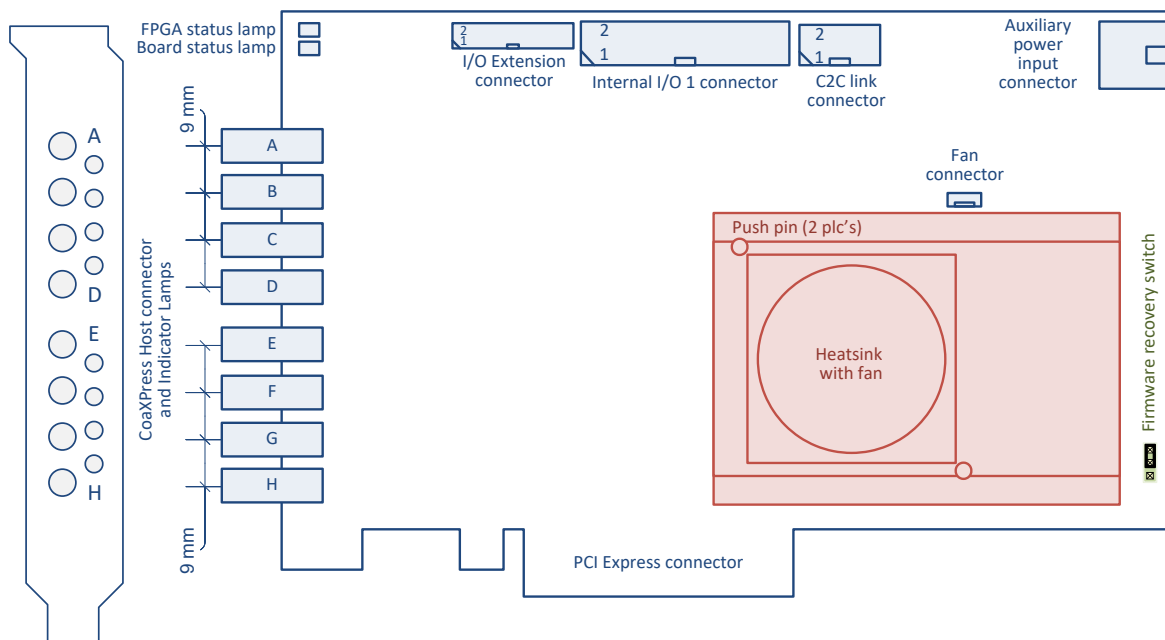
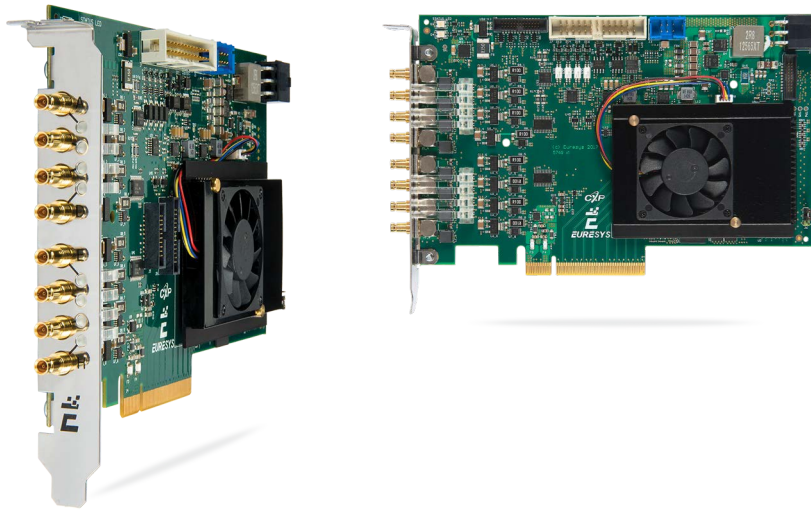


Connectors

- "Auxiliary Power Input Connector for PoCXP and GPIO" on page 318
- "C2C-Link Connector" on page 317
- "CoaXPress DIN 4 Connector" on page 301
- "External I/O Connector" on page 307
- "Internal I/O 1 Connector" on page 310
- "Internal I/O 2 Connector" on page 312
- "CoaXPress LED Lamps" on page 321
- "Board Status LED" on page 323
- "FPGA Status LED" on page 324
- "Firmware Recovery Switch" on page 325

Lamps and switches

PC3602-T Coaxlink Octo



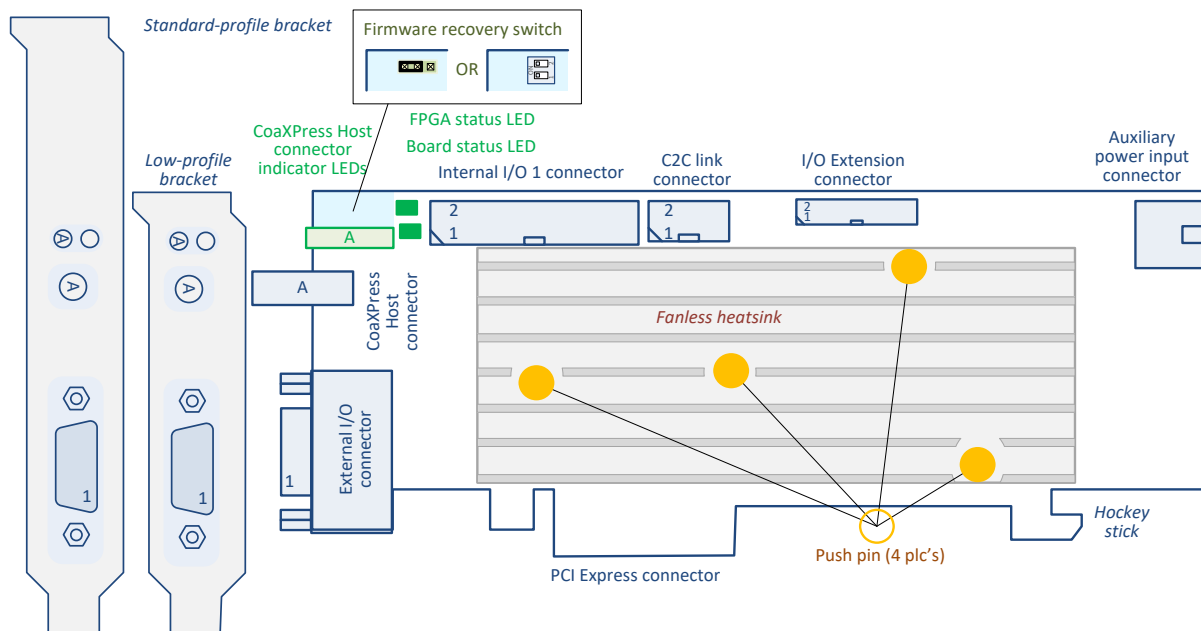
Connectors

- "Auxiliary Power Input Connector w/o SenseIN" on page 319
- "C2C-Link Connector" on page 317
- "CoaXPRESS DIN 8 Connector" on page 302
- "I/O Extension Connector" on page 315
- "Internal I/O 1 Connector" on page 310

Lamps and switches

- "CoaXPRESS LED Lamps" on page 321
- "Board Status LED" on page 323
- "FPGA Status LED" on page 324
- "Firmware Recovery Switch" on page 325

PC3621-LH-T Coaxlink Mono CXP-12-LH



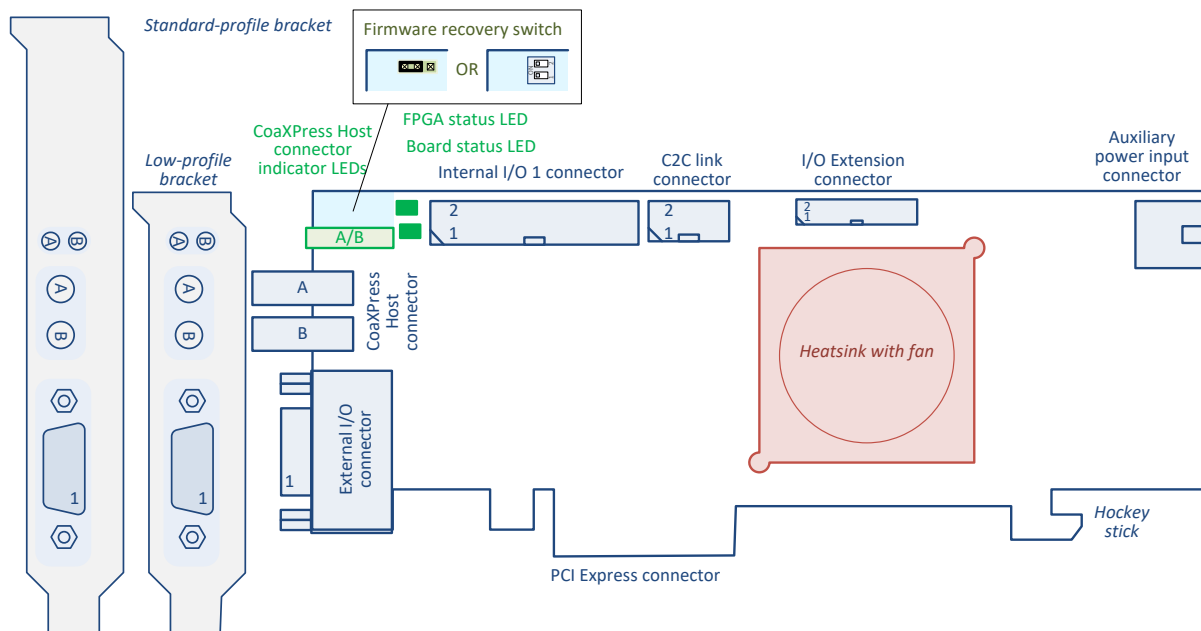
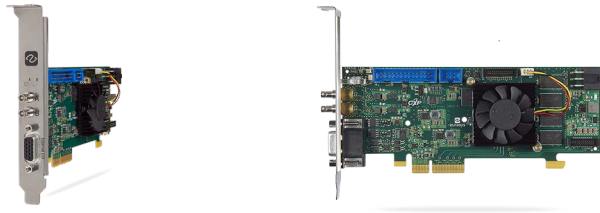
Connectors

- "Auxiliary Power Input Connector w/o SenseIN" on page 319
- "C2C-Link Connector" on page 317
- "CoaXPress Micro-BNC 1 Connector" on page 303
- "External I/O 15-pin Connector " on page 309
- "I/O Extension Connector" on page 315
- "Internal I/O 1 Connector" on page 310

Lamps and switches

- "CoaXPress LED Lamps" on page 321
- "Board Status LED" on page 323
- "FPGA Status LED" on page 324
- "Firmware Recovery Switch" on page 325

PC3622-T Coaxlink Duo CXP-12



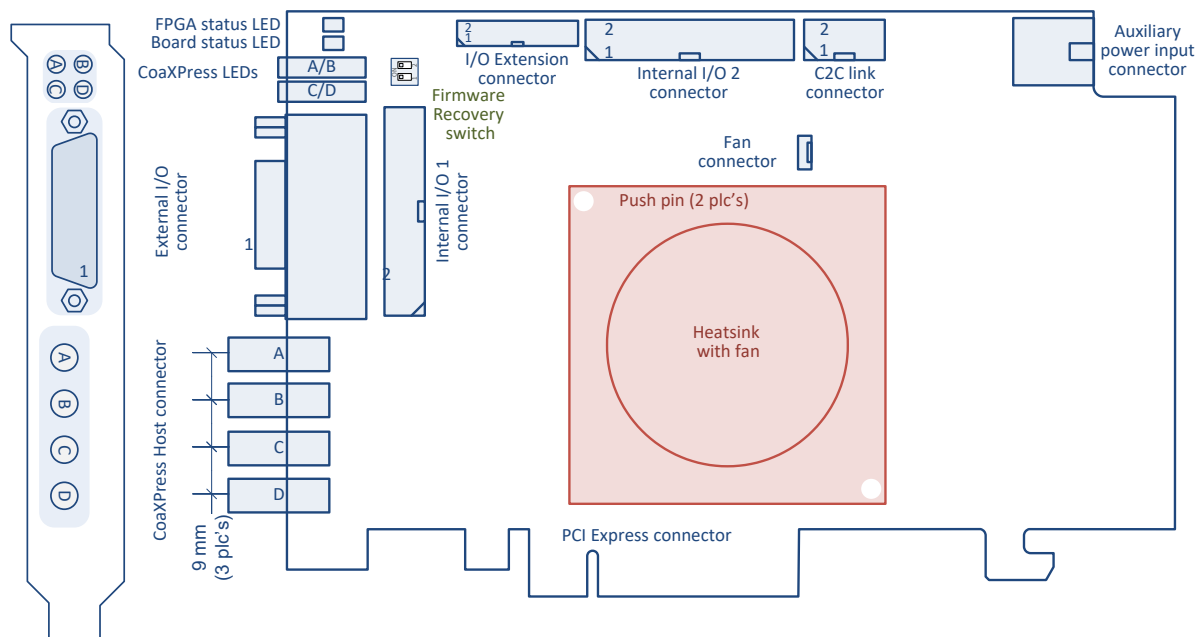
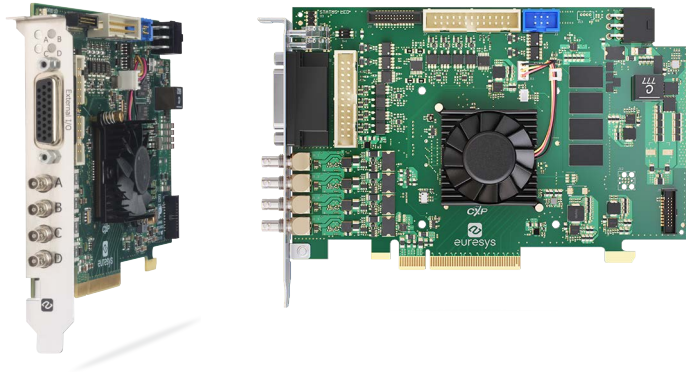
Connectors

- "Auxiliary Power Input Connector w/o SenseIN" on page 319
- "C2C-Link Connector" on page 317
- "CoaXPress Micro-BNC 2 Connector" on page 304
- "External I/O 15-pin Connector " on page 309
- "I/O Extension Connector" on page 315
- "Internal I/O 1 Connector" on page 310

Lamps and switches

- "CoaXPress LED Lamps" on page 321
- "Board Status LED" on page 323
- "FPGA Status LED" on page 324
- "Firmware Recovery Switch" on page 325

PC3623-T Coaxlink Quad CXP-12 Value



Connectors

- "Auxiliary Power Input Connector w/o SenseIN" on page 319
- "C2C-Link Connector" on page 317
- "CoaXPress Micro-BNC 4 Connector" on page 305
- "External I/O Connector" on page 307
- "I/O Extension Connector" on page 315
- "Internal I/O 1 Connector" on page 310
- "Internal I/O 2 Connector" on page 312

Lamps and switches

- "CoaXPress LED Lamps" on page 321
- "Board Status LED" on page 323
- "FPGA Status LED" on page 324
- "Firmware Recovery Switch" on page 325

1.2. Camera and Data Output Connectors

CoaXPress DIN 4 Connector	301
CoaXPress DIN 8 Connector	302
CoaXPress Micro-BNC 1 Connector	303
CoaXPress Micro-BNC 2 Connector	304
CoaXPress Micro-BNC 4 Connector	305

CoaXPress DIN 4 Connector

Applies to PC1633-T Coaxlink Quad G3.



Connector description

Property	Value
Name, Label	A B C D
Type	4 x DIN 1.0/2.3 75 Ohms coaxial receptacles
Location	Card bracket
Usage	CoaXPress Host Interface



Pin assignments

Pin	Signal	Usage
Inner1	CXP_A	CoaXPress Host Connection A
Outer1	GND	Ground
Inner2	CXP_B	CoaXPress Host Connection B
Outer2	GND	Ground
Inner3	CXP_C	CoaXPress Host Connection C
Outer3	GND	Ground
Inner4	CXP_D	CoaXPress Host Connection D
Outer4	GND	Ground

CoaXPress DIN 8 Connector

Applies to PC3602-T Coaxlink Octo.



Connector description

Property	Value
Name, Label	A B C D E F G H
Type	8 x DIN 1.0/2.3 75 Ohms coaxial receptacles
Location	Card bracket
Usage	CoaXPress Host Interface



Pin assignments

Pin	Signal	Usage
Inner1	CXP_A	CoaXPress Host Connection A
Outer1	GND	Ground
Inner2	CXP_B	CoaXPress Host Connection B
Outer2	GND	Ground
Inner3	CXP_C	CoaXPress Host Connection C
Outer3	GND	Ground
Inner4	CXP_D	CoaXPress Host Connection D
Outer4	GND	Ground
Inner5	CXP_E	CoaXPress Host Connection E
Outer5	GND	Ground
Inner6	CXP_F	CoaXPress Host Connection F
Outer6	GND	Ground
Inner7	CXP_G	CoaXPress Host Connection G
Outer7	GND	Ground
Inner8	CXP_H	CoaXPress Host Connection H
Outer8	GND	Ground

CoaXPress Micro-BNC 1 Connector

Applies to PC3621-LH-T Coaxlink Mono CXP-12-LH.



Connector description

Property	Value
Name, Label	A
Type	Micro-BNC 75 Ohms coaxial receptacle
Location	Card bracket
Usage	CoaXPress Host Interface



Pin assignments

Pin	Signal	Usage
Inner1	CXP_A	CoaXPress Host Connection A
Outer1	GND	Ground

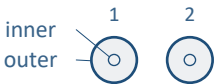
CoaXPress Micro-BNC 2 Connector

Applies to PC3622-T Coaxlink Duo CXP-12.



Connector description

Property	Value
Name, Label	A B
Type	2 x Micro-BNC 75 Ohms coaxial receptacles
Location	Card bracket
Usage	CoaXPress Host Interface



Pin assignments

Pin	Signal	Usage
Inner1	CXP_A	CoaXPress Host Connection A
Outer1	GND	Ground
Inner2	CXP_B	CoaXPress Host Connection B
Outer2	GND	Ground

CoaXPress Micro-BNC 4 Connector

Applies to PC3623-T Coaxlink Quad CXP-12 Value.



Connector description

Property	Value
Name, Label	A B C D
Type	4 x Micro-BNC 75 Ohms coaxial receptacles
Location	Card bracket
Usage	CoaXPress Host Interface



Pin assignments

Pin	Signal	Usage
Inner1	CXP_A	CoaXPress Host Connection A
Outer1	GND	Ground
Inner2	CXP_B	CoaXPress Host Connection B
Outer2	GND	Ground
Inner3	CXP_C	CoaXPress Host Connection C
Outer3	GND	Ground
Inner4	CXP_D	CoaXPress Host Connection D
Outer4	GND	Ground

1.3. GPIO Connectors

External I/O Connector307

External I/O 15-pin Connector309

Internal I/O 1 Connector310

Internal I/O 2 Connector312

External I/O Connector

Applies to PC1633-T Coaxlink Quad G3 and PC3623-T Coaxlink Quad CXP-12 Value.



Connector description

Property	Value
Name, Label	External I/O
Type	26-pin 3-row high-density D-Sub female socket with UNC4-40 jack socket screws
Location	Card bracket
Usage	I/O lines and I/O power output



Pin assignments

Standard I/O set #1

Pin	Signal	Usage
1	GND	Ground
2	DIN12+	High-speed differential input #12 – Positive pole
3	IIN11+	Isolated input #11 – Positive pole
4	IIN13-	Isolated input #13 – Negative pole
5	IIN14-	Isolated input #14 – Negative pole
6	IOUT12-	Isolated contact output #12 – Negative pole
7	GND	Ground
8		Not connected
9	GND	Ground
10	GND	Ground
11	DIN12-	High-speed differential input #12 – Negative pole
12	IIN11-	Isolated input #11 – Negative pole
13	IIN12+	Isolated input #12 – Positive pole
14	IIN13+	Isolated input #13 – Positive pole
15	IIN14+	Isolated input #14 – Positive pole
16	IOUT12+	Isolated contact output #12 – Positive pole

Pin	Signal	Usage
17	TTLIO12	TTL input/output #12
18	GND	Ground
19	DIN11-	High-speed differential input #11 – Negative pole
20	DIN11+	High-speed differential input #11 – Positive pole
21	IIN12-	Isolated input #12 – Negative pole
22	IOUT11-	Isolated contact output #11 – Negative pole
23	IOUT11+	Isolated contact output #11 – Positive pole
24	GND	Ground
25	TTLIO11	TTL input/output #11
26	+12V	+12 V Power output

External I/O 15-pin Connector

Applies to **PC3621-LH-T Coaxlink Mono CXP-12-LH** and **PC3622-T Coaxlink Duo CXP-12**.



Connector description

Property	Value
Name, Label	External I/O
Type	15-pin 3-row high-density D-Sub female socket with UNC4-40 jack socket screws
Location	Card bracket
Usage	I/O lines and I/O power output



Pin assignments

Pin	Signal	Usage
1	DIN12+	High-speed differential input #12 – Positive pole
2	IIN11+	Isolated input #11 – Positive pole
3	IIN12+	Isolated input #12 – Positive pole
4	TTLIO11	TTL input/output #11
5	GND	Ground
6	DIN11+	High-speed differential input #11 – Positive pole
7	DIN12-	High-speed differential input #12 – Negative pole
8	IIN12-	Isolated input #12 – Negative pole
9	IOUT11+	Isolated contact output #11 – Positive pole
10	GND	Ground
11	DIN11-	High-speed differential input #11 – Negative pole
12	IIN11-	Isolated input #11 – Negative pole
13	IOUT11-	Isolated contact output #11 – Negative pole
14	TTLIO12	TTL input/output #12
15	+12V	+12 V Power output

Internal I/O 1 Connector

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.



Connector description

Property	Value
Name, Label	Internal I/O 1
Type	26-pin 2-row 0.1" pitch pin header with shrouding
Location	Printed circuit board
Usage	I/O lines and I/O power output



Pin assignments

Standard I/O set #1

Pin	Signal	Usage
1	GND	Ground
2	GND	Ground
3	DIN11+	High-speed differential input #11 – Positive pole
4	DIN11-	High-speed differential input #11 – Negative pole
5	DIN12+	High-speed differential input #12 – Positive pole
6	DIN12-	High-speed differential input #12 – Negative pole
7	IIN11+	Isolated input #11 – Positive pole
8	IIN11-	Isolated input #11 – Negative pole
9	IIN12+	Isolated input #12 – Positive pole
10	IIN12-	Isolated input #12 – Negative pole
11	IIN13+	Isolated input #13 – Positive pole
12	IIN13-	Isolated input #13 – Negative pole
13	IIN14+	Isolated input #14 – Positive pole
14	IIN14-	Isolated input #14 – Negative pole

Pin	Signal	Usage
15	IOUT11+	Isolated contact output #11 – Positive pole
16	IOUT11-	Isolated contact output #11 – Negative pole
17	IOUT12+	Isolated contact output #12 – Positive pole
18	IOUT12-	Isolated contact output #12 – Negative pole
19	TTLIO11	TTL input/output #11
20	GND	Ground
21	TTLIO12	TTL input/output #12
22	GND	Ground
23		Not connected
24	GND	Ground
25	+12V	+12 V Power output
26	+12V_RTN	Ground

Internal I/O 2 Connector

Applies to PC1633-T Coaxlink Quad G3 and PC3623-T Coaxlink Quad CXP-12 Value.



Connector description

Property	Value
Name, Label	Internal I/O 2
Type	26-pin 2-row 0.1" pitch pin header with shrouding
Location	Printed circuit board
Usage	I/O lines and I/O power output



Pin assignments

Standard I/O set #2

Pin	Signal	Usage
1	GND	Ground
2	GND	Ground
3	DIN21+	High-speed differential input #21 – Positive pole
4	DIN21-	High-speed differential input #21 – Negative pole
5	DIN22+	High-speed differential input #22 – Positive pole
6	DIN22-	High-speed differential input #22 – Negative pole
7	IIN21+	Isolated input #21 – Positive pole
8	IIN21-	Isolated input #21 – Negative pole
9	IIN22+	Isolated input #22 – Positive pole
10	IIN22-	Isolated input #22 – Negative pole
11	IIN23+	Isolated input #23 – Positive pole
12	IIN23-	Isolated input #23 – Negative pole
13	IIN24+	Isolated input #24 – Positive pole
14	IIN24-	Isolated input #24 – Negative pole
15	IOU21+	Isolated contact output #21 – Positive pole
16	IOU21-	Isolated contact output #21 – Negative pole

Pin	Signal	Usage
17	IOUT22+	Isolated contact output #22 – Positive pole
18	IOUT22-	Isolated contact output #22 – Negative pole
19	TTLIO21	TTL input/output #21
20	GND	Ground
21	TTLIO22	TTL input/output #22
22	GND	Ground
23		Not connected
24	GND	Ground
25	+12V	+12 V Power output
26	+12V_RTN	Ground

1.4. Other Connectors

I/O Extension Connector315

C2C-Link Connector317

Auxiliary Power Input Connector for PoCXP and GPIO318

Auxiliary Power Input Connector w/o SenseIN319

I/O Extension Connector

Applies to PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

Octo	Prelim	Mono12LH	Prelim	Duo12	Prelim	Value12	Prelim
------	--------	----------	--------	-------	--------	---------	--------

Connector description

Property	Value
Name, Label	I/O Extension
Type	26-pin 2-row 0.05" pitch pin header with shrouding
Location	Printed circuit board
Usage	I/O extension cable socket



Pin assignments

Pin	Signal	Usage
1	IOEXT1WIRE	1-wire serial I/O
2	GND	Ground
3	IOEXT01	I/O Extension #1
4	+3V3	+3.3 V Power
5	IOEXT02	I/O Extension #2
6	GND	Ground
7	IOEXT03	I/O Extension #3
8	+3V3	+3.3 V Power
9	IOEXT04	I/O Extension #4
10	GND	Ground
11	IOEXT05	I/O Extension #5
12	+3V3	+3.3 V Power
13	IOEXT06	I/O Extension #6
14	GND	Ground
15	IOEXT07	I/O Extension #7
16	+3V3	+3.3 V Power
17	IOEXT08	I/O Extension #8

Pin	Signal	Usage
18	GND	Ground
19	IOEXT09	I/O Extension #9
20	12V	12 V Power
21	IOEXT10	I/O Extension #10
22	GND	Ground
23	IOEXT11	I/O Extension #11
24	12V	12 V Power
25	IOEXT12	I/O Extension #12
26	GND	Ground

C2C-Link Connector

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.



Connector description

Property	Value
Name, Label	C2C-Link
Type	6-pin 2-row 0.1" pitch pin header with shrouding
Location	Printed circuit board
Usage	Card-to-card link



Pin assignments

Pin	Signal	Usage
1	GND	Ground
2	CSync1	Card-to-card synchronization bus – Signal 1
3	GND	Ground
4	CSync2	Card-to-card synchronization bus – Signal 2
5	GND	Ground
6	CSync3	Card-to-card synchronization bus – Signal 3

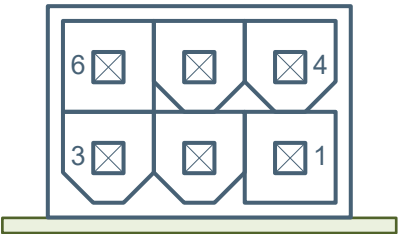
Auxiliary Power Input Connector for PoCXP and GPIO

Applies to PC1633-T Coaxlink Quad G3.



Connector description

Property	Value
Name, Label	Auxiliary Power Input
Type	6-pin PEG power socket
Location	Printed circuit board
Usage	12 V DC power input for PoCXP and I/O power output



Pin assignments

Pin	Signal	Usage
1	+12VIN	Auxiliary +12 V input
2	+12VIN	Auxiliary +12 V input
3	+12VIN	Auxiliary +12 V input
4	GND	Ground
5	SenseIN	Power source presence detection
6	GND	Ground

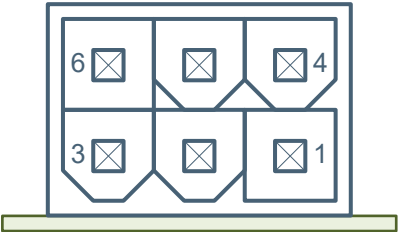
Auxiliary Power Input Connector w/o SenseIN

Applies to PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

Octo	Prelim	Mono12LH	Prelim	Duo12	Prelim	Value12	Prelim
------	--------	----------	--------	-------	--------	---------	--------

Connector description

Property	Value
Name, Label	Auxiliary Power Input
Type	6-pin PEG power socket
Location	Printed circuit board
Usage	12 V DC power input for PoCXP and I/O power output



Pin assignments

Pin	Signal	Usage
1	+12VIN	Auxiliary +12 V input
2	+12VIN	Auxiliary +12 V input
3	+12VIN	Auxiliary +12 V input
4	GND	Ground
5	GND	Ground
6	GND	Ground

1.5. LEDs and Switches

CoaXPress LED Lamps	321
Board Status LED	323
FPGA Status LED	324
Firmware Recovery Switch	325

CoaXPress LED Lamps

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.



Each CoaXPress connection is associated with a LED lamp mounted on the bracket (for PCIe cards only).




CoaXPress Host Indicator LED lamps states

States description

Symbol	LED State	Meaning
	Off	No power
	Solid orange	System booting
	AlternateFlash_12_5 green / orange ¹	Connection detection in progress; PoCXP active
	Flash_12_5 orange ²	Connection detection in progress; PoCXP not in use
	AlternateFlash_0_5 red / green	Device/ Host incompatible; PoCXP active
	AlternateFlash_0_5 red / orange	Device/ Host incompatible; PoCXP not in use
	Solid red	PoCXP over-current
	Solid green	Device / Host connected, but no data being transferred
	Flash_1 orange	Device / Host connected, waiting for event (e.g. trigger, exposure pulse)
	Flash_12_5 green	Device / Host connected, data being transferred

¹ Shown for a minimum of 1 second even if the connection detection is faster

² Shown for a minimum of 1 second even if the connection detection is faster

Symbol	LED State	Meaning
	500 ms red pulse ¹	Error during data transfer (e.g. CRC error, single bit error detected)
	AlternateFlash_0_5 green / orange	Connection test packets being sent
	Flash_12_5 red	System error (e.g. internal error)

Flashing states timing definitions

Indication	Frequency	Duty Cycle
Flash_12_5	12.5 Hz	25% (20 milliseconds on, 60 milliseconds off)
Flash_1	1 Hz	20% (200 milliseconds on, 800 milliseconds off)
Flash_0_5	0.5 Hz	50% (1 second on, 1 second off)
AlternateFlash_12_5	12.5 Hz	25% (20 milliseconds on color 1, 60 milliseconds off, 20 milliseconds on color 2, 60 milliseconds off)
AlternateFlash_0_5	0.5 Hz	50% (1 second on color 1, 1 second off, 1 second on color 2, 1 second off)

LED lamps mode control




The **LampMode** feature of the Interface module defines the lamps operation mode:

- When set to **Standard** (default value), the lamps indicate the state of the CoaXPress Link connection.
- When set to **Dark**, all lamps are turned off.
- When set to **Error**, all lamps are turned off unless error conditions are detected.
- When set to **Custom**, all lamps are controlled by **LampCustomValue**, a bitfield where each bit is mapped onto a lamp with 1 for orange and 0 for off by the **LampCustomLedA** ... **LampCustomLedH** boolean features.

¹ In case of multiple errors, there shall be at least two green Flash_12_5 pulses before the next error is indicated




Board Status LED

Board status LED indicator states

LED state	Symbol	Meaning
Off		No power. The board is not powered or the power distribution network is not functional.
		FPGA NOK. The FPGA start-up procedure is not completed. <i>The normal completion time is around 100 milliseconds.</i>
Solid green		Board status OK. The main power distribution network is operational and the FPGA start-up procedure has successfully completed.
Solid red		Board status NOK. Possible causes are: <ul style="list-style-type: none">• There is no power delivered on the +12 V rail of the PCI Express connector slot• At least one power converter of the main power distribution network is unable to operate properly. <i>This might be caused by excessive temperature due to inadequate board cooling, accidental short-circuits having blown one (or more) protection fuses, inappropriate supply voltages, etc.</i>

FPGA Status LED

FPGA status LED indicator states

LED state	Symbol	Meaning
Solid green		FPGA status OK. All the FPGA clock networks and the DDR memory are operating normally.
Solid red		FPGA status NOK. Possible causes: <ul style="list-style-type: none">• At least one FPGA clock network is not operating normally. <i>This might be caused by excessive jitter on external clock signals of the CoaXPress or the PCI Express interfaces.</i>• The DDR memory controller has not been able to successfully perform the calibration procedure.
Solid orange		FPGA status NOK. Possible causes: <ul style="list-style-type: none">• The DDR memory controller has not been able to successfully perform the calibration procedure.• The eGrabber package doesn't have the second part of the FPGA binary file. <i>A firmware update of the frame grabber is required.</i>¹

¹ Only for boards using Tandem PCIe FPGA configuration: **PC3623 Coaxlink Quad CXP-12 Value**, **PC3624 Coaxlink Quad CXP-12 DF** and **PC3633 Coaxlink Quad CXP-12 3D-LLE**

Firmware Recovery Switch

Switch types and location

The *firmware recovery switch* is implemented with one of the following components:

- 3-pin header and a jumper
- 2-way DIP switch

See also: "Product Layouts" on page 294 in the Handbook to locate the firmware recovery switch. .

Switch positions



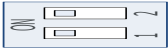
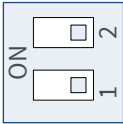
The *firmware recovery switch* has two positions:

Normal position (factory default)

At the next power ON, the latest firmware successfully written into the Flash EEPROM is used to program the FPGA. After FPGA startup completion, the card exhibits the *standard PCI ID* and the driver allows normal operation.

Recovery position

At the next power ON, the last but one firmware successfully written into the Flash EEPROM is used to program the FPGA. After FPGA startup completion, the card exhibits the *recovery PCI ID* and the driver inhibits image acquisition.

Switch type	Normal position	Recovery position
3-pin header and a jumper		
2-way DIP switch		

1.6. Physical Characteristics

Weight

Product	Gross weight		Net weight	
PC1633-T Coaxlink Quad G3	274 g	9.66 oz	173 g	6.10 oz
PC3602-T Coaxlink Octo	330 g	11.64 oz	185 g	6.53 oz
PC3621-LH-T Coaxlink Mono CXP-12-LH	275 g	9.70 oz	159 g	5.61 oz
PC3622-T Coaxlink Duo CXP-12	232 g	8.18 oz	122 g	4.30 oz
PC3623-T Coaxlink Quad CXP-12 Value	287 g	10.12 oz	187 g	6.60 oz

Dimensions

Product	Dimensions	
PC1633-T Coaxlink Quad G3	167,65 x 111,15 mm	6,6 x 4,38 in
PC3602-T Coaxlink Octo	167,65 x 111,15 mm	6,6 x 4,38 in
PC3621-LH-T Coaxlink Mono CXP-12-LH	167,65 x 68,9 mm	6,6 x 2,71 in
PC3622-T Coaxlink Duo CXP-12	167,65 x 68,9 mm	6,6 x 2,71 in
PC3623-T Coaxlink Quad CXP-12 Value	167,65 x 111,15 mm	6,6 x 4,38 in

2. Electrical Specification

Electrical specification of the product(s) including: electrical characteristics of all the input/output ports, description of the power distribution, power requirements, etc.

2.1. Camera Interfaces	328
CoaXPress CXP-6 Host Interface	329
CoaXPress CXP-12 Host Interface	331
2.2. PCI Express Interfaces	333
4-lane Rev 3.0 PCIe end-point	334
8-lane Rev 3.0 PCIe end-point	335
2.3. Power	336
Power Distribution Schemes	337
Main Power Input Requirements	344
Auxiliary Power Input	345
I/O Power Output	347
PoCXP Power Output Specifications	349
2.4. I/O Interfaces	351
Differential Input (Version 1)	352
Differential Input (Version 2)	354
TTL Input/Output (Version 1)	356
TTL Input/Output (Version 2)	359
Isolated Input (Version 1)	362
Isolated Input (Version 2)	365
Isolated Input (Version 4)	368
Isolated Output	371

2.1. Camera Interfaces

Electrical specification of the camera interfaces

CoaXPress CXP-6 Host Interface	329
CoaXPress CXP-12 Host Interface	331

CoaXPress CXP-6 Host Interface

Applies to **PC1633-T Coaxlink Quad G3** and **PC3602-T Coaxlink Octo**.



The *CoaXPress CXP-6 Host Interface* implements for each individual CoaXPress connection:

- a high-speed cable receiver
- a low-speed cable driver
- a power transmitting unit (PTU)
- a connector

See also: "Camera and Data Output Connectors" on page 300

It fulfills the electrical specification of the CoaXPress 1.1 standard. Namely:

- The cable receiver requirements for the high-speed connection described in Table 2 of the Annex B of the CoaXPress Standard 1.1
- The cable driver requirements for the low-speed connection described in Table 3 of the Annex B of the CoaXPress Standard 1.1

Cable driver and receiver specification

Parameter	Conditions	Min.	Typ.	Max.	Unit
High-speed receiver bit rate		1.25		6.25	Gbps
Low-speed driver bit rate			20.833		Mbps
Max. cable length	BELDEN 1694 @ 1.25 Gbps	130			m
	BELDEN 1694 @ 2.5 Gbps	110			m
	BELDEN 1694 @ 3.125 Gbps	100			m
	BELDEN 1694 @ 5 Gbps	60			m
	BELDEN 1694 @ 6.25 Gbps	40			m

PTU - Power transmitting unit specification

Parameter	Min.	Typ.	Max.	Unit
DC output voltage	22	24	26	V
Available output power	17 ¹			W
OCP holding current	790			mA
OCP nominal trip current			5	A
Device detection sense resistance		4.7		kΩ

¹ Per connection

The Power Transmitting Unit provides 24 V DC power, over-current protection (OCP) and PoCXP device detection as specified by the CoaXPress Standard.

NOTE: The above specification applies over the whole operating temperature range of the card.

See also: Refer to Power Over CoaXPress in the Functional Guide

CoaXPress CXP-12 Host Interface

Applies to **PC3621-LH-T Coaxlink Mono CXP-12-LH**, **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12 Value**.



The *CoaXPress CXP-12 Host Interface* implements for each individual CoaXPress connection:

- a high-speed cable receiver
- a low-speed cable driver
- a power transmitting unit (PTU)
- a connector

See also: "Camera and Data Output Connectors" on page 300

It fulfills the electrical specification of the CoaXPress 2.0 standard. Namely:

- The cable receiver requirements for the high-speed connection described in Table 2 of the Annex B of the CoaXPress Standard Version 2.0
- The cable driver requirements for the low-speed connection described in Table 3 of the Annex B of the CoaXPress Standard Version 2.0

Cable Driver and Receiver Specification

Parameter	Conditions	Min.	Typ.	Max.	Unit
High-speed receiver bit rate		1.25		12.50	Gbps
Low-speed driver bit rate	1.25 Gbps up to 6.25 Gbps		20.833		Mbps
	10.0 Gbps and 12.5 Gbps		41.666		Mbps
Max. cable length	BELDEN 1694 @ 1.25 Gbps	130			m
	BELDEN 1694 @ 2.5 Gbps	115			m
	BELDEN 1694 @ 3.125 Gbps	100			m
	BELDEN 1694 @ 5 Gbps	80			m
	BELDEN 1694 @ 6.25 Gbps	70			m
	BELDEN 1694 @ 10.0 Gbps	50			m
	BELDEN 1694 @12.5Gbps	40			m

PTU - Power transmitting unit specification

Parameter	Min.	Typ.	Max.	Unit
DC output voltage	22	24	26	V
Available output power ¹	17 ²			W

¹ Per connection

² 25 W for **PC3621-LH Coaxlink Mono CXP-12 LH**

Parameter	Min.	Typ.	Max.	Unit
OCP holding current	790			mA
OCP nominal trip current			5	A
Device detection sense resistance		4.7		k Ω

The Power Transmitting Unit provides 24 V DC power, over-current protection (OCP) and PoCXP device detection as specified by the CoaXPress Standard.

NOTE: The above specification applies over the whole operating temperature range of the card.

See also: Refer to Power Over CoaXPress in the Functional Guide

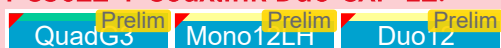
2.2. PCI Express Interfaces

Specification of the PCI Express interfaces

4-lane Rev 3.0 PCIe end-point	334
8-lane Rev 3.0 PCIe end-point	335

4-lane Rev 3.0 PCIe end-point

Applies to **PC1633-T Coaxlink Quad G3**, **PC3621-LH-T Coaxlink Mono CXP-12-LH** and **PC3622-T Coaxlink Duo CXP-12**.



The PCI Express Interface implements a *PCIe end-point* interface and provides *electrical power* to the on-board circuits.

The 4-lane Rev 3.0 PCIe end-point:

- complies with Revision 3.0 of the PCI Express Card Electromechanical specification,
- supports 1-lane, 2-lane, and 4-lane link width,
- supports PCIe Rev 3.0 link speed (8.0 GT/s with 128b/130b coding),
- supports PCIe Rev 2.0 link speed (5.0 GT/s with 8b/10b coding),
- *doesn't support* the PCIe Rev 1.0 link speed (2.5 GT/s with 8b/10b coding),
- supports payload size up to 512 bytes,
- offers the optimal performance when it is configured for 4-lane PCIe Rev 3.0 link speed (8 GT/s).

4-lane Rev 3.0 PCIe end-point to PC memory data transfer performance

Parameter	Conditions	Min.	Typ.	Max.	Unit
Sustainable output data rate	4-lane @ 8 GT/s (PCIe Rev 3.0)		3,350		MB/s
	4-lane @ 5 GT/s (PCIe Rev 2.0)		1,700		MB/s
	2-lane @ 8 GT/s (PCIe Rev 3.0)		1,700		MB/s
	2-lane @ 5 GT/s (PCIe Rev 2.0)		800		MB/s
	1-lane @ 8 GT/s (PCIe Rev 3.0)		800		MB/s

8-lane Rev 3.0 PCIe end-point

Applies to PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value.



The PCI Express Interface implements a *PCIe end-point* interface and provides *electrical power* to the on-board circuits.

The 8-lane Rev 3.0 PCIe end-point:

- complies with Revision 3.0 of the PCI Express Card Electromechanical specification,
- supports 1-lane, 2-lane, 4-lane and 8-lane link width,
- supports PCIe Rev 3.0 link speed (8.0 GT/s with 128b/130b coding),
- supports PCIe Rev 2.0 link speed (5.0 GT/s with 8b/10b coding)
- supports the PCIe Rev 1.0 link speed (2.5 GT/s with 8b/10b coding),
- supports payload size up to 512 bytes,
- offers the optimal performance when it is configured for 8-lane PCIe Rev 3.0 link speed (8 GT/s).

8-lane Rev 3.0 PCIe end-point to PC memory data transfer performance

Parameter	Conditions	Min.	Typ.	Max.	Unit
Sustainable output data rate	8-lane @ 8 GT/s (PCIe Rev 3.0)		6,700		MB/s
	8-lane @ 5 GT/s (PCIe Rev 2.0)		3,400		MB/s
	4-lane @ 8 GT/s (PCIe Rev 3.0)		3,350		MB/s
	4-lane @ 5 GT/s (PCIe Rev 2.0)		1,700		MB/s
	2-lane @ 8 GT/s (PCIe Rev 3.0)		1,700		MB/s
	2-lane @ 5 GT/s (PCIe Rev 2.0)		800		MB/s
	1-lane @ 8 GT/s (PCIe Rev 3.0)		800		MB/s
	1-lane @ 5 GT/s (PCIe Rev 2.0)		400		MB/s

2.3. Power

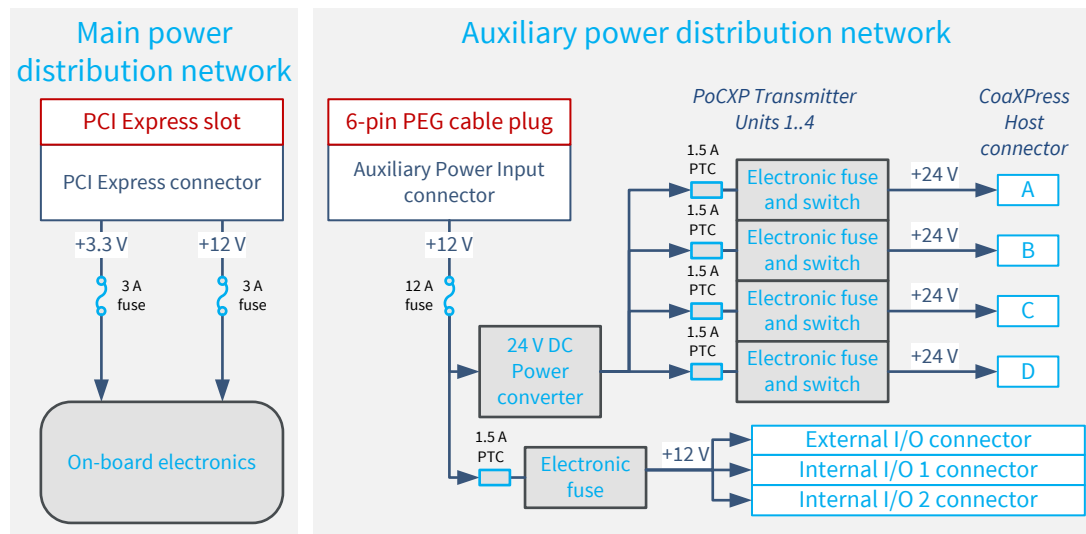
Power requirements and specifications

Power Distribution Schemes	337
Main Power Input Requirements	344
Auxiliary Power Input	345
I/O Power Output	347
PoCXP Power Output Specifications	349

Power Distribution Schemes

PC1633-T Coaxlink Quad G3	338
PC3602-T Coaxlink Octo	339
PC3621-LH-T Coaxlink Mono CXP-12-LH	341
PC3622-T Coaxlink Duo CXP-12	342
PC3623-T Coaxlink Quad CXP-12 Value	343

PC1633-T Coaxlink Quad G3



Main power distribution network

The main power distribution network delivers power to *all the on-board electronic devices* including FPGA, memory chips, CoaXPress transceivers, I/O drivers and receivers, fan motor.

The network is fed by the Host PC motherboard through the +3.3 V and the +12 V power rails of the PCI Express slot connector. Protection fuses inserted at the input side of each power rail prevent potential fire hazards.

The *board status LED* reflects the global status of all the power converters of the main distribution network.

See also: "Main Power Input Requirements" on page 344

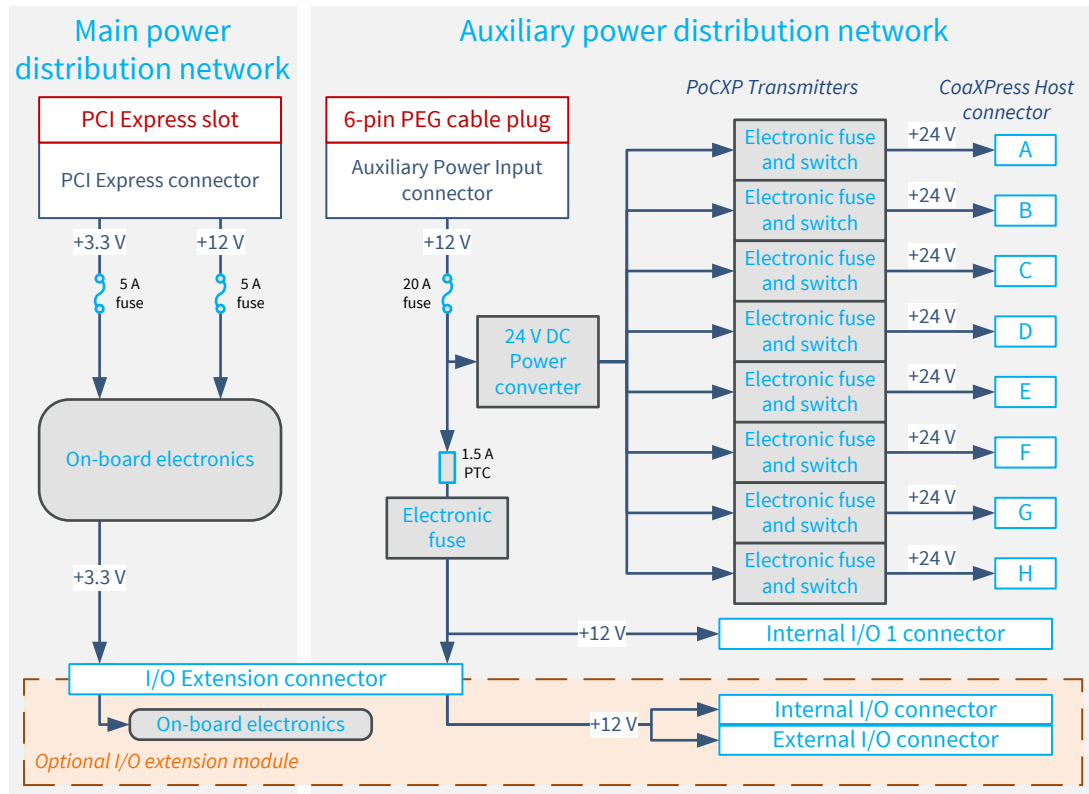
Auxiliary power distribution network

The auxiliary power distribution network delivers power to the external devices including:

- CoaXPress cameras using the PoCXP capability available on all connections of the CoaXPress Host connector
- System devices using the +12 V power output available on all I/O connectors

See also: "Auxiliary Power Input" on page 345 and "PoCXP Power Output Specifications" on page 349

PC3602-T Coaxlink Octo



Main power distribution network

The main power distribution network delivers power to *all the on-board electronic devices* including FPGA, memory chips, CoaXPress transceivers, I/O drivers and receivers, fan motor.

The network is fed by the Host PC motherboard through the +3.3 V and the +12 V power rails of the PCI Express slot connector. Protection fuses inserted at the input side of each power rail prevent potential fire hazards.

The *board status LED* reflects the global status of all the power converters of the main distribution network.

See also: "Main Power Input Requirements" on page 344

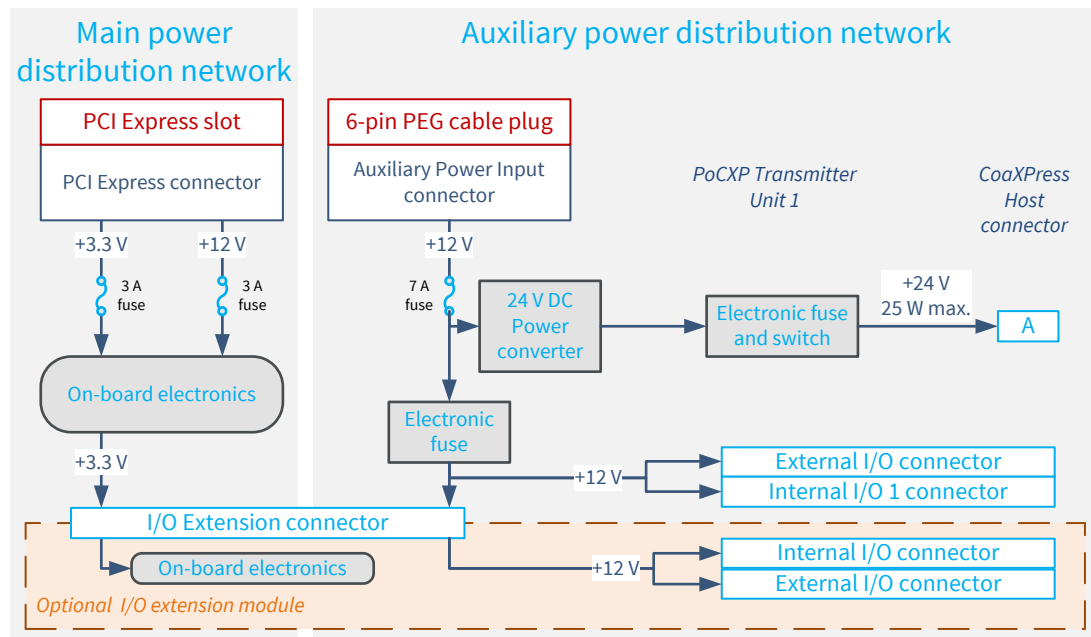
Auxiliary power distribution network

The auxiliary power distribution network delivers power to the external devices including:

- CoaXPress cameras using the PoCXP capability available on all connections of the CoaXPress Host connector
- System devices using the +12 V power output available on all I/O connectors

See also: "Auxiliary Power Input" on page 345 and "PoCXP Power Output Specifications" on page 349

PC3621-LH-T Coaxlink Mono CXP-12-LH



Main power distribution network

The main power distribution network delivers power to *all the on-board electronic devices* including FPGA, memory chips, CoaXPress transceivers, I/O drivers and receivers, fan motor.

The network is fed by the Host PC motherboard through the +3.3 V and the +12 V power rails of the PCI Express slot connector. Protection fuses inserted at the input side of each power rail prevent potential fire hazards.

The *board status LED* reflects the global status of all the power converters of the main distribution network.

See also: "Main Power Input Requirements" on page 344

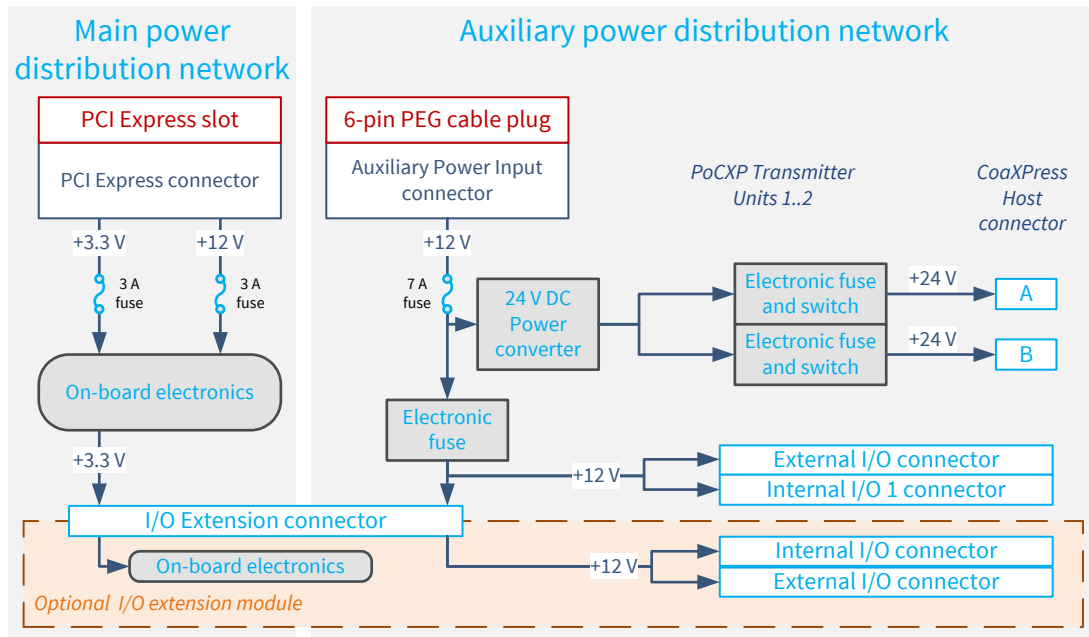
Auxiliary power distribution network

The auxiliary power distribution network delivers power to the external devices including:

- CoaXPress cameras using the PoCXP capability available on all connections of the CoaXPress Host connector
- System devices using the +12 V power output available on all I/O connectors

See also: "Auxiliary Power Input" on page 345 and "PoCXP Power Output Specifications" on page 349

PC3622-T Coaxlink Duo CXP-12



Main power distribution network

The main power distribution network delivers power to *all the on-board electronic devices* including FPGA, memory chips, CoaXPress transceivers, I/O drivers and receivers, fan motor.

The network is fed by the Host PC motherboard through the +3.3 V and the +12 V power rails of the PCI Express slot connector. Protection fuses inserted at the input side of each power rail prevent potential fire hazards.

The *board status LED* reflects the global status of all the power converters of the main distribution network.

See also: "Main Power Input Requirements" on page 344

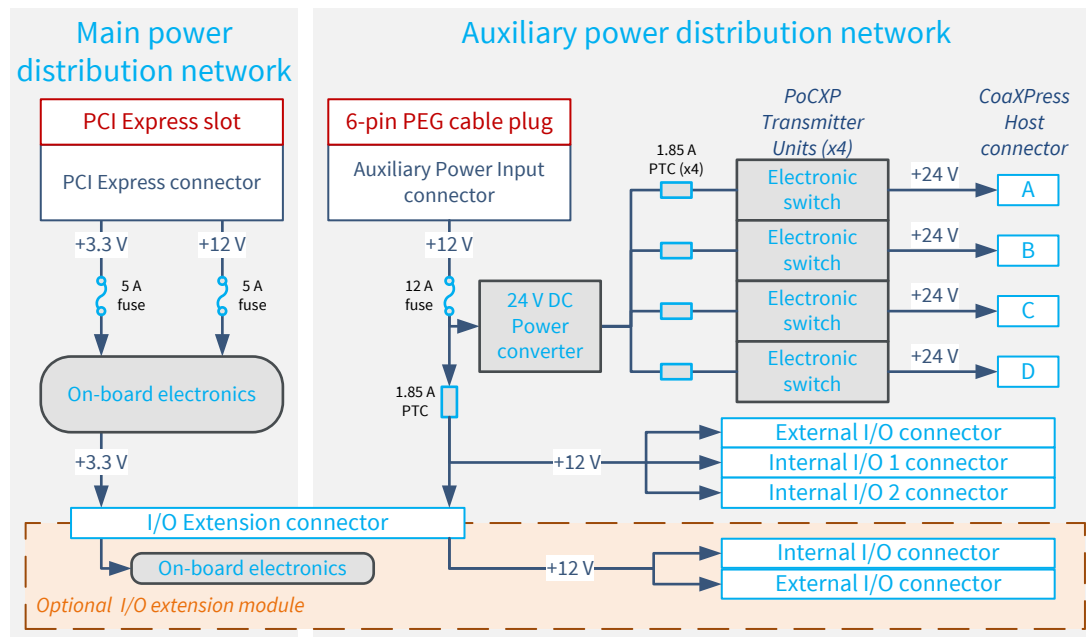
Auxiliary power distribution network

The auxiliary power distribution network delivers power to the external devices including:

- CoaXPress cameras using the PoCXP capability available on all connections of the CoaXPress Host connector
- System devices using the +12 V power output available on all I/O connectors

See also: "Auxiliary Power Input" on page 345 and "PoCXP Power Output Specifications" on page 349

PC3623-T Coaxlink Quad CXP-12 Value



Main power distribution network

The main power distribution network delivers power to *all the on-board electronic devices* including FPGA, memory chips, CoaXPress transceivers, I/O drivers and receivers, fan motor.

The network is fed by the Host PC motherboard through the +3.3 V and the +12 V power rails of the PCI Express slot connector. Protection fuses inserted at the input side of each power rail prevent potential fire hazards.

The *board status LED* reflects the global status of all the power converters of the main distribution network.

See also: "Main Power Input Requirements" on page 344

Auxiliary power distribution network

The auxiliary power distribution network delivers power to the external devices including:

- CoaXPress cameras using the PoCXP capability available on all connections of the CoaXPress Host connector
- System devices using the +12 V power output available on all I/O connectors

See also: "Auxiliary Power Input" on page 345 and "PoCXP Power Output Specifications" on page 349

Main Power Input Requirements

Typical PCI Express power consumption

The following table provides the *typical PCI Express power consumption* for each product when it operates under the following conditions:

- Acquiring image data using all CoaXPress Host Interface connections operating at their maximum speed
- Delivering image data on the PCI Express configured for the largest link width and the highest link speed
- Operating @25°C [77 °F] ambient temperature and nominal supply voltages

Product	+12 V	+3.3 V	Total	Units
PC1633-T Coaxlink Quad G3	13	3.8	16.8	W
PC3602-T Coaxlink Octo	11.8	4.2	16	W
PC3621-LH-T Coaxlink Mono CXP-12-LH	8.5	3	11.5	W
PC3622-T Coaxlink Duo CXP-12	10.5	4.3	14.8	W
PC3623-T Coaxlink Quad CXP-12 Value	13.4	3.3	16.7	W



WARNING

The above table provides typical power consumption to help the system integrator with the sizing of the Host PC power supply.

Voltage requirements

Parameter	Min.	Typ.	Max.	Units
+3.3 V voltage	3.0	3.3	3.6	V
+12 V voltage	11.0	12.0	13.0	V

Auxiliary Power Input

Auxiliary power input requirements

The following table provides the 'worst case' auxiliary power consumption for each product:

- The *I/O power* column specifies the maximum required input power dedicated to external I/O powering.
- The *Camera power* column specifies the maximum required input power dedicated to camera powering. This number takes care of the power conversion efficiency.

Product	Connector	I/O power [W]	Camera power [W]	Total AUX power [W]
PC1633-T Coaxlink Quad G3	PEG	12	4 x 19	88
PC3602-T Coaxlink Octo	PEG*	12	8 x 19	164
PC3621-LH-T Coaxlink Mono CXP-12-LH	PEG*	12	1 x 28	40
PC3622-T Coaxlink Duo CXP-12	PEG*	12	2 x 19	50
PC3623-T Coaxlink Quad CXP-12 Value	PEG*	12	4 x 19	88

PEG*: PEG connectors without *SenseIN* input!

Requirements for PEG connector (PCIe products)

A +12V power source must be connected to the AUXILIARY POWER INPUT connector using the appropriate PEG cable:

- When the *total AUX power* required by your application exceeds 75 W, it is mandatory to use an 8-pin PEG cable terminated with a 2x3 plug + 2x1 plug.

NOTE: Only the 2x3 plug from the PEG cable must be connected to the AUXILIARY POWER INPUT connector.

- Otherwise, it is allowed to use an 6-pin PEG cable terminated with a 2x3 plug



A protection fuse inserted at the input side prevents potential fire hazards.

Parameter	Min.	Typ.	Max.	Units
Voltage	11.0	12.0	13.0	V



WARNING

On the card side of the power cable, check carefully the voltage on each pin of the 6-pin PEG connector plug before insertion!

See also: Avoid Mixing Power Supply Cables

Monitoring the auxiliary power input

The **AuxiliaryPower12VInput** GenApi feature of the Interface module reports the status of the "12V Auxiliary Power Input" measured after the input fuse.

The **AuxiliaryPowerInput** GenApi feature of the Interface module reports the status of the auxiliary power input cable connection. This feature is only available for products having a PEG connector with a *SenseIN* input.



NOTE

The *SenseIN* input of the PEG connector is used for power source cable presence detection. It should be grounded at the power supply level.

I/O Power Output

I/O Power Output (Version 1)

Applies to **PC1633-T Coaxlink Quad G3**, **PC3602-T Coaxlink Octo**, **PC3621-LH-T Coaxlink Mono CXP-12-LH** and **PC3622-T Coaxlink Duo CXP-12**.



Description

A non-isolated +12 V power output is available on every I/O connector.

The power originates from an external 12 V power supply plugged into the Auxiliary Power Input connector. It is distributed from a common electronic fuse to all the I/O connectors.

The electronic fuse provides the following protections:

- Limits the inrush current during power on sequence
- Protects the frame grabber and the power source against overload
- Protects the frame grabber and the power source against short-circuits.

The sum of the load currents drawn from all the 12 V outputs of the I/O connectors must be lower or equal to the specified maximum output current.

Specification

Parameter	Conditions	Min.	Typ.	Max.	Units
Aggregated output current	Operating temperature range			1.0	A
Voltage drop across the electronic fuse	Max. output current			0.2	V

NOTE: The above specification applies over the whole operating temperature range of the frame grabber.

I/O Power Output (Version 2)

Applies to PC3623-T Coaxlink Quad CXP-12 Value.

Prelim
Value 12

A non-isolated +12 V power output is available on every I/O connector.

The power originates from an external 12 V power supply plugged into the Auxiliary Power Input connector. It is distributed from a common resettable fuse to all the I/O connectors.

The resettable fuse provides the following protections:

- Limits the inrush current during power on sequence
- Protects the frame grabber and the power source against overload
- Protects the frame grabber and the power source against short-circuits.
- The resettable fuse is a positive temperature coefficient device 1.85 A PTC.

The sum of the load currents drawn from all the 12 V outputs of the I/O connectors must be lower or equal to the specified maximum output current.

Specification

Parameter	Conditions	Min.	Typ.	Max.	Units
Aggregated output current	Operating temperature range			1.0	A
Voltage drop across the electronic fuse	Max. output current			0.2	V

NOTE: The above specification applies over the whole operating temperature range of the frame grabber.

PoCXP Power Output Specifications

PoCXP power output

Applies to **PC1633-T Coaxlink Quad G3** and **PC3602-T Coaxlink Octo**.



Description

A 24-volt DC power converter provides power to each camera connection through a PoCXP transmitter unit.

Each PoCXP transmitter unit implements an over-current-protection – OCP– that prevents potential fire hazards.

Specification

Parameter	Conditions	Min.	Typ.	Max.	Units
Nominal output power (per connector)	Operating temperature range			17	W
Startup time			50		μs
Startup current	100 μs pulse		10		A

PoCXP power output - 25W

Applies to **PC3621-LH-T Coaxlink Mono CXP-12-LH**.



Description

A 24-volt DC power converter provides power to each camera connection through a PoCXP transmitter unit.

Each PoCXP transmitter unit implements an over-current-protection – OCP– that prevents potential fire hazards.

Specification

Parameter	Conditions	Min.	Typ.	Max.	Units
Nominal output power (per connector)	Operating temperature range			25	W
Startup time			50		μs
Startup current	100 μs pulse		10		A

PoCXP power output (Version 3)

Applies to **PC3622-T Coaxlink Duo CXP-12** and **PC3623-T Coaxlink Quad CXP-12** Value.

Prelim

Duo 12

Prelim

Value 12


Description

A 24-volt DC power converter provides power to each camera connection through a PoCXP transmitter unit.


Each PoCXP transmitter unit implements an over-current-protection – OCP– that prevents potential fire hazards.

Specification

Parameter	Conditions	Min.	Typ.	Max.	Units
Nominal output power (per connector)	Operating temperature range			17	W
Startup time			TBD		µs
Startup current	100 µs pulse		TBD		A

**WARNING**

- The OCP is sized to sustain the nominal output power over the whole operating temperature range without tripping.
- Extracting more than the nominal output power and/or operating the Coaxlink card above the operating temperature range is prohibited since it may induce unexpected PTC trips.

**TIP**

The "CoaXPress LED Lamps" on page 321 reflect the state of each CoaXPress host connection.

**TIP**

The CoaXPress LED Lamps for PC3629 reflect the state of each CoaXPress host connection.

See also: Power Over CoaXPress in the Functional Guide

2.4. I/O Interfaces

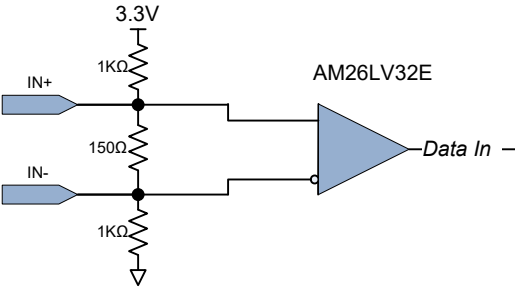
Electrical specification of the I/O interfaces

Differential Input (Version 1)	352
Differential Input (Version 2)	354
TTL Input/Output (Version 1)	356
TTL Input/Output (Version 2)	359
Isolated Input (Version 1)	362
Isolated Input (Version 2)	365
Isolated Input (Version 4)	368
Isolated Output	371

Differential Input (Version 1)

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo and PC3623-T Coaxlink Quad CXP-12 Value.

QuadG3 Prelim Octo Prelim Value12 Prelim



Differential Input Simplified Schematic

The receiver complies with the ANSI/TIA/EIA-422B specification.

DC Characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Common mode voltage		-7		+7	V
Differential sensitivity				200	mV
Input impedance			120		Ohm
ESD protection	Human Body Model (HBM)	15			kV
	Contact discharge	8			kV
	Air gap discharge	15			kV

AC characteristics

Parameter	Min.	Typ.	Max.	Units
Pulse width	100			ns
Pulse rate	0		5	MHz
10%-90% rise/fall time			1	µs

Logical map

The state of the port is reported as follows:

Relative V+/V- voltage	Logical State
V+ > V-	HIGH
V+ < V-	LOW
Unconnected input	HIGH

Compatible drivers

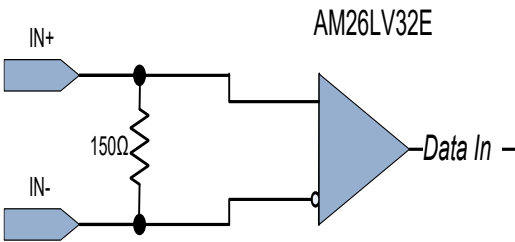
The following drivers are compatible with the high-speed differential input ports:

- RS-422/RS-485 differential line drivers
- Complementary TTL drivers

Differential Input (Version 2)

Applies to **PC3621-LH-T Coaxlink Mono CXP-12-LH** and **PC3622-T Coaxlink Duo CXP-12**.

Mono12LH Prelim **Duo12** Prelim



Differential Input Simplified Schematic

The receiver complies with the ANSI/TIA/EIA-422B specification.

DC Characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Common mode voltage		-7		+7	V
Differential sensitivity				200	mV
Input impedance			150		Ohm
ESD protection	Human Body Model (HBM)	15			kV
	Contact discharge	8			kV
	Air gap discharge	15			kV

AC characteristics

Parameter	Min.	Typ.	Max.	Units
Pulse width	100			ns
Pulse rate	0		5	MHz
10%-90% rise/fall time			1	μs

Logical map

The state of the port is reported as follows:

Relative V+/V- voltage	Logical State
V+ > V-	HIGH
V+ < V-	LOW
Unconnected input	Unknown

Compatible drivers

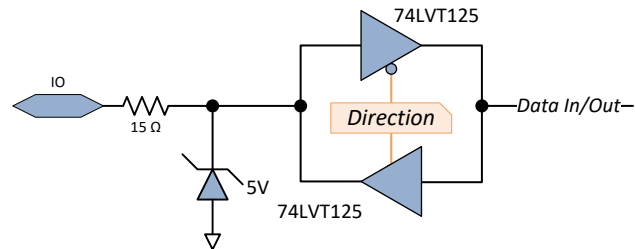
The following drivers are compatible with the high-speed differential input ports:

- RS-422/RS-485 differential line drivers
- Complementary TTL drivers

TTL Input/Output (Version 1)

Applies to PC1633-T Coaxlink Quad G3.

Prelim
QuadG3



TTL Input/Output Simplified schematic

The port implements a 3.3 V LVTTTL driver and a 5 V-compliant 3.3 V LVTTTL receiver.

DC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
ESD protection	Human Body Model (HBM)	2			kV



NOTE

The I/O port includes a latch-up protection.

Driver

Parameter	Conditions	Min.	Typ.	Max.	Units
Low-level output current				64	mA
	@ 8 mA		0.34	0.36	V
Low-level output voltage	@ 16 mA		0.48	0.55	V
	@ 32 mA		0.78	0.81	V
	@ 64 mA		1.34	1.36	V
High-level output current				-32	mA
	@-8 mA; (1)	2.60	3.00		V
High-level output voltage	@-16 mA; (1)	2.20	2.70		V
	@-32 mA; (1)	1.75	2.20		V
ESD protection	Human Body Model (HBM)	2			kV

Condition (1): 300 Ohms line termination resistor to GND.

Receiver

Parameter	Conditions	Min.	Typ.	Max.	Units
Absolute maximum voltage rating		0		5	V

AC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Pulse width		100			ns
Pulse rate		0		5	MHz
10%-90% rise/fall time	(1)		10	20	ns

Condition (1): Short cable (1 m) and a 300 Ohms line termination resistor to GND.

Logical Map

The state of the port is reported as follows:

Input voltage	Logical State
$V_{IN} > 2.0\text{ V}$	HIGH
$V_{IN} < 0.8\text{ V}$	LOW
Unconnected input port	<i>Undetermined</i>

Compatible sources

Sources with the following drivers are compatible:

- LVTTTL (3.3 V low-voltage TTL)
- TTL (5 V TTL)
- CMOS (5 V CMOS)

Compatible loads

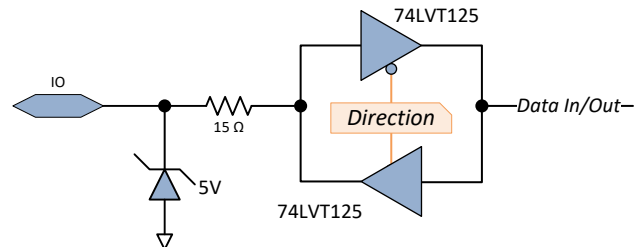
Loads with the following receivers are compatible:

- LVTTTL (3.3 V low-voltage TTL)
- TTL (5 V TTL)

TTL Input/Output (Version 2)

Applies to PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

Octo Prelim Mono12LH Prelim Duo12 Prelim Value12 Prelim



TTL Input/Output Simplified schematic

The port implements a 3.3 V LVTTTL driver and a 5 V-compliant 3.3 V LVTTTL receiver.

DC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
ESD protection	Human Body Model (HBM)	2			kV



NOTE

The I/O port includes a latch-up protection.

Driver

Parameter	Conditions	Min.	Typ.	Max.	Units
Low-level output current				64	mA
	@ 8 mA		0.34	0.36	V
	@ 16 mA		0.48	0.55	V
Low-level output voltage	@ 32 mA		0.78	0.81	V
	@ 64 mA		1.34	1.36	V
High-level output current				-32	mA
	@-8 mA; (1)	2.60	3.00		V
High-level output voltage	@-16 mA; (1)	2.20	2.70		V
	@-32 mA; (1)	1.75	2.20		V
ESD protection	Human Body Model (HBM)	2			kV

Condition (1): 300 Ohms line termination resistor to GND.

Receiver

Parameter	Conditions	Min.	Typ.	Max.	Units
Absolute maximum voltage rating		0		5	V

AC characteristics

Driver

Parameter	Conditions	Min.	Typ.	Max.	Units
Pulse width		100			ns
Pulse rate		0		5	MHz
10%-90% rise time			8		ns
10%-90% fall time			7.5		ns

Receiver

Parameter	Conditions	Min.	Typ.	Max.	Units
Pulse width		100			ns
Pulse rate		0		5	MHz

Logical Map

The state of the port is reported as follows:

Input voltage	Logical State
VIN > 2.0 V	HIGH
VIN < 0.8 V	LOW
Unconnected input port	<i>Undetermined</i>

Compatible sources

Sources with the following drivers are compatible:

- ❑ LVTTTL (3.3 V low-voltage TTL)
- ❑ TTL (5 V TTL)
- ❑ CMOS (5 V CMOS)

Compatible loads

Loads with the following receivers are compatible:

- ❑ LVTTTL (3.3 V low-voltage TTL)
- ❑ TTL (5 V TTL)

Isolated Input (Version 1)

Applies to **PC1633-T Coaxlink Quad G3** and **PC3602-T Coaxlink Octo**.

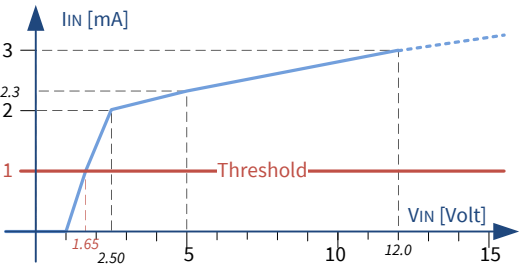
QuadG3

Prelim

Octo

Prelim

Isolated current-sense input with wide voltage input range up to 30V, compatible with totem-pole LVTTTL, TTL, 5V CMOS drivers, RS-422 differential line drivers, potential free contacts, solid-state relays and opto-couplers



Simplified schematic Input Current vs. Input Voltage Characteristics

DC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Differential voltage		-30		+30	V
Input current threshold			1		mA
Differential voltage	@1 mA	1.5	1.65	1.9	V
Input current	@(VIN+ - VIN-) < 1 V			10	µA
	@(VIN+ - VIN-) = 1.65 V		1		mA
	@(VIN+ - VIN-) = 2.5 V		2		mA
	@(VIN+ - VIN-) = 5 V		2.3		mA
	@(VIN+ - VIN-) = 12 V		3		mA
	@(VIN+ - VIN-) = 30 V			5	mA

AC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Positive pulse width		10			µs
Negative pulse width		10			µs
Pulse rate		0		50	kHz
Turn-ON delay ¹	30°C; 50 kHz; 2 V square wave signal		2.1		µs
Turn-OFF delay ²	LineFilterStrength = Lowest (LineFilterDelay = 500 ns)		4.5		µs



NOTE

1. The "Turn-ON" delay is defined as the time difference between a transition of state at the input that turns ON the opto-coupler and the subsequent transition in the FPGA.
2. The "Turn-OFF" delay is defined as the time difference between a transition of state at the input that turns OFF the opto-coupler and the subsequent transition in the FPGA.
3. The response of the Isolated I/Os vary in function of the activation frequency. It appears that the variation is caused by a variation on temperature: Higher is the frequency, higher is the temperature, which increases the propagation delay.

These delays include the delay introduced by the digital line filter controlled by the **LineFilterStrength** GenApi feature!

Isolation characteristics

Parameter	Value
Isolation grade	Functional
Max. DC voltage	250 V
Max. AC voltage	170 V _{RMS}



NOTE

The functional isolation is only for the circuit technical protection. It does not provide an isolation that can protect a human being from electrical shock!

Logical map

The state of the port is reported as follows:

Input current	Logical State
$I_{IN} > 1 \text{ mA}$	HIGH
$I_{IN} < 1 \text{ mA}$	LOW
Unconnected input port	LOW

Compatible drivers

The following drivers are compatible with this version of the isolated current-sense inputs:

- Totem-pole LVTTTL, TTL, 5 V CMOS drivers
- RS-422 Differential line drivers
- Potential free contact, solid-state relay, or opto-isolators
- 12 V and 24 V signaling voltages are also accepted



NOTE

- The +12 V power supply on the I/O connector(s) can be used for powering drivers requiring a power supply.
- No external resistors are required. However, to obtain the best noise immunity with 12 V and 24 V signaling, it is recommended to insert a series resistor in the circuit. The recommended resistor values are: 4.7k Ohms for 12 V signaling and 10k Ohms for 24 V signaling.

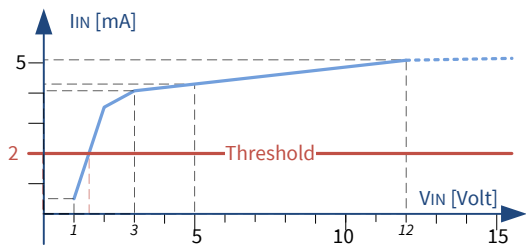
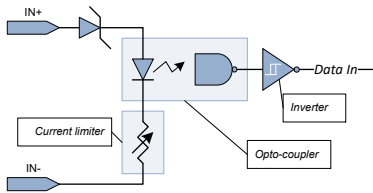
See also: [Using Isolated Inputs](#)

Isolated Input (Version 2)

Applies to PC3623-T Coaxlink Quad CXP-12 Value.

Prelim
Value 12

Isolated current-sense input with wide voltage input range up to 30V, compatible with totem-pole (push-pull) HTL drivers, 5V TTL/RS-422 differential line drivers, 5V CMOS drivers, potential free contacts, solid-state relays and opto-couplers



Simplified schematic

Input Current vs. Input Voltage Characteristics

DC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Differential voltage		-30		+30	V
Input current threshold			2	3	mA
Differential voltage	@3 mA		2.2		V
	@(VIN+ - VIN-) < 1 V		0.5		mA
	@(VIN+ - VIN-) = 3 V		4.1		mA
Input current	@(VIN+ - VIN-) = 5 V		4.4		mA
	@(VIN+ - VIN-) = 12 V		5.1		mA
	@(VIN+ - VIN-) = 30 V		6.8		mA

AC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Positive pulse width		2.5			μs
Negative pulse width		2.5			μs
Pulse rate		0		200	kHz
Turn-ON delay ¹	25 °C; 100 kHz; 5 V square signal <i>LineFilterStrength</i> = Lowest		800		ns
Turn-OFF delay ²	(<i>LineFilterDelay</i> = 500 ns)		1350		ns



NOTE

1. The "Turn-ON" delay is defined as the time difference between a transition of state at the input that turns ON the opto-coupler and the subsequent transition in the FPGA.
2. The "Turn-OFF" delay is defined as the time difference between a transition of state at the input that turns OFF the opto-coupler and the subsequent transition in the FPGA.
3. The response of the Isolated I/Os vary in function of the activation frequency. It appears that the variation is caused by a variation on temperature: Higher is the frequency, higher is the temperature, witch increases the propagation delay.

These delays include the delay introduced by the digital line filter controlled by the *LineFilterStrength* GenApi feature!

Isolation characteristics

Parameter	Value
Isolation grade	Functional
Max. DC voltage	250 V
Max. AC voltage	170 V _{RMS}



NOTE

The functional isolation is only for the circuit technical protection. It does not provide an isolation that can protect a human being from electrical shock!

Logical map

The state of the port is reported as follows:

Input current	Logical State
$I_{IN} \geq 3 \text{ mA}$	HIGH
$I_{IN} < 0.5 \text{ mA}$	LOW

Input current	Logical State
Unconnected input port	LOW

Compatible drivers

The following drivers are compatible with this version of the isolated current-sense inputs:

- Totem-pole (push-pull) HTL drivers, 5V TTL/RS-422 differential line drivers, 5 V CMOS drivers
- Potential free contact, solid-state relay, or opto-isolators
- 12 V and 24 V signaling voltages are also accepted



NOTE

- The +12 V power supply on the I/O connector(s) can be used for powering drivers requiring a power supply.
- No external resistors are required. However, to obtain the best noise immunity with 12 V and 24 V signaling, it is recommended to insert a series resistor in the circuit. The recommended resistor values are: 4.7k Ohms for 12 V signaling and 10k Ohms for 24 V signaling.

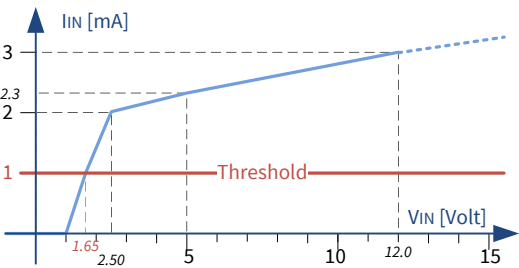
See also: [Using Isolated Inputs](#)

Isolated Input (Version 4)

Applies to **PC3621-LH-T Coaxlink Mono CXP-12-LH** and **PC3622-T Coaxlink Duo CXP-12**.

Mono12LH Prelim **Duo12** Prelim

Isolated current-sense input with wide voltage input range up to 30V, compatible with totem-pole LVTTTL, TTL, 5V CMOS drivers, RS-422 differential line drivers, potential free contacts, solid-state relays and opto-couplers



Simplified schematic

Input Current vs. Input Voltage Characteristics

DC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Differential voltage		-30		+30	V
Input current threshold			1		mA
Differential voltage	@1 mA	1.5	1.65	1.9	V
Input current	@(VIN+ - VIN-) < 1 V			10	μA
	@(VIN+ - VIN-) = 1.65 V		1		mA
	@(VIN+ - VIN-) = 2.5 V		2		mA
	@(VIN+ - VIN-) = 5 V		2.3		mA
	@(VIN+ - VIN-) = 12 V		3		mA
	@(VIN+ - VIN-) = 30 V			5	mA

AC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Positive pulse width		10			μs
Negative pulse width		10			
Pulse rate		0		50	kHz
Turn-ON delay ¹	30°C; 50 kHz; 2 V square wave signal LineFilterStrength = Lowest		6.9		μs
Turn-OFF delay ²	(LineFilterDelay = 500 ns)		9.8		μs



NOTE

1. The "Turn-ON" delay is defined as the time difference between a transition of state at the input that turns ON the opto-coupler and the subsequent transition in the FPGA.
2. The "Turn-OFF" delay is defined as the time difference between a transition of state at the input that turns OFF the opto-coupler and the subsequent transition in the FPGA.
3. The response of the Isolated I/Os vary in function of the activation frequency. It appears that the variation is caused by a variation on temperature: Higher is the frequency, higher is the temperature, witch increases the propagation delay.

These delays include the delay introduced by the digital line filter controlled by the **LineFilterStrength** GenApi feature!

Isolation characteristics

Parameter	Value
Isolation grade	Functional
Max. DC voltage	250 V
Max. AC voltage	170 V _{RMS}



NOTE

The functional isolation is only for the circuit technical protection. It does not provide an isolation that can protect a human being from electrical shock!

Logical map

The state of the port is reported as follows:

Input current	Logical State
$I_{IN} > 1 \text{ mA}$	HIGH
$I_{IN} < 1 \text{ mA}$	LOW

Input current	Logical State
Unconnected input port	LOW

Compatible drivers

The following drivers are compatible with this version of the isolated current-sense inputs:

- Totem-pole LVTTTL, TTL, 5 V CMOS drivers
- RS-422 Differential line drivers
- Potential free contact, solid-state relay, or opto-isolators
- 12 V and 24 V signaling voltages are also accepted



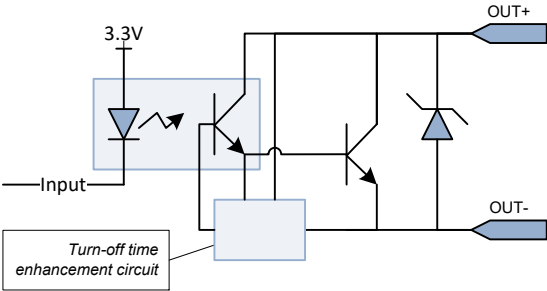
NOTE

- The +12 V power supply on the I/O connector(s) can be used for powering drivers requiring a power supply.
- No external resistors are required. However, to obtain the best noise immunity with 12 V and 24 V signaling, it is recommended to insert a series resistor in the circuit. The recommended resistor values are: 4.7k Ohms for 12 V signaling and 10k Ohms for 24 V signaling.

See also: [Using Isolated Inputs](#)

Isolated Output

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.



Isolated Output Simplified schematic

The output port implements an isolated contact output.

DC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Current				100	mA
	Open state	-30		30	V
Differential voltage	Closed state @ 1 mA			0.4	V
	Closed state @ 100 mA			1.0	V



NOTE

- The output port in the closed state has no current limiter, the user circuit must be designed to avoid excessive currents that could destroy the output port.
- The output port remains in the OFF-state until it is under control of the application.

AC characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Pulse rate		0		100	kHz
Turn-on time	See "Test conditions" below			5	µs
Turn-off time				5	µs

Test conditions

- LVTTTL signaling (0.4V-2.4V) at IOUT output. Tr/Tf is measured between 0.4V & 2.4V.
- Signal span of 5V. The receiver is supposed to be TTL 5V Compliant (like, for instance, our TTLIO). The span of 5V is achieved by a pull-up/pull-down ratio of about 1.2 (for instance 220R/180R or 470R/390R) supplied by 12VIO.
- R-R divider must be strong enough to maintain good fall time (Rdown must be low enough). A R-R divider current of 10 mA is the minimum to get good results.

Performance above 70 kHz

Inter-symbol Interface (ISI) appears for pulse rates higher than 70 KHz.

The ISI adds an extra delay (i.e. jitter) of about 0.5 µsec at 100kHz.

NOTE: This jitter on the IOUT output delay parameter can be limiting in some applications.

Typical switching performance @ 25°C

Current [mA]	Turn ON time [µs]	Turn OFF time [µs]
0.5	2.0	4.8
1.0	2.0	3.9
4.0	2.2	3.3
10	2.3	2.7
40	2.3	2.7
100	2.3	2.7

Isolation characteristics

Parameter	Value
Isolation grade	Functional
Max. DC voltage	250 V
Max. AC voltage	170 V _{RMS}



NOTE

The functional isolation is only for the circuit technical protection. It does not provide an isolation that can protect a human being from electrical shock!

Logical map

The state of the output port is determined as follows:

Logical State	Output port state
HIGH	The contact switch is closed (ON)
LOW	The contact switch is open (OFF)

Compatible loads

The following loads are compatible with the isolated contact output ports:

- Any load within the 30 V / 100 mA envelope is accepted. The power originates from an external power source or alternatively from the power delivered through the 12 V and GND pins of the I/O connectors.

3. Environmental Specification

Environmental specification of the product(s) including: climatic requirements, electromagnetic standards compliance statements, safety standards compliance statements, etc.

3.1. Storage Conditions 375

3.2. Operating Conditions 376

3.3. Temperature Monitor 377

3.4. Thermal Design Data 380

3.5. Compliance Statements 381

3.1. Storage Conditions

Standard (-20°C/+70°C) storage range

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

QuadG3

Octo

Mono12LH

Duo12

Value12

Parameter	Conditions	Min	Max	Units
Ambient air temperature		-20 [-4]	70 [158]	°C [°F]
Ambient air humidity	Non-condensing	10	90	% RH

3.2. Operating Conditions

Standard operating conditions (0°C ~ 55°C) — 80°C max FPGA temperature

Applies to PC1633-T Coaxlink Quad G3 and PC3623-T Coaxlink Quad CXP-12 Value.



Parameter	Conditions	Min	Max	Units
FPGA die temperature			80 [176]	°C [°F]
Ambient air temperature		0 [32]	55 [131]	°C [°F]
Ambient air humidity	Non-condensing	10	90	% RH



WARNING

- The thermal design of the host PC must ensure that, at any time, the FPGA die temperature never exceeds the recommended limit.
- Exceeding the upper limit of the FPGA die temperature can permanently damage the card.

3.3. Temperature Monitor

Monitoring the FPGA temperature

Temperature sensor

All FPGA based products embed a temperature sensor located inside the FPGA, on the die.

To read the FPGA die temperature expressed in °C, the application must :

1. Set the **TemperatureSensorSelector** feature of the Interface Module to **Grabber**, and then ...
2. Get the **Temperature** feature value of the Interface Module.



WARNING

The user application is invited to check regularly the FPGA die temperature to ensure that the board operates within the specified operating limits.

Dual-threshold (85°C/100°C) temperature detection with automatic acquisition stop

Applies to **PC3621-LH-T Coaxlink Mono CXP-12-LH** and **PC3622-T Coaxlink Duo CXP-12**.



The above listed frame grabbers implement a dual-threshold excessive FPGA die temperature detection and an automatic acquisition stopping service.

Warning threshold (85 °C) - Notify

When the measured FPGA die temperature reaches 87°C, the frame grabber posts a **FPGA temperature is too high** Memento message.

The message is sent repeatedly every second until the measured temperature decreases below 83°C or increases above 103°C.



WARNING

- Operation is still possible but is not recommended!
- When such event occurs, the user is invited to check and, possibly, improve the card cooling in the host PC!

Error threshold (100 °C) - Notify - Stop acquisition

When the measured FPGA die temperature reaches 103°C, the frame grabber stops image acquisition and posts a **FPGA temperature is too high; operation may stop to prevent damaging the card** Memento message.

The message is sent repeatedly every second until the measured temperature decreases below 97°C.



TIP

Stopping the acquisition reduces significantly the heat production of the FPGA. This action is aimed to reduce the die temperature, to prevent the application against unexpected FPGA behavior and to prevent damaging the card.



WARNING

- Random errors could occur in the FPGA if its core temperature becomes excessive!
- When such event occurs, the user must immediately shut down the system and revise the card cooling in the host PC before restarting!

Dual-threshold (85°C/100°C) temperature detection

Applies to **PC3623-T Coaxlink Quad CXP-12 Value**.

 **Value 12**

The above listed frame grabbers implement a dual-threshold excessive FPGA die temperature detection.

Warning threshold (85 °C) - Notify

When the measured FPGA die temperature reaches 87°C, the frame grabber posts a **FPGA temperature is too high** Memento message.

The message is sent repeatedly every second until the measured temperature decreases below 83°C or increases above 103°C.



WARNING

- Operation is still possible but is not recommended!
- When such event occurs, the user is invited to check and, possibly, improve the card cooling in the host PC!

Error threshold (100 °C) - Notify

When the measured FPGA die temperature reaches 103°C, the frame grabber posts a **FPGA temperature is too high; operation may stop to prevent damaging the card** Memento message.

The message is sent repeatedly every second until the measured temperature decreases below 97°C.



TIP

Stopping the acquisition reduces significantly the heat production of the FPGA. This action is aimed to reduce the die temperature, to prevent the application against unexpected FPGA behavior and to prevent damaging the card.



WARNING

- Random errors could occur in the FPGA if its core temperature becomes excessive!
- When such event occurs, the user must immediately shut down the system and revise the card cooling in the host PC before restarting!

3.4. Thermal Design Data

Main contributors

The main heat contributors of **Coaxlink frame grabbers** are:

- 1. The electronic devices and power converters of the *main power distribution* network.
- 2. The losses of the PoCXP 12V-24 V power converter of the *auxiliary* power distribution network. The actual contribution depends on the effectively delivered PoCXP power!

NOTE: The I/O powering contribution is not significant!

Generated heat power estimation and cooling method

Requirements for air-cooled products with fan

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.

QuadG3	Octo	Duo12	Value12
--------	------	-------	---------

The heat is dissipated into the ambient air inside the Host PC. The heat exchange is facilitated by a heat sink and a fan mounted on the FPGA (the component having the largest heat source).

The thermal design must ensure sufficient air flow along both sides to keep the FPGA die temperature below the upper limit of the allowed temperature range. The application is responsible for regularly checking the temperature and for taking the appropriate action in case of excessive temperature.

The heat is dissipated into the ambient air surrounding the appliance. The heat exchange is facilitated by a heat sink and a fan mounted on the FPGA (the component having the largest heat source).

The thermal design must ensure sufficient air flow to keep the FPGA die temperature below the upper limit of the allowed temperature range. The application is responsible for regularly checking the temperature and for taking the appropriate action in case of excessive temperature.

3.5. Compliance Statements

EMC compliance statements for Europe and Great Britain



Notice for Europe

This product is in conformity with the Council Directive 2014/30/EU



Notice for Great Britain

This product is in conformity with Electromagnetic Compatibility Regulations 2016

EN55022/32 Class A emission | EN55024/35 immunity

Applies to **PC3602-T Coaxlink Octo**.



This piece of equipment has been tested and found to comply with:

- Class A EN 55022 / CISPR 22 or EN 55032 / CISPR 32 electromagnetic emission requirements for information technology equipment
- EN 55024 / CISPR 24 or EN 55035 / CISPR 35 electromagnetic immunity requirements for information technology equipment

This product has been tested in typical class A compliant host systems. It is assumed that this product will also achieve compliance in any class A compliant unit.

To meet EC requirements, shielded cables must be used to connect a peripheral to the product.

EN55022/32 Class B emission | EN55024/35 immunity

Applies to **PC1633-T Coaxlink Quad G3, PC3621-LH-T Coaxlink Mono CXP-12-LH and PC3622-T Coaxlink Duo CXP-12**.



This piece of equipment has been tested and found to comply with:

- Class B EN 55022 / CISPR 22 or EN 55032 / CISPR 32 electromagnetic emission requirements for information technology equipment
- EN 55024 / CISPR 24 or EN 55035 / CISPR 35 electromagnetic immunity requirements for information technology equipment

This product has been tested in typical class A and class B compliant host systems. It is assumed that this product will also achieve compliance in any class A or class B compliant unit.

To meet EC requirements, shielded cables must be used to connect a peripheral to the product.

EN55022/32 Class B emission | EN55024/35 EN61000-6-2 immunity

Applies to **PC3623-T Coaxlink Quad CXP-12 Value**.

Prelim
Value 12

This piece of equipment has been tested and found to comply with:

- Class B EN 55022 / CISPR 22 or EN 55032 / CISPR 32 electromagnetic emission requirements for information technology equipment
- EN 55024 / CISPR 24 or EN 55035 / CISPR 35 electromagnetic immunity requirements for information technology equipment
- EN 61000-6-2 Immunity standard for industrial environments

This product has been tested in typical class A and class B compliant host systems. It is assumed that this product will also achieve compliance in any class A or class B compliant unit.

To meet EC requirements, shielded cables must be used to connect a peripheral to the product.

EMC compliance statement for USA



Notice for USA

Compliance Information Statement (Declaration of Conformity Procedure) DoC FCC
Part 15

Class B emission

Applies to **PC1633-T Coaxlink Quad G3, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value**.

Prelim **Prelim** **Prelim** **Prelim**
Quad G3 Mono 12 LH Duo 12 Value 12

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules.

These limits are designed to provide reasonable protection against harmful interference in a residential installation or when the equipment is operated in a commercial environment.

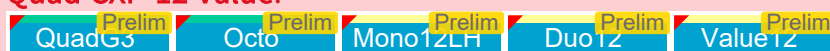
This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

EMC Compliance statement for Korea

Applies to PC1633-T Coaxlink Quad G3, PC3602-T Coaxlink Octo, PC3621-LH-T Coaxlink Mono CXP-12-LH, PC3622-T Coaxlink Duo CXP-12 and PC3623-T Coaxlink Quad CXP-12 Value.



Notice for Korea

Registered products under the Clause 3, Article 58-2 of Radio Waves Act:

Product	Registration Number
PC1633-T Coaxlink Quad G3	MSIP-REM-EUr-PC1633
PC3602-T Coaxlink Octo	R-R-EUr-PC3602
PC3621-LH-T Coaxlink Mono CXP-12-LH	R-R-EUr-PC3622
PC3622-T Coaxlink Duo CXP-12	R-R-EUr-PC3622
PC3623-T Coaxlink Quad CXP-12 Value	R-R-EUr-PC3623

RoHS compliance statement



This product is in conformity with the European Union 2015/863 (ROHS3) directive, that stands for "the restriction of the use of certain hazardous substances in electrical and electronic equipment".

REACH statement



This product is in conformity with the European Union 1907/2006 (REACH) regulation.

WEEE statement



According the European Union 2012/19/EU directive, the product must be disposed of separately from normal household waste. It must be recycled according to the local regulations.

PART IV

GENAPI FEATURES

To be added later