
DALSA Coreco • 7075 Place Robert-Joncas, Suite 142 • St-Laurent, Quebec, H4M 2Z2 • Canada
http://www.imaging.com

# X64-CL Express™

## User's Manual

**Edition 1.00**

**Part number OC-X1CM-USER0**

**NOTICE**



Edition 1.00 printed on: April 10, 2006

Document Number: OC-X1CM-USER0

Printed in Canada

# Contents

# Introduction

## Overview of the Manual

The X64-CL Express User's Manual covers the following topics:

- **Overview of the X64-CL Express**
  Description of the X64-CL Express product and a brief summary of its capabilities.
- **Installing the X64-CL Express**
  Installation procedures for the X64-CL Express board and driver under Windows 2000 or Windows XP, as well as information on camera connectivity.
- **CamExpert Quick Start for the X64-CL Express**
  User's guide to interfacing cameras with CamExpert and using the CamExpert demo tools.
- **The Sapera Demo Applications**
  Overview of the Sapera demo programs which can test the X64-CL Express installation.
- **X64-CL Express Reference**
  Descriptions of X64-CL Express hardware, block diagram, capabilities, and acquisition modes supported.
- **X64-CL Express Sapera Servers & Resources**
  Specifications specific to the Sapera Imaging Library.
- **Technical Specifications**
  X64-CL Express connector locations and pin-out descriptions.
- **Camera Link Interface**
  Overview of the Camera Link specification.
- **X-I/O Module Option**
  Describes the X-I/O module, its configuration, cabling, and usage.
- **DALSA Coreco Contact Information**
  Phone numbers, web site, and important email addresses.

# About the Manual

This manual exists in printed, compiled HTML help, and Adobe® Acrobat (PDF) formats. The help and PDF formats make full use of hypertext cross-references and include links to the **DALSA Coreco** home page on the Internet, located at http://www.imaging.com/, accessed using any web browser.

For information specific to the X64-CL Express, visit the **DALSA Coreco** web site at **www.imaging.com** or http://www.x64.info/.

# Using the Manual

File names, directories, and Internet sites will be in bold text
(e.g., **image2.bmp**, **c:\sapera**, **http://www.imaging.com**).

Text that must be entered using the keyboard will be in typewriter-style text
(e.g., `c:\temp`).

Menu and dialog actions will be indicated in bold text in the order of the instructions to be executed, with each instruction separated by bullets. For example, going to the **File** menu and choosing **Save** would be written as **File•Save**.

# Overview of the X64-CL Express

## Product Part Numbers

### X64-CL Express Board

| Item | Product Number |
|---|---|
| *All models have 85MHz Pixel Clock*<br><br>X64-CL Express with 32 MB of memory | P0-X1C0-XPD00 |
| X-I/O Module (optional): provides 8 input & 8 output general I/Os<br>(see "Appendix: X-I/O Module Option" on page 95) | OC-IO01-STD00 |
| For OEM clients, this manual in printed form, is available on request | OC-X1CM-USER0 |

### X64-CL Express Software

| Item | Product Number |
|---|---|
| Sapera LT version 5.30 or later (required but sold separately)<br>1. Sapera LT: Provides everything you will need to build your imaging application<br>2. Current Sapera compliant board hardware drivers<br>3. Board and Sapera documentation (compiled HTML help, and Adobe Acrobat® (PDF) formats) | OC-SL00-0000000 |
| *(optional)*<br>Sapera Processing Imaging Development Library includes over 600 optimized image processing routines. | Contact Sales at<br>**DALSA Coreco** |

## X64-CL Express Cables & Accessories

| Item | Product Number |
|---|---|
| External Signals Connector Bracket Assembly supplied with each X64-CL Express (connects to J4 – see "X64-CL Express: External Signals Connector Bracket Assembly" on page 84)<br>Note: not used when the X-I/O Module is installed. | OC-64CC-0TIO1 |
| *(optional)* Power interface cable required when supplying power to cameras | OC-COMC-PCPWR |
| *(optional)* Camera Link Video Input Cable:<br>1 meter<br>2 meter | <br>OC-COMC-CLNK0<br>OC-COMC-CLNK6 |
| *(optional)* DB25 male to color coded blunt end cable – 6 foot (1.82 meter) length | OC-COMC-XEND1 |

# About the X64-CL Express Frame Grabber

## X64-CL Express Series Key Features

- Supports Medium Monochrome Camera Link with Flat Field Correction
- Supports Medium RGB Camera Link
- Supports Medium Camera Link with Bayer Filter Decoding
- Supports 2 Base Monochrome Camera Link with Flat Field Correction
- Supports 2 Base RGB Camera Link
- Input lookup tables available for each mode

See “Technical Specifications” on page **71** for detailed information.

## X64-CL Express User Programmable Configurations

Using the X64-CL Express firmware loader utility, firmware for one of the supported modes is easily selected, either during driver installation or manually later on (see "Firmware Update: Manual Mode" on page 12).

For the X64-CL Express board the firmware choices are:

- **One Medium Camera Link Input with Flat Field Correction** (*installation default selection*)
  Support for one Base or one Medium Camera Link port. Flat Field Correction (FFC) includes Fixed Pattern Noise (FPN), Pixel Replacement, Photo Response Non Uniformity (PRNU), and Shading Correction.
- **Two Base Camera Link Inputs with Flat Field Correction**
  Support for two independent Base Camera Link ports. Flat Field Correction (FFC) includes Fixed Pattern Noise (FPN), Pixel Replacement, Photo Response Non Uniformity (PRNU), and Shading Correction.
- **One Medium Camera Link Input with Bayer Decoder**
  Support for one Base or Medium Camera Link port with Hardware Bayer CFA (Color Filter Array) Decoder. No Flat Field Correction is available.

## ACUPlus: Acquisition Control Unit

ACUPlus consists of two sets of independent grab controllers, one pixel packer, and one time base generator. ACUPlus delivers a flexible acquisition front end plus it supports pixel clock rates of up to 85MHz.

ACUPlus acquires variable frame sizes up to 256KB per horizontal line and up to 16 million lines per frame. ACUPlus can also capture an infinite number of lines from a linescan camera without losing a single line of data.

ACUPlus supports standard Camera Link multi-tap configurations from 8 to 24-bit/pixels. Additionally, alternate tap configurations can support up to 8 taps of 8-bits each.

The X64-CL Express model can support two cameras with different tap configurations simultaneously with the Dual Base firmware loaded.

### Camera Link Maximum Acquisition Rates:

This table specifies the X64-CL Express acquisition hardware maximums, not the maximum data transfer rate through the PCI Express 1x bus to system memory.

| Cameras connected | Camera Link standard | Maximum Acquisition rate 85 MHz components | Maximum PCIe 1x transfer rate (including overhead) |
|---|---|---|---|
| 2 | Base | 300 Mbytes/sec | 160 Mbytes/sec |
| 1 | Medium | 300 Mbytes/sec | 160 Mbytes/sec |

## DTE: Intelligent Data Transfer Engine

The X64-CL Express intelligent Data Transfer Engine ensures fast image data transfers between the board and the host computer with zero CPU usage. The DTE provides a high degree of data integrity during continuous image acquisition in a non-real time operating system like Windows. DTE consists of multiple independent DMA units, Tap Descriptor Tables, and Auto-loading Scatter-Gather tables.

### PCI Express 1x Interface

The X64-CL Express is a universal PCI Express1x board, compliant with the PCI Express 1.0 specification. The PCI Express 1x interface supports maximum transfer rates up to 250 Mbytes/sec, while the X64-CL Express board can achieve transfer rates around 160 Mbytes/sec due to memory management overhead. The X64-CL Express board occupies one computer expansion slot and one chassis opening (two slots with the optional X-I/O Module Option).

## Advanced Controls Overview

### Visual Indicators

X64-CL Express features a LED indicator to facilitate system installation and setup. This provides visual feedback indicating when the camera is connected properly and sending data.

### External Event Synchronization

Two sets of dedicated trigger inputs and strobe signals are provided to synchronize precisely image captures with external events.

### Camera Link Communications ports

Two PC independent communication ports provide Camera Link controls for camera configurations. These ports do not require addition PC resources like free interrupts or I/O address space. Accessible via the board device driver, the communication ports preset a seamless interface to Windows-based standard communication applications like HyperTerminal, etc. The communication ports are accessible directly from the Camera Link connectors.

### Quadrature Shaft Encoder

Important feature for web scanning applications, the Quadrature-Shaft-Encoder inputs allow synchronized line captures from external web encoders.

# About the Optional X-I/O Module

The optional X-I/O module adds general purpose software controllable I/O signals to the X64-CL Express. The X-I/O module provides 2 opto-coupled inputs, 6 logic signal inputs (5V or 24V), and 8 TTL outputs (NPN or PNP type selectable). The module also makes available 5V or 12V dc power from the host system.

The X-I/O module can be either purchased with the X64-CL Express board or installed into the computer system at a later time. The module occupies one adjacent PCI slot and connects to the X64-CL Express via a ribbon cable. X-I/O Module external connections are made via the DB37 connector on the module bracket.

X-I/O requires X64-CL Express board driver version 1.00 or later and Sapera LT version 5.30 or later.

See "Appendix: X-I/O Module Option" on page 95 for details and specifications.

# Development Software Overview

## Sapera LT Library

Sapera LT is a powerful development library for image acquisition and control. Sapera LT provides a single API across all current and future DALSA Coreco hardware. Sapera LT delivers a comprehensive feature set including program portability, versatile camera controls, flexible display functionality and management, plus easy to use application development wizards.

Sapera LT comes bundled with CamExpert, an easy to use camera configuration utility to create new, or modify existing camera configuration files.

## Sapera Processing Library

Sapera Processing is a comprehensive set of C++ classes for image processing and analysis. Sapera Processing offers highly optimized tools for image processing, blob analysis, search (pattern recognition), OCR and barcode decoding.

# Installing the X64-CL Express

## Warning! (Grounding Instructions)

Static electricity can damage electronic components. Please discharge any static electrical charge by touching a grounded surface, such as the metal computer chassis, before performing any hardware installation.

If you do not feel comfortable performing the installation, please consult a qualified computer technician.

**Never** remove or install any hardware component with the computer power on.

## Upgrading Sapera or any DALSA Coreco Board Driver

When installing a new version of Sapera or a DALSA Coreco acquisition board driver in a computer with a previous installation, the current version **must** be un-installed first. Upgrade scenarios are described below.

### Board Driver Upgrade Only

Minor upgrades to acquisition board drivers are typically distributed as ZIP files available in the DALSA Coreco web site http://www.imaging.com/downloads. Board driver revisions are also available on the next release of the Sapera CD-ROM.

Often minor board driver upgrades do not require a new revision of Sapera. To confirm that the current Sapera version will work with the new board driver:

- Check the new board driver ReadMe.txt file before installing, for information on the minimum Sapera version required.
- If the ReadMe.txt file does not specify the Sapera version, contact DALSA Coreco Technical Support (see "Technical Support" on page 110 ).

To upgrade the board driver only:

- Logon the computer as an administrator or with an account that has administrator privileges.
- From the Windows start menu select **Start • Programs • DALSA Coreco • X64-CL Express Device Driver • Modify-Repair-Remove**.
- Click on **Remove**.
- When the driver un-install is complete, reboot the computer.

- Logon the computer as an administrator again.
- Install the new board driver. Run **Setup.exe** if installing manually from a downloaded driver file.
- If the new driver is on a Sapera CD-ROM follow the installation procedure described in "Installing X64-CL Express Hardware and Driver" on page 11.
- Note that you can not install a DALSA Coreco board driver without Sapera LT installed on the computer.

## Sapera and Board Driver Upgrades

When both Sapera and the DALSA Coreco acquisition board driver are upgraded, follow the procedure described below.

- Logon the computer as an administrator or with an account that has administrator privileges.
- From the Windows start menu select **Start • Programs • DALSA Coreco • X64-CL Express Device Driver • Modify-Repair-Remove**.
- Click on **Remove** to uninstall the board driver.
- From the Windows start menu select **Start • Programs • DALSA Coreco • Sapera LT • Modify-Repair-Remove**.
- Click on **Remove** to uninstall Sapera.
- Reboot the computer and logon the computer as an administrator again.
- Install the new versions of Sapera and the board driver as if this was a first time installation. See "Sapera LT Library Installation" on page 10 and "Installing X64-CL Express Hardware and Driver" on page 11 for installation procedures.

# Sapera LT Library Installation

Note: to install Sapera LT and the X64-CL Express device driver, logon to the workstation as an administrator or with an account that has administrator privileges.

The Sapera LT Development Library (or 'runtime library' if application execution without development is preferred) must be installed before the X64-CL device driver.

- Insert the DALSA Coreco Sapera CD-ROM. If **AUTORUN** is enabled on your computer, the DALSA Coreco installation menu is presented.
- If **AUTORUN** is not enabled, use Windows Explorer and browse to the root directory of the CD-ROM. Execute **launch.exe** to start the DALSA Coreco installation menu and install the required Sapera components.
- The installation program will prompt you to reboot the computer.

Refer to *Sapera LT User's Manual* for additional details about Sapera LT.

# Installing X64-CL Express Hardware and Driver

## In a Windows 2000 or Windows XP System

- Turn the computer off and open the computer chassis to allow access to the expansion slot area.
- Install the X64-CL Express into a free PCI Express 1x expansion slot. The X64-CL Express could also be installed in a PCI Express 4x or 16x slot if that is the free choice.
- Close the computer chassis and turn the computer on. Driver installation requires administrator rights for the current user of the computer.
- Windows will find the X64-CL Express and start its **Found New Hardware Wizard**. Click on the **Cancel** button to close the Wizard Application.
- Insert the DALSA Coreco Sapera CD-ROM. If **AUTORUN** is enabled on your computer, the DALSA Coreco installation menu is presented. Install the X64-CL Express driver.
- If **AUTORUN** is not enabled, use Windows Explorer and browse to the root directory of the CD-ROM. Execute **launch.exe** to start the DALSA Coreco installation menu and install the X64-CL Express driver.
- Reboot the computer when prompted. During the early stages of the Windows reboot, the X64-CL Express firmware loader application starts. This is described in detail in the following section. Allow Windows to complete its reboot before proceeding.
- When using **Windows 2000**, the **Digital Signature Not Found** message is displayed. Click on Yes to continue the X64-CL Express driver installation. Reboot the computer when prompted.
- When using **Windows XP**, a message stating that the X64-CL Express software has not passed **Windows Logo testing** is displayed. Click on **Continue Anyway** to finish the X64-CL Express driver installation. Reboot the computer when prompted.

# X64-CL Express Firmware Loader

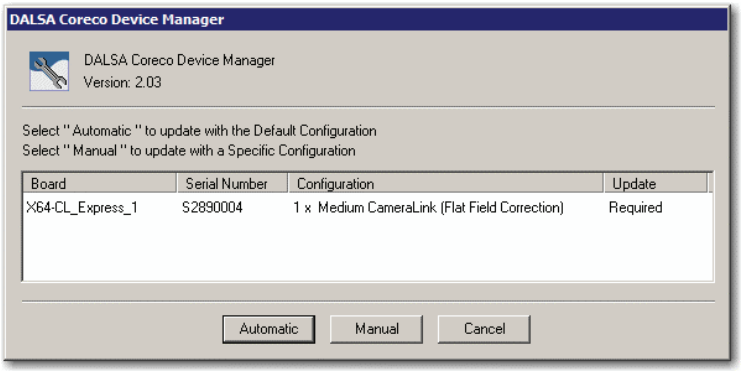
After Windows boots, the Device Manager-Firmware Loader program automatically executes. If it determines that the X64-CL Express board requires a firmware update, a message is displayed asking to accept an update. Click YES to start the Coreco Device Manager and update the board firmware. As described below, choose automatic mode for the default board configuration or manual mode to select an alternative.

***Important*:** In the vary rare case of firmware loader errors please see "Recovering from a Firmware Update Error" on page 20.

## Firmware Update: Automatic Mode

Click **Automatic** to update the X64-CL Express firmware. The X64-CL Express board supports three firmware configurations with the default being one Medium camera with Flat Field correction mode. See "X64-CL Express Series Key Features" on page 5 and "X64-CL Express User Programmable Configurations" on page 5 for details on all supported modes, which can be selected via a manual firmware update.

If there are multiple X64-CL Express boards in the system, all boards will be updated with new firmware. If any installed X64-CL Express boards installed in a system already have the correct firmware version, an update is not required. In the following screen shot, a single X64-CL Express board is installed in the system and the default Medium configuration is ready to be programmed.




## Firmware Update: Manual Mode

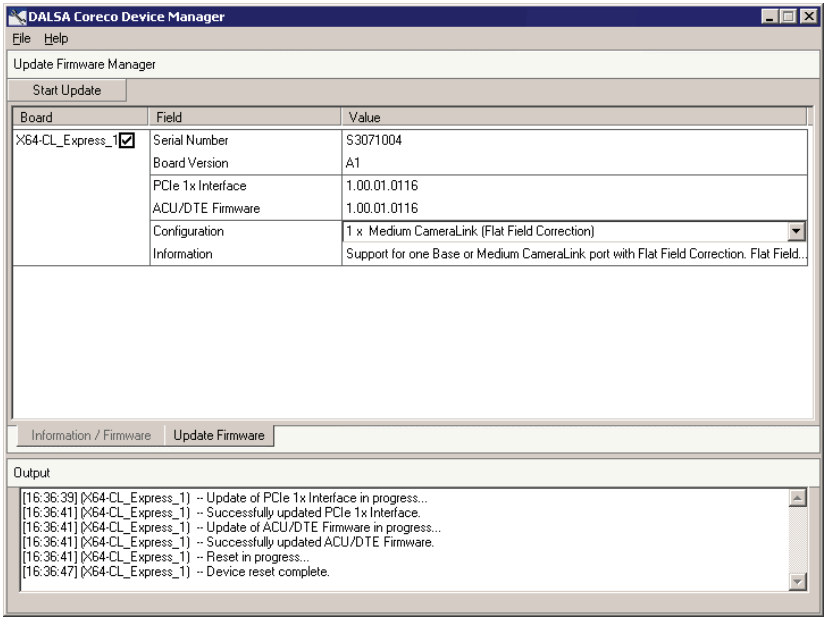
Select **Manual** mode to load firmware other then the default version or when, in the case of multiple X64-CL Express boards in the same system, each requires different firmware.

The figure below shows the Device Manager manual firmware screen. Information on all installed X64-CL Express boards, their serial numbers, and their firmware components are shown.

A manual firmware update is as follows:

- Select the X64-CL Express via the board selection box (if there are multiple boards in the system)

- From the Configuration field drop menu select the firmware version required
- Click on the Start Update button
- Observe the firmware update progress in the message output window
- Close the Device manager program when the X64-CL Express board reset complete message is shown.




## Executing the Firmware Loader from the Start Menu

If required, the X64-CL Firmware Loader program is executed via the Windows Start Menu shortcut **Start • Programs • DALSA Coreco • X64-CL Express Device Driver •Firmware Update**. A firmware change after installation is required to select a different Camera Link mode. See "X64-CL Express User Programmable Configurations" on page 5.

# Enabling the Camera Link Serial Control Port

The Camera Link cabling specification includes a serial communication port for direct camera control by the frame grabber (see "J1: Camera Link Connector 1 " on page 78). The X64-CL Express driver supports this serial communication port either directly or by mapping it to a host computer COM port. Any serial port communication program, such as Windows HyperTerminal, can connect to the camera in use and modify its function modes via its serial port controls. The X64-CL Express serial port supports communication speeds from 9600 to 115 kbps.

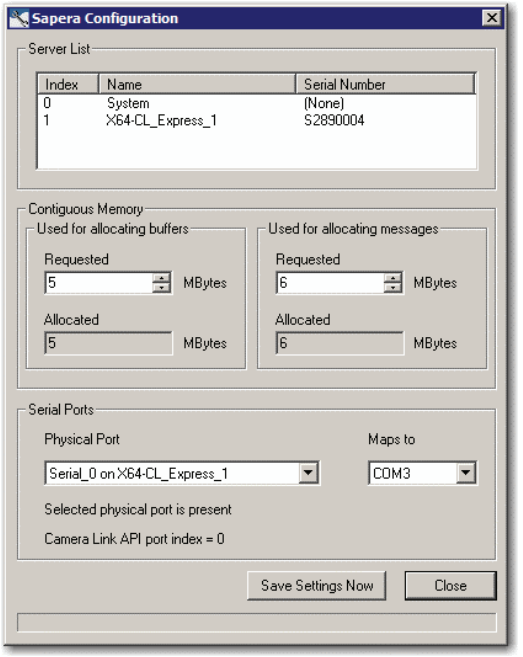
Note: if your serial communication program can directly select the X64-CL Express serial port then mapping to a system COM port is not necessary.

The X64-CL Express serial port is mapped to an available COM port by using the Sapera Configuration tool. Run the program from the Windows start menu: **Start•Programs•DALSA Coreco•Sapera LT•Sapera Configuration**.

## COM Port Assignment

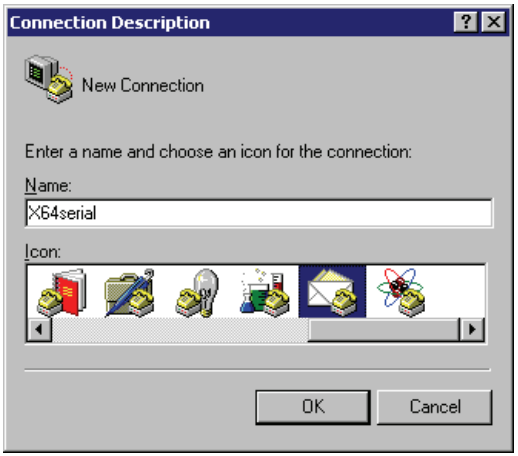
The lower section of the Sapera Configuration program screen contains the serial port configuration menu. Configure as follows:

- Use the **Physical Port** drop menu to select the Sapera board device from all available Sapera boards with serial ports (when more then one board is in the system).
- Use the **Maps to** drop menu to assign an available COM number to that Sapera board serial port.
- Click on the **Save Settings Now** button then the **Close** button. You are prompted to reboot your computer to enable the serial port mapping.
- The X64-CL Express serial port, now mapped to COM3 in this example, is available as a serial port to any serial port application for camera control. Note that this serial port is not listed in the **Windows Control Panel•System Properties•Device Manager** because it is a logical serial port mapping.
- An example setup using Windows HyperTerminal follows (see "Setup Example with Windows HyperTerminal" on page 15).

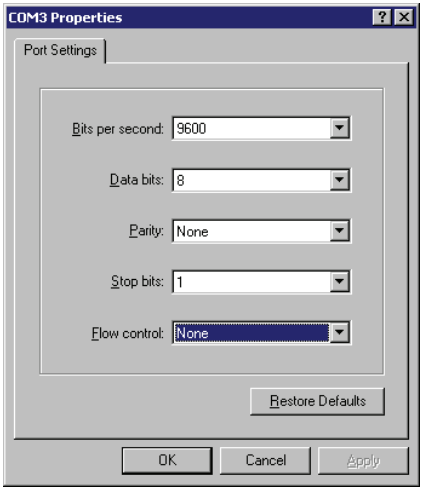



## Setup Example with Windows HyperTerminal

- Run HyperTerminal and type a name for the new connection when prompted. Then click OK.
- On the following dialog screen select the COM port to connect with. In this example the X64-CL Express serial port was previously mapped to COM3 by the Sapera Configuration program.




- HyperTerminal now presents a dialog to configure the COM port properties. Change settings as required by the camera you are connecting to. Note that the X64-CL Express serial port does not support hardware flow control.



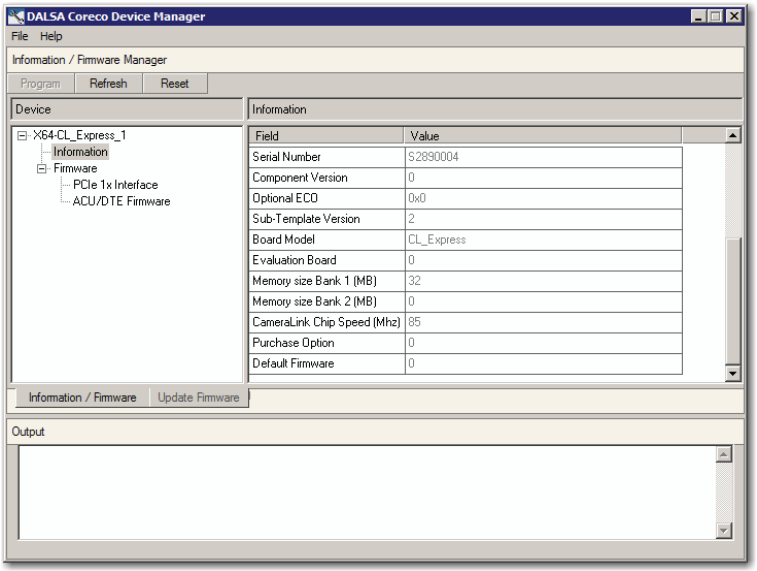
# Displaying X64-CL Express Board Information

The Coreco Device Manager program also displays information about the X64-CL Express boards installed in the system. To view board information run the program via the Windows Start Menu shortcut **Start • Programs • DALSA Coreco • X64-CL Express Device Driver • CorDeviceManager**.

## Coreco Device Manager – Board Viewer

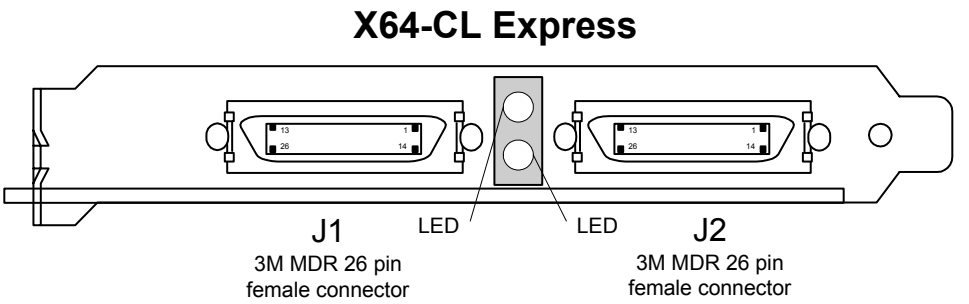
The following screen image shows the Coreco Device Manager program with the Information/Firmware tab active. The left window displays all X64-CL Express boards in the system and their individual device components. The right window displays the information stored in the selected board device. This example screen shows the X64-CL Express information contained in the EEProm component.

The X64-CL Express device manager report file (BoardInfo.txt) is generated by clicking **File • Save Device Info**. This report file may be requested by DALSA Coreco Technical Support to aid in troubleshooting installation or operational problems.



# Camera to Camera Link Connections

## X64-CL Express End Bracket




The hardware installation process is completed with the connection of a supported camera to the X64-CL Express board using Camera Link cables (see "Camera Link Cables" on page **93**).

- The X64-CL Express board supports a camera with one or two Camera Link MDR-26 connectors (two Base or one Medium – see "Data Port Summary" on page 92 for information on Camera Link configurations).
- Connect the camera to the J1 connector with a Camera Link cable. When using a Medium camera, connect the second camera connector to J2.

Refer to section "Connector and Switch Specifications" on page **76** for details on the Camera Link connectors.

Caution: If the camera is powered by the X64-CL Express, it is very important that the correct power supply voltage is selected correctly. Refer to "J8: Power to Camera Voltage Selector" on page **87** for information on the selection jumper.

Contact DALSA Coreco or browse our web site http://www.imaging.com/camsearch for the latest information on X64-CL Express supported cameras.

# Configuring Sapera

## Viewing Installed Sapera Servers

The Sapera configuration program (**Start • Programs • DALSA Coreco • Sapera LT • Sapera Configuration**) allows the user to see all available Sapera servers for the installed Sapera-compatible boards. The **System** entry represents the system server. It corresponds to the host machine (your computer) and is the only server that should always be present.

## Increasing Contiguous Memory for Sapera Resources

The **Contiguous Memory** section lets the user specify the total amount of contiguous memory (a block of physical memory, occupying consecutive addresses) reserved for the resources needed for **Sapera buffers** allocation and **Sapera messaging**. For both items, the **Requested** value dialog box shows the driver default memory setting while the **Allocated** value displays the amount of contiguous memory that has been allocated successfully. The default values will generally satisfy the needs of most applications.

The **Sapera buffers** value determines the total amount of contiguous memory reserved at boot time for the allocation of dynamic resources used for host frame buffer management such as DMA descriptor tables plus other kernel needs. Adjust this value higher if your application generates any out-of-memory error while allocating host frame buffers. You can approximate the amount of contiguous memory required as follows:

- Calculate the total amount of host memory used for frame buffers
  [ number of frame buffers • number of pixels per line • number of lines • (2 - if buffer is 10 or 12 bits) ].
- Provide 1MB for every 256 MB of host frame buffer memory required.
- Add an additional 1 MB if the frame buffers have a short line length, say 1k or less
  ( the increased number of individual frame buffers requires more resources ).
- Add an additional 2 MB for various static and dynamic Sapera resources.
- Test for any memory error when allocating host buffers. Simply use the Buffer menu of the Sapera Grab demo program (see "Using the Grab Demo" on page 39) to allocate the number of host buffers required for your acquisition source. Feel free to test the maximum limit of host buffers possible on your host system – the Sapera Grab demo will not crash when the requested number of host frame buffers cannot be allocated.

### Host Computer Frame Buffer Memory Limitations

When planning a Sapera application and its host frame buffers used, plus other Sapera memory resources, do not forget the Windows operating system memory needs. Window XP as an example, should always have a minimum of 128 MB for itself.

A Sapera application using *scatter gather buffers* could consume most of the remaining system memory. When using frame buffers allocated as a *single contiguous memory block*, typical limitations are one third

of the total system memory with a maximum limit of approximately 100 MB. See the Buffer menu of the Sapera Grab demo program for information on selecting the type of host buffer memory allocation.

## Contiguous Memory for Sapera Messaging

The current value for **Sapera messaging** determines the total amount of contiguous memory reserved at boot time for messages allocation. This memory space is used to store arguments when a Sapera function is called. Increase this value if you are using functions with large arguments, such as arrays and experience any memory errors.

# Troubleshooting Installation Problems

The X64-CL Express (and the X64 family of products) has been tested by DALSA Coreco in a variety of computers. Although unlikely, installation problems may occur due to the constant changing nature of computer equipment and operating systems. This section describes what the user can verify to determine the problem or the checks to make before contacting DALSA Coreco Technical Support. Note that information provided within this section will be updated with the latest information DALSA Coreco can provide for each manual version released.

If you require help and need to contact DALSA Coreco Technical Support, make detailed notes on your installation and/or test results for our technical support to review. See "Technical Support" on page 110 for contact information.

## Recovering from a Firmware Update Error

This procedure is required if any failure occurred while updating the X64-CL Express firmware on installation or during a manual firmware upgrade. On the rare occasion the board has corrupted firmware, any Sapera application such as CamExpert or the grab demo program will not find an installed board to control.

Possible reasons for firmware loading errors or corruption are:

- Computer system mains power failure or deep brown-out.
- PCI bus or checksum errors.
- PCI bus timeout conditions due to other devices.
- User forcing a partial firmware upload using an invalid firmware source file.

When the X64-CL Express firmware is corrupted, executing a manual firmware upload will not work because the firmware loader can not communicate with the board. In an extreme case, corrupted firmware may even prevent Windows from booting.

***Solution*:** The user manually forces the board to initialize from write protected firmware designed only to allow driver firmware uploads. When the firmware upload is complete, the board is then rebooted to initialize in its normal operational mode.

- This procedure requires removing the X64-CL Express board several times from the computer.

- *Important*: Referring to the board's user manual (in the connectors and jumpers reference section), identify the configuration jumper location. The Boot Recovery Mode jumper for the X64-CL Express is J11 (see "J11: Start Mode" on page **88**).
- Shut down Windows and power OFF the computer.
- Remove the shorting jumper installed on J11. The default position is J11 installed for normal operation, while the jumper removed is for the boot recovery mode position.
- Power on the computer. Windows will boot normally.
- When Windows has started, do a manual firmware update procedure to update the firmware again (see "Executing the Firmware Loader from the Start Menu" on page 13).
- When the update is complete, shut down Windows and power off the computer.
- Set the shorting jumper back onto J11 (i.e. default position) and power on the computer once again.
- Verify that the Coreco frame grabber is functioning by running a Sapera application such as CamExpert. The Sapera application will now be able to communicate with the X64-CL Express board.

## Windows Event Viewer

Windows Event Viewer (**Computer Management • System Tools • Event Viewer**), lists various events that have taken place during the Operating System boot sequence. If a driver generates an error, it will normally log an entry in the event list.

## Coreco Device Manager Program

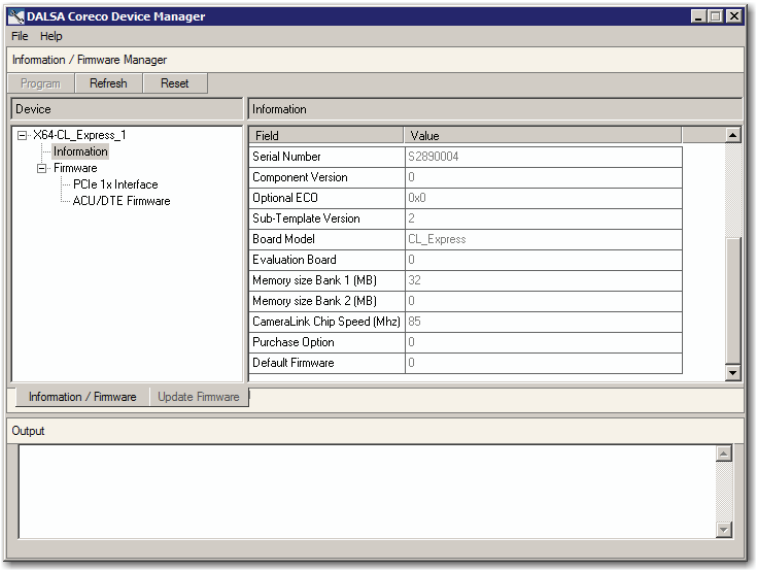
The Coreco Device Manager program provides a convenient method of collecting information about the installed X64-CL Express. System information such as operating system, computer CPU, system memory, PCI configuration space, plus X64-CL Express firmware information can be displayed or written to a text file (default file name – BoardInfo.txt). Note that this is a second function mode of the same program used to manually upload firmware to the X64-CL Express.

Execute the program via the Windows Start Menu shortcut **Start • Programs • DALSA Coreco • X64-CL Express Device Driver • CorDeviceManager**. If the Coreco Device Manager program does not run, it will exit with a message that the board was not found. Since the X64-CL Express board must have been in the system to install the board driver, possible reasons for an error are:

- Board was removed
- Board driver did not start or was terminated
- PCI conflict after some other device was installed

## Information Window

The following figure shows the Coreco Device Manager information screen. Click to highlight one of the board components and the information for that item is shown on the right hand window, as described below.

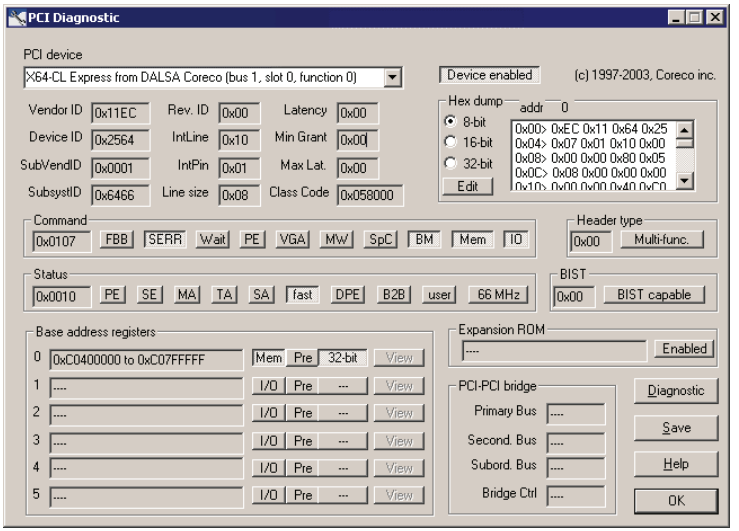



- Select **Information** to display identification and information stored in the X64-CL Express firmware.
- Select **Firmware** to display version information for the firmware components.
- Select one of the firmware components to load *custom* firmware when supplied by DALSA Coreco engineering for a future feature.
- Click on **File • Save Device Info** to save all information to a text file. Email this file when requested by Coreco Technical Support.

# PCI Configuration

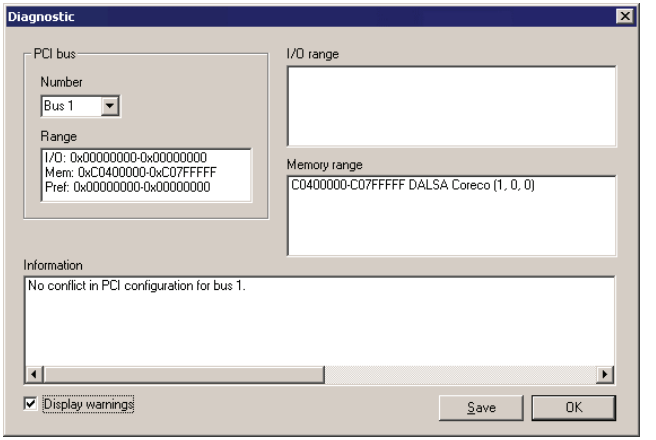
One of the first items to check when there is a problem with any PCI board is to examine the system PCI configuration and ensure that there are no conflicts with other PCI or system devices. The *DALSA Coreco PCI Diagnostic* program (**cpcidiag.exe**) allows examination of the PCI configuration registers and can save this information to a text file. Run the program via the Windows Start Menu shortcut **Start • Programs • DALSA Coreco • Sapera LT • Tools • PCI Diagnostics**.

As shown in the following screen image, use the first drop menu to select the PCI device to examine. Select the device from DALSA Coreco. Note the bus and slot number of the installed board (this will be unique for each system unless systems are setup identically). Click on the **Diagnostic** button to view an analysis of the system PCI configuration space.




Clicking on the **Diagnostic** button opens a new window with the diagnostic report. From the PCI Bus Number drop menu select the bus number that the X64-CL Express is installed in—in this example the slot is bus 1.

The window now shows the I/O and memory ranges used by each device on the selected PCI bus. The information display box will detail any PCI conflicts. If there is a problem, click on the **Save** button. A file named '**pcidiag.txt'** is created (in the Sapera\bin directory) with a full dump of the PCI configuration registers. Email this file when requested by the DALSA Coreco Technical Support group along with a full description of your computer.




## Sapera and Hardware Windows Drivers

The next step is to make certain the appropriate DALSA Coreco drivers have started successfully during the boot sequence. Example, click on the **Start • Programs • Accessories • System Tools • System Information • Software Environment**. Click on **Drivers** (Windows 2000) or **System Drivers** (Windows XP). Make certain the following drivers have started for the **X64-CL Express** .

| Device | Description |
|---|---|
| Corx64Expr | *X64-CL Express messaging* |
| CorLog | *Sapera Log viewer* |
| CorMem | *Sapera Memory manager* |
| CorPci | *Sapera PCI configuration* |
| CorSerial | *Sapera Serial Port manager* |

DALSA Coreco Technical Support may request that you check the status of these drivers as part of the troubleshooting process.

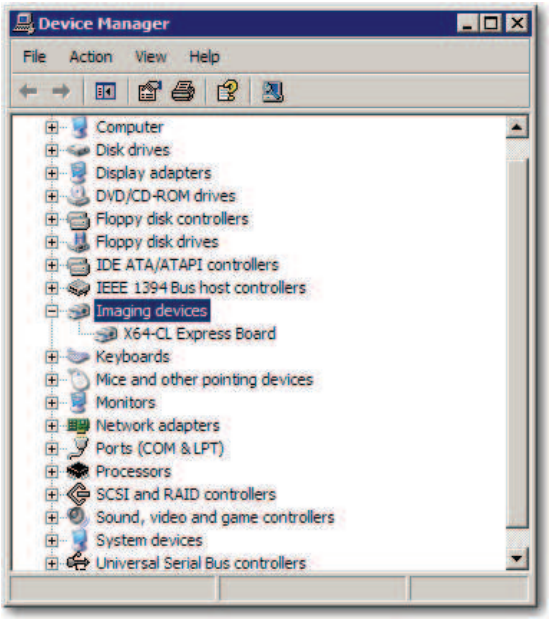
## Log Viewer

The third step in the verification process is to save in a text file the information collected by the Coreco Log Viewer program. Run the program via the Windows Start Menu shortcut **Start • Programs • DALSA Coreco • Sapera LT • Tools • Coreco Log Viewer**.

The Coreco Log Viewer lists information about the installed DALSA Coreco drivers. Click on **File • Save** and you will be prompted for a text file name to save the Log Viewer contents. Email this text file to DALSA Coreco Technical Support when requested or as part of your initial contact email.

## Windows Device Manager

In Windows 2000 or XP, use the Start Menu shortcut **Start • Settings • Control Panel • System • Hardware • Device Manager**. As shown in the following screen images, look for *X64-CL Express* board under “Imaging Devices”. Double-click and look at the device status. You should see “This device is working properly.” Go to “Resources” tab and make certain that the device is mapped and has an interrupt assigned to it, without any conflicts.




## Memory Requirements with Area Scan Acquisitions

The X64-CL Express allocates two frame buffers in onboard memory, each equal in size to the acquisition frame buffer. This double buffering memory allocation is automatic at the driver level. The X64-CL Express driver uses two buffers to ensure that the acquired video frame is complete and not corrupted in cases where the image transfer to host system memory may be interrupted and delayed by other host system processes. That is, the image acquisition to one frame buffer is not interrupted by any delays in transfer of the other frame buffer (which contains the previously acquired video frame) to system memory.

The total size of the two internal frame buffers must be somewhat smaller than the total onboard memory due to memory overhead required for image transfer management. When the X64-CL Express does not have enough onboard memory for two frame buffers, the memory error message
[ **Error: "CorXferConnect" <Xfer module> - No memory ()** ] occurs when loading a Sapera camera file, or when the application configures a frame buffer.

## Symptoms: CamExpert Detects no Boards

- **If using Sapera version 5.30 or later:**
  When starting CamExpert, if no DALSA Coreco board is detected, CamExpert will start in offline mode. There is no error message and CamExpert is functional for creating or modifying a camera configuration file. If CamExpert should have detected the installed board, troubleshoot the installation problem as described below.

### Troubleshooting Procedure

When CamExpert detects no installed DALSA Coreco board, there could be a hardware problem, a PnP problem, a PCI problem, a kernel driver problem, or a software installation problem.

- Make certain that the card is properly seated in PCIe slot.
- Perform all installation checks described in this section ("Troubleshooting Installation Problems" on page 20) before contacting Technical Support.
- Try the board in a different PCIe slot if available.

## Symptoms: X64-CL Express Does Not Grab

You are able to start Sapera CamExpert but you do not see an image and the frame rate displayed is 0.

- Verify power is connected to the camera.
- Verify the camera and timing parameters with the camera in free run mode.
- Verify you can grab with the camera in free run mode.
- Make certain that you provide an external trigger if the camera configuration file requires one. Use the software trigger feature of CamExpert if you do not have a trigger source.
- Make certain that the camera is properly connected to the cable.
- Make certain that the camera is configured for the proper mode of operation. This must match the camera configuration file. Refer to your camera datasheet.
- Try to snap one frame instead of continuous grab.
- Perform all installation checks described in this section ("Troubleshooting Installation Problems" on page 20) before contacting Technical Support.

## Symptoms: Card grabs black

You are able to use Sapera CamExpert, the displayed frame rate is as expected, but the display is always black.

- Set your camera to manual exposure mode and set the exposure to a longer period, plus open the lens iris.
- Try to snap one frame instead of continuous grab.
- Make certain that the input LUT is not programmed to output all '0's.
- This problem is sometimes caused by a PCIe transfer issue. No PCIe transfer takes place, so the frame rate is above 0 but nevertheless no image is displayed in CamExpert.
- Make certain that BUS MASTER bit in the PCIe configuration space is activated. Look in PCI Diagnostics for **BM** button under "Command" group. Make certain that the **BM** button is activated.
- Perform all installation checks described in this section ("Troubleshooting Installation Problems" on page 20) before contacting Technical Support.

## Symptoms: Card acquisition bandwidth is less than expected

The X64-CL Express acquisition bandwidth is less than expected.

- Review the system for problems or conflicts with other expansion boards or drivers.
- Remove other PCI Express, PCI-32 or PCI-64 boards and check acquisition bandwidth again. Coreco engineering has seen this case where other PCI boards in some systems cause limitations in transfers. Each system, with its combination of system motherboard and PCI boards, will be unique and will need to be tested for bandwidth limitations affecting the imaging application.

# CamExpert Quick Start for the X64-CL Express

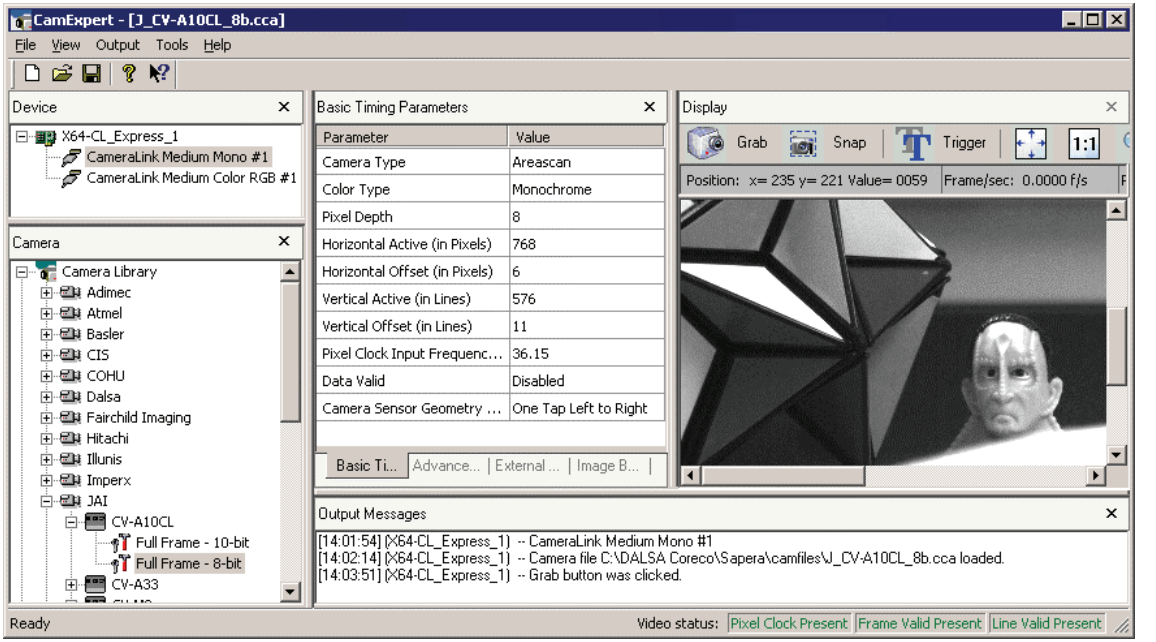
## Interfacing Cameras with CamExpert

CamExpert is the camera interfacing tool for frame grabber boards supported by the Sapera library. CamExpert generates the Sapera camera configuration file (*yourcamera.ccf*) based on timing and control parameters entered. For backward compatibility with previous versions of Sapera, CamExpert also reads and writes the *.cca and *.cvi camera parameter files.

Every Sapera demo program starts by a dialog window to select a camera configuration file. Even when using the X64-CL Express with common video signals, a camera file is required. Therefore CamExpert is typically the first Sapera application run after an installation. Obviously existing .ccf files can be copied to the new installation when similar cameras are used.

### CamExpert Example with a Monochrome Camera

The image below shows CamExpert with the X64-CL Express. The camera outputs monochrome 8-bit video on a Base Camera Link interface. After selecting the camera model, the timing parameters are displayed and the user can test by clicking on *Grab*. Descriptions of the CamExpert windows follows the image.




The CamExpert sections are:

- **Device:** Select which acquisition device to control and configure a camera file for. Required in cases where there are multiple boards in a system and also when one board supports multiple acquisition types. Note in this example, the X64-CL Express was installed with Medium Camera Link support for monochrome or RGB cameras.
- **Camera:** Select the timing for a specific camera model included with the Sapera installation or a standard video standard. The *User's* subsection is where created camera files are stored.
- **Timing & Control Parameters:** The central section of CamExpert provides access to the various Sapera parameters supported by X64-CL Express. There are four or five tabs dependent on the acquisition board, as described below:

| | |
|---|---|
| **Basic Timing Parameters** | Basic parameters used to define the timing of the camera. This includes the vertical, horizontal, and pixel clock frequency. This tab is sufficient to configure a free-running camera. |
| **Advanced Control Parameters** | Advanced parameters used to configure camera control mode and strobe output. Also provides analog signal conditioning (brightness, contrast, DC restoration, etc.) for analog boards. |
| **External Trigger Parameters** | Parameters to configure the external trigger characteristics. |
| **Image Buffer and AOI Parameters** | Control of the host buffer dimension and format. |
| **Multi-Camera Control Parameters** | Dependent on the frame acquisition board, provides camera selection and color planar transfer selection. |

- **Display:** An important component of CamExpert is its live acquisition display window which allows immediate verification of timing or control parameters without the need to run a separate acquisition program. **Grab** starts continuous acquisition (button then toggles to **Freeze** to stop). **Snap** is a single frame grab. **Trigger** is a software trigger to emulate an external source.
- **Output Messages and Bottom Status Bar:** Events and errors are logged for review. Camera connection status is displayed where green indicates signal present.

For context sensitive help click on the  button then click on a camera configuration parameter. A short description of the configuration parameter will be shown in a popup. Click on the  button to open the help file for more descriptive information on CamExpert.

# CamExpert Demonstration and Test Tools

The CamExpert utility also includes a number of demonstration features which make CamExpert the primary tool to configure, test and calibrate your camera and imaging setup. Display tools include, image pixel value readout, image zoom, and line profiler.

Functional tools include hardware Flat Field calibration and operation support (see “Using the Flat Field Correction Tool” on page 34), plus support for either hardware based or software Bayer filter camera decoding with auto white balance calibration (see “Using the Bayer Filter Tool” on page 36).

# Camera Types & Files Applicable to the X64-CL Express

The X64-CL Express supports digital area scan or linescan cameras using the Camera Link interface standard. See "Camera to Camera Link Connections" on page 18 for information on connecting a Camera Link camera.

Contact DALSA Coreco or browse our web site [ http://www.imaging.com/camsearch ] for the latest information and application notes on X64-CL Express supported cameras.

## Camera Files Distributed with Sapera

The Sapera distribution CDROM includes camera files for a selection of X64-CL Express supported cameras. Using the Sapera CamExpert program, you may use the camera files (CCA) provided to generate a camera configuration file (CCF) that describes the desired camera and frame grabber configuration..

DALSA Coreco continually updates a camera application library composed of application information and prepared camera files. Along with the camera search utility on the DALSA Coreco web site, as described above, a number of camera files are ready to download from the DALSA Coreco FTP site [ftp://ftp.coreco.com/public/Sapera/CamFile_Updates]. Camera files are ASCII text and can be read with Windows Notepad on any computer without having Sapera installed.

# CamExpert Memory Errors when Loading Camera Configuration Files

The memory error message [ **Error: "CorXferConnect" <Xfer module> - No memory ()** ] may occur when loading a Sapera camera file, or when the application configures a frame buffer for area scan cameras. The problem is that the X64-CL Express does not have enough onboard memory for two frame buffers.

The X64-CL Express when used with area scan cameras, allocates two internal frame buffers in onboard memory, each equal in size to the acquisition frame buffer. This allocation is automatic at the driver level. The X64-CL Express driver allocates two buffers to ensure that the acquired video frame is complete and not corrupted in cases where the transfer to host system memory may be interrupted by other host system processes.

The total size of the two internal frame buffers must be somewhat smaller than the total onboard memory due to memory overhead required for image transfer management. Also note that the X64-CL Express board when configured for two Base inputs, equally divides the onboard memory between the two acquisition modules, reducing the available memory for the two buffers by half.

# Overview of Sapera Acquisition Parameter Files (*.ccf or *.cca/*.cvi)

## Concepts and Differences between the Parameter Files

There are two components to the legacy Sapera acquisition parameter file set: CCA files (also called cam-files) and CVI files (also called VIC files, i.e. video input conditioning). The files store video-signal parameters (CCA) and video conditioning parameters (CVI), which in turn simplifies programming the frame-grabber acquisition hardware for the camera in use. **Sapera LT 5.0** introduces a new camera configuration file (**CCF**) that combines the CCA and CVI files into one file.

Typically, a camera application will use a CCF file per camera operating mode (or one CCA file in conjunction with several CVI files, where each CVI file defines a specific camera operating mode). An application can also have multiple CCA/CCF files so as to support different image format modes supported by the camera or sensor (such as image binning or variable ROI).

## CCF File Details

Files using the ".CCF" extension, (CORECO Camera Configuration files), are essentially the camera (CCA) and frame grabber (CVI) parameters grouped into one file for easier configuration file management. This is the default Camera Configuration file used with Sapera LT 5.0 and the CamExpert utility.

## CCA File Details

DALSA Coreco distributes camera files using the ".CCA" extension, (CORECO CAMERA files), which contain all parameters describing the camera video signal characteristics and operation modes (what the camera outputs). The Sapera parameter groups within the file are:

- Video format and pixel definition.

- Video resolution (pixel rate, pixels per line, lines per frame).
- Synchronization source and timing.
- Channels/Taps configuration.
- Supported camera modes and related parameters.
- External signal assignment.

### CVI File Details

Legacy files using the ".CVI" extension, (CORECO VIDEO files), contain all operating parameters related to the frame grabber board - what the frame grabber can actually do with camera controls or incoming video. The Sapera parameter groups within the file are:

- Activate and set any supported camera control mode or control variable.
- Define the integration mode and duration.
- Define the strobe output control.
- Allocate the frame grabber transfer ROI, the host video buffer size and buffer type (RGB888, RGB101010, MONO8, MONO16).
- Configuration of line/frame trigger parameters such as source (internal via the frame grabber /external via some outside event), electrical format (TTL, LVDS, OPTO-isolated), and signal active edge or level characterization.

## Camera Interfacing Check List

Before undertaking the task of interfacing a camera from scratch with CamExpert:

- Confirm that DALSA Coreco has not already published an application note with camera files [http://www.imaging.com/camsearch].
- Confirm that the correct version or board revision of X64-CL Express is used. Confirm that the required firmware is loaded into the X64-CL Express .
- Confirm that Sapera does not already have a .cca file for your camera installed on your hard disk. If there is a .cca file supplied with Sapera, then use CamExpert to automatically generate the .ccf file with default parameter values matching the frame grabber capabilities.
- Check if the Sapera installation has a similar type of camera file. A similar .cca file can be loaded into CamExpert where it is modified to match timing and operating parameters for your camera, and lastly save them as Camera Configuration file (.ccf).
- Finally, if your camera type has never been interfaced, run CamExpert after installing Sapera and the acquisition board driver, select the board acquisition server, and manually enter the camera parameters.

# Using the Flat Field Correction Tool

Flat Field Correction is the process of eliminating small gain differences between pixels in a sensor array. That sensor when exposed to a uniformly lit field will have no gray level differences between pixels when calibrated flat field correction is applied to the image. The CamExpert Flat Field tool functions with hardware supporting flat field processing.

## X64-CL Express Flat Field Support

The X64-CL Express supports hardware based real-time Flat Field Correction when used with its dual Base or one Medium configuration.

**Important:** Flat field and flat line correction impose limitations to the maximum acquisition frame rate. Please contact the DALSA Coreco support group for more details on camera specific maximum supported acquisition rates.

### Loading the Required Camera File

Select the required camera configuration file for the connected camera. Verify the acquisition with the live grab function. Make camera adjustments to get good images.

Also at this time make preparations to grab a flat gray level image such as a clean evenly lighted white wall or non-glossy paper. Note the lens iris position for a white but not saturated image. This white image is required for the calibration process.

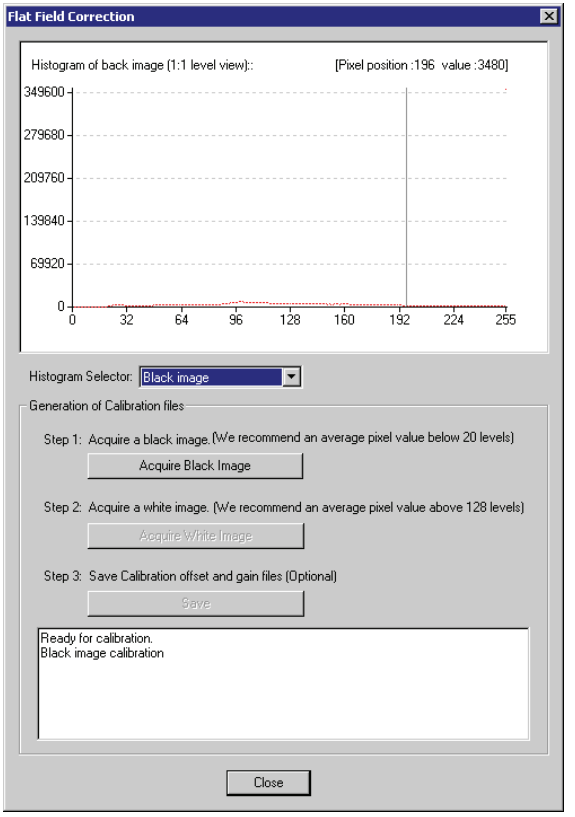
## Flat Field Correction Calibration Procedure

Calibration is the process of taking two reference images, one of a black field – one of a light gray field (not saturated), to generate correction data for images captured by the CCD. Each CCD pixel data is modified by the correction factor generated by the calibration process, so that each pixel now has an identical response to the same illumination.

Start the Flat Field calibration tool via the CamExpert menu bar:
**Tools • Flat Field Correction • Calibration**.

### Flat Field Calibration Window

The Flat Field calibration window provides a three step process to acquire two reference images and then save the flat field correction data for the camera used. To aid in determining if the reference images are valid, a histogram tool is provided so that the user can review the images used for the correction data.




- Setup the camera to capture a uniform black image. Black paper with no illumination and the camera lens' iris closed to minimum can provide such a black image.
- Click on **Acquire Black Image**. The flat field demo will grab a video frame, analyze the pixel gray level spread, and present the statistics. The desired black reference image should have pixel values less then 20. If acceptable accept the image as the black reference.
- Setup the camera to acquire a uniform white image (but not saturated white). Even illumination on white paper can be used, with a gray level of 128 minimum. It is preferable to prepare for the white level calibration before the calibration procedure.
- Click on **Acquire White Image**. The flat field demo will grab a video frame, analyze the pixel gray level spread, and present the statistics. The captured gray level for all pixels should be greater than 128. If acceptable accept the image as the white reference.
- Click on **Save**. The flat field correction data is saved as a TIF image with a file name of your choice (such as camera name and serial number).

## Using Flat Field Correction

From the CamExpert menu bar enable Flat Field correction (**Tools • Flat Field Correction • Enable**). Now when doing a live grab or snap, the incoming image is corrected by the current flat field calibration data for each pixel.

Use the menu function **Tools • Flat Field Correction • Load** to load in a flat field correction image from a previous saved calibration data. CamExpert allows saving and loading calibration data for all cameras used with the imaging system.

# Using the Bayer Filter Tool

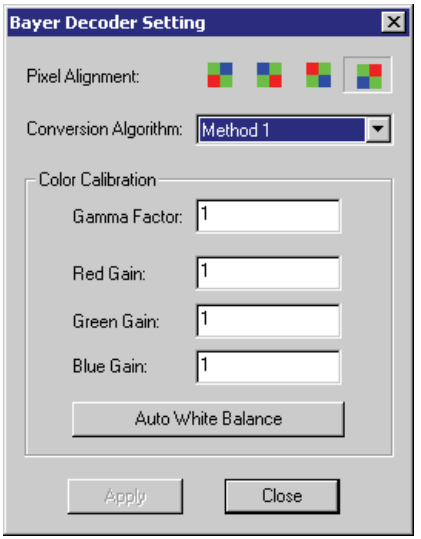
CamExpert supports the use of Bayer Filter cameras by providing a tool to select the Bayer filter mosaic pattern and to perform an auto white balance. Color calibration can then be manually fine tuned with RGB gain and gamma adjustments.

The CamExpert Bayer filter tool supports using both software or hardware based decoding. With boards that have Bayer filter decoding in hardware, such as the X64-CL Express with its Bayer decoding configuration, CamExpert directly controls the hardware for high performance real-time acquisitions from Bayer filter cameras. When standard acquisition boards are used, CamExpert performs software Bayer filter decoding using the host system.

## Bayer Filter White Balance Calibration Procedure

The following procedure uses an X64-CL Express with hardware Bayer filter support (i.e. Medium Camera Link Bayer Decoder firmware loaded) and any supported Bayer color camera. It is assumed that CamExpert was used to generate a camera file with correct camera timing parameters.

- On the CamExpert menu bar, click on **Tools • Bayer Filter**. The following menu should show **Hardware** selected by default when the X64-CL Express has Bayer support.
- Select **Setting** to access the color calibration window (see following figure).




- Click **Grab** to start live acquisition.
- Aim and focus the camera. The camera should see an area of white or place white paper in front of the object being imaged.
- Click on one of the four Bayer pixel alignment patterns to match the camera (best color before calibration). Typically the CamExpert default is correct for a majority of cameras.
- Adjust the lens iris to reduce the exposure brightness so that the white image area is now darker. Make certain that no pixel in the white area is saturated.
- Using the mouse left button, click and drag a ROI enclosing a portion of the white area.
- Click on the **Auto White Balance** button. CamExpert will make RGB gain adjustments.
- Open the camera iris to have a correctly exposed image.
- Review the image for color balance.
- Manually make additional adjustments to the RGB gain values. Fine tune the color balance to achieve best results. Adjust the gamma factor to additionally improve the display.
- Stop the live acquisition and save camera file (which now contains the Bayer RGB calibration information). Note that the gamma factor is not save because it is not a Sapera parameter but only a display tool.

## Using the Bayer Filter

A Sapera application, when loading the camera file parameters, will have the RGB gain adjustment values. The application can provide the calibration window to make RGB adjustments as required.

# Sapera Demo Applications

## Grab Demo Overview

| | |
|---|---|
| **Program** | **Start•Programs•DALSA Coreco•Sapera LT•Demos•Grab Demo** |
| **Program file** | **\DALSA Coreco\Sapera\Demos\Classes\vc\GrabDemo\Release\GrabDemo.exe** |
| **Workspace** | **\DALSA Coreco\Sapera\Demos\Classes\vc\SapDemos.dsw** |
| **.NET Solution** | **\DALSA Coreco\Sapera\Demos\Classes\vc\SapDemos_2003.sln** |
| **Description** | This program demonstrates the basic acquisition functions included in the Sapera library. The program allows you to acquire images, either in continuous or in one-shot mode, while adjusting the acquisition parameters. The program code may be extracted for use within your own application. |
| **Remarks** | This demo is built using Visual C++ 6.0 using the MFC library. It is based on Sapera C++ classes. See the Sapera User's and Reference manuals for more information. |

### Using the Grab Demo

#### Server Selection

Run the grab demo from the start menu **Start•Programs•Sapera LT•Demos•Grab Demo**.

The demo program first displays the acquisition configuration menu. The first drop menu displayed permits selecting from any installed Sapera acquisition servers (installed DALSA Coreco acquisition hardware using Sapera drivers). The second drop menu permits selecting from the available input devices present on the selected server.

#### CCF File Selection

The acquisition configuration menu is also used to select the required camera configuration file for the connected camera. Sapera camera files contain timing parameters and video conditioning parameters. The default folder for camera configuration files is also used by the CamExpert utility to save user generated or modified camera files.

Use the Sapera CamExpert utility program to generate the camera configuration file based on timing and control parameters entered. The CamExpert live acquisition window allows immediate verification of those parameters. CamExpert reads both Sapera *.cca and *.cvi for backward compatibility with the original Sapera camera files.

### Grab Demo Main Window

Refer to the Sapera LT User's Manual (OC-SAPM-USER), in section "Demos and Examples – Acquiring with Grab Demo", for more information on the Grab Demo.

# Flat-Field Demo Overview

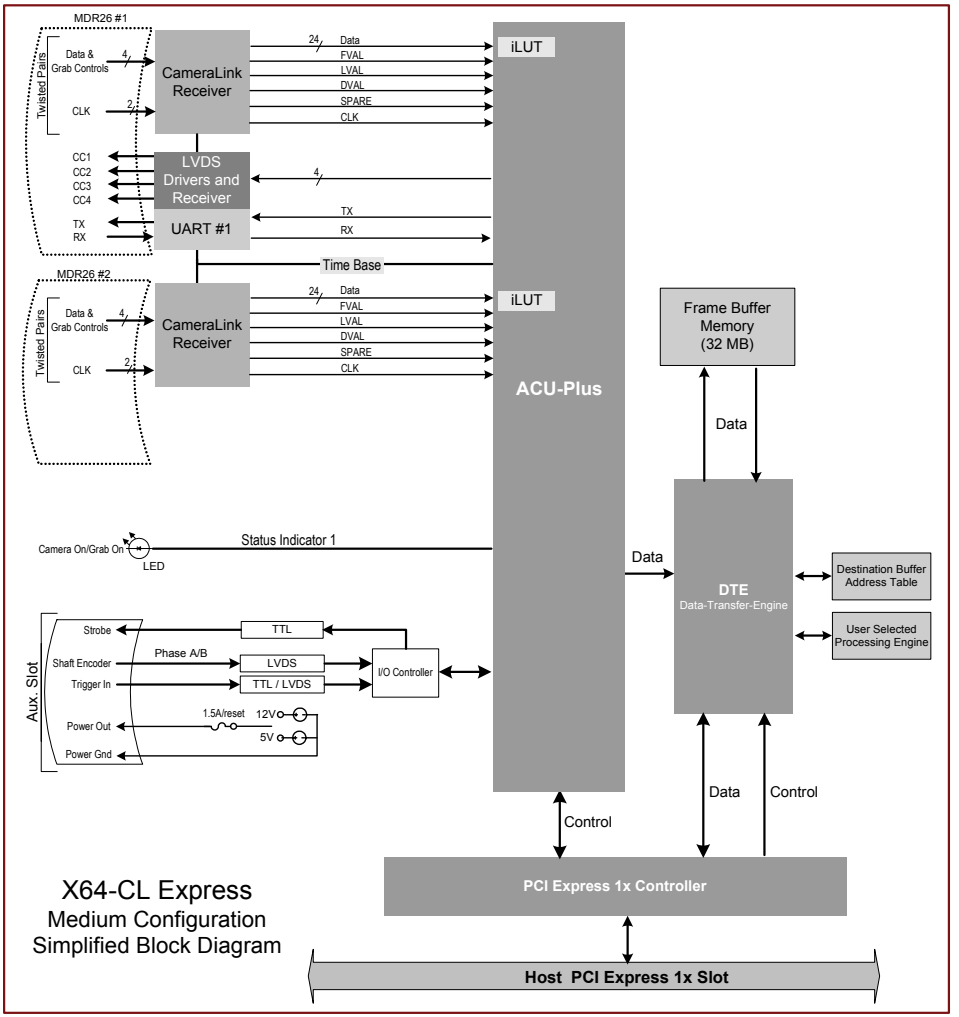
| | |
|---|---|
| **Program** | **Start•Programs•Sapera LT•Demos•Sapera++•Flat Field Demo** |
| **Program file** | **\Coreco\Sapera\Demos\Classes\vc\FlatFieldDemo\Release\FlatfieldDemo.exe** |
| **Workspace** | **\Coreco\Sapera\Demos\Classes\vc\SapDemos.dsw** |
| **Description** | This program demonstrates Flat Field or Flat Line processing, either performed by supporting DALSA Coreco hardware or performed on the host system via the Sapera library. The program allows you to acquire a flat field or flat line reference image, and then do real time correction either in continuous or single acquisition mode. The program code may be extracted for use within your own application. |
| **Remarks** | This demo is built using Visual C++ 6.0 using the MFC library. It is based on Sapera C++ classes. See the Sapera User's and Reference manuals for more information. |

## Using the Flat Field Demo

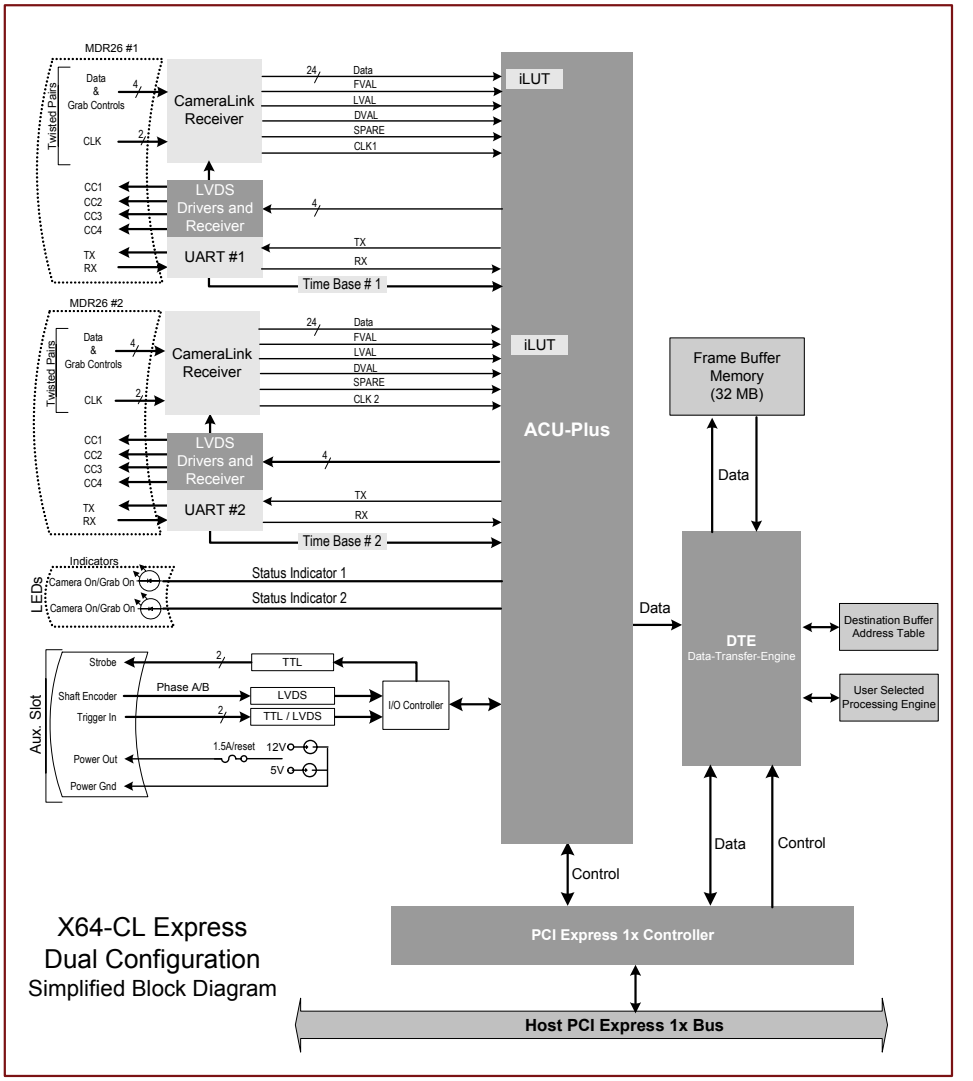
Refer to the Sapera LT User's Manual (OC-SAPM-USER), in section "Using the Flat Field Demo", for more information.

# X64-CL Express Reference

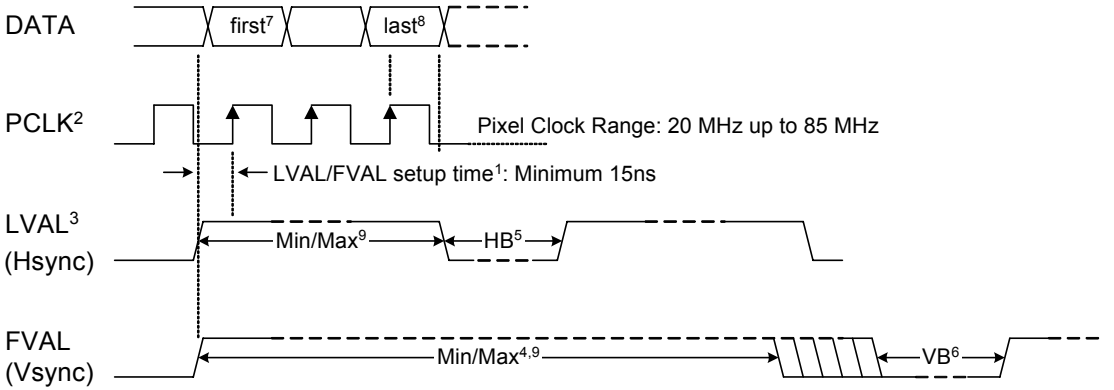
## X64-CL Express Medium Block Diagram



# X64-CL Express Dual Base Block Diagram



# X64-CL Express Acquisition Timing




- [1] The setup times for LVAL and FVAL are the same. Both must be high and stable before the rising edge of the Pixel Clock.
- [2] Pixel Clock must always be present.
- [3] LVAL must be active high to acquire camera data.
- [4] Minimum of 1.
- [5] HB - Horizontal Blanking:

  Minimum: 4 clocks/cycle
  Maximum: no limits

- [6] VB - Vertical Blanking:

  Minimum: 1 line
  Maximum: no limits

- [7] First Active Pixel (unless otherwise specified in the CCA file – "Horizontal Back invalid = x" where 'x' defines the number of pixels to be skipped).
- [8] Last Active Pixel – defined in the CCA file under "Horizontal active = y" – where 'y' is the total number of active pixels per tap.
- [9] Maximum Valid Data:
  - 8-bits/pixel x 256K Pixels/line (LVAL)
  - 16-bits/pixel x 128K Pixels/line (LVAL)
  - 32-bits/pixel x 64K Pixels/line (LVAL)
  - 64-bits/pixel x 32K Pixels/line (LVAL)
  - 16,000,000 lines (FVAL)

# Line Trigger Source Selection for Linescan Applications

Linescan imaging applications require some form of external event trigger to synchronize linescan camera exposures to the moving object. This synchronization signal is either an external trigger source (one exposure per trigger event) or a shaft encoder source composed of a single or dual phase (quadrature) signal. The X64-CL Express shaft encoder inputs provide additional functionality with pulse drop or pulse multiply support.

The following table describes the line trigger source types supported by the X64-CL Express. Refer to the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00) for descriptions of the Sapera parameters.

## CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE – Parameter Values Specific to the X64-CL series

| PRM Value | Active Shaft Encoder Input |
|---|---|
| 0 | Default |
| 1 | Use phase A |
| 2 | Use phase B |
| 3 | Use phase A & B |

## CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE full description relative to trigger type and X64-CL Express configuration used:

| PRM Value | X64-CL Express configuration & camera input used | External Line Trigger Signal used | External Shaft Encoder Signal used |
|---|---|---|---|
| | | *if* CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE = *true* | *if* CORACQ_PRM_SHAFT_ENCODER_ENABLE =*true* |
| **0** | Dual - Camera #1 | Shaft Encoder Phase A | Shaft Encoder Phase A |
| | Dual - Camera #2 | Shaft Encoder Phase B | Shaft Encoder Phase B |
| | Medium - Camera #1 | Shaft Encoder Phase A | Shaft Encoder Phase A & B |
| **1** | Dual - Camera #1 | Shaft Encoder Phase A | Shaft Encoder Phase A |
| | Dual - Camera #2 | Shaft Encoder Phase A | Shaft Encoder Phase A |
| | Medium - Camera #1 | Shaft Encoder Phase A | Shaft Encoder Phase A |
| **2** | Dual - Camera #1 | Shaft Encoder Phase B | Shaft Encoder Phase B |
| | Dual - Camera #2 | Shaft Encoder Phase B | Shaft Encoder Phase B |
| | Medium - Camera #1 | Shaft Encoder Phase B | Shaft Encoder Phase B |
| **3** | Dual - Camera #1 | n/a | Shaft Encoder Phase A & B |
| | Dual - Camera #2 | n/a | Shaft Encoder Phase A & B |
| | Medium - Camera #1 | n/a | n/a – use prm value = 0 |

See "J4: External Signals Connector " on page **81** for shaft encoder input connector details.

## CVI/CCF File Parameters Used

- External Line Trigger Source = prm value
- External Line Trigger Enable = true/false
- Shaft Encoder Enable = true/false

# Shaft Encoder Interface Timing

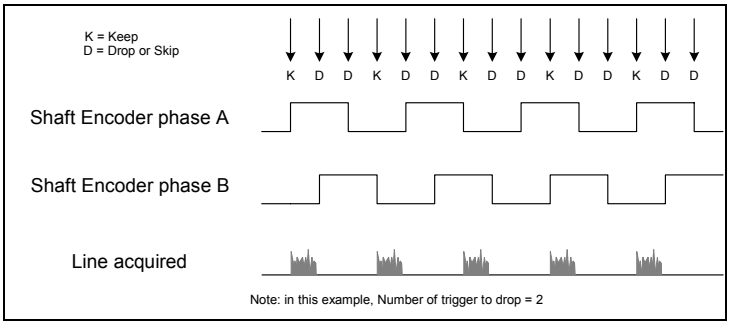
## Connector J4, Dual Balanced Shaft Encoder Inputs:

- Input 1: Pin 15 (Phase A +) & Pin 16 (Phase A -)
  (see "J4: External Signals Connector " on page **81** for complete connector signal details)
- Input 2: Pin 17 (Phase B +) & Pin 18 (Phase B -)
- For X64-CL or X64-EM rev. A0, A1: use External Signals Connector cable assembly OC-64CC-0TIO1
  (see "X64-CL Express: External Signals Connector Bracket Assembly" on page **84** for pinout)

Web inspection systems with variable web speeds typically provide one or two synchronization signals from a web mounted encoder to coordinate trigger signals. These trigger signals are used by the acquisition linescan camera. The X64-CL Express supports single or dual shaft encoder signals. Dual encoder signals are typically 90 degrees out of phase relative to each other and provide greater web motion resolution.

Important: When using only one shaft encoder input phase, say phase A, then the phase B inputs must be terminated by connecting the + input to a voltage a minimum of 100 mV positive relative to the – input.

When enabled, the camera is triggered and acquires one scan line for each shaft encoder pulse edge. To optimize the web application, a second Sapera parameter defines the number of triggers to skip between valid acquisition triggers. The figure below depicts a system where a valid camera trigger is any pulse edge from either shaft encoder signal. After a trigger the two following triggers are ignored (as defined by the Sapera pulse drop parameter).




Note that camera file parameters are best modified by using the Sapera CamExpert program.

### CVI/CCF File Parameters Used

**Shaft Encoder Enable** = X, where:

- If X = 1, Shaft Encoder is enabled
- If X = 0, Shaft Encoder is disabled

**Shaft Encoder Pulse Drop** = X, where:

- X = number of trigger pulses ignored between valid triggers

---

For information on camera configuration files see the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

---

# Virtual Frame_Reset for Linescan Cameras

When using linescan cameras a frame buffer is allocated in host system memory to store captured video lines. To control when a video line is stored as the first line in this "virtual" frame buffer, an external frame trigger signal called **FRAME_RESET** is used. The number of lines sequentially grabbed and stored in the virtual frame buffer is controlled by the Sapera vertical cropping parameter.

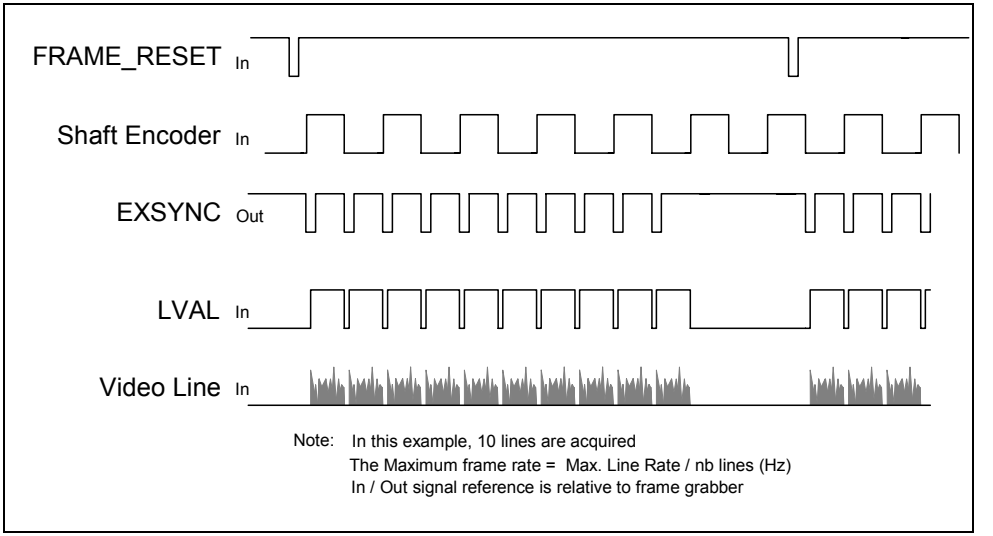
## Virtual Frame_Reset Timing Diagram

The following timing diagram shows an example of grabbing 10 video lines from a linescan camera and the use of **FRAME_RESET** to define when a video line is stored at the beginning of the virtual frame buffer. The **FRAME_RESET** signal (generated by some external event) is input on the X64-CL Express trigger input.

- **FRAME_RESET** can be TTL or LVDS and be rising or falling edge active.
- **FRAME_RESET** control is configured for rising edge trigger in this example.
- **FRAME_RESET** connects to the X64-CL Express via the Trigger In 1 balanced inputs on connector J4 pin 11 (+) and 12 (-).
- After the X64-CL Express receives **FRAME_RESET**, the **EXSYNC** control signal is output to the camera to trigger n lines of video as per the defined virtual frame size.
- The **EXSYNC** control signal is either based on timing controls input on one or both X64-CL Express shaft encoder inputs (see "J4: External Signals Connector " on page **81** pinout) or an internal X64-CL Express clock.
- The number of lines captured is specified by the Sapera vertical cropping parameter.

### Synchronization Signals for a Virtual Frame of 10 Lines.

The following timing diagram shows the relationship between external Frame_Reset input, external Shaft Encoder input (one phase used with the second terminated), and EXSYNC out to the camera.




## CVI File (VIC) Parameters Used

The VIC parameters listed below provide the control functionality for virtual frame reset. Applications either load pre-configured .cvi files or change VIC parameters directly during runtime.

Note that camera file parameters are best modified by using the Sapera CamExpert program.

**External Frame Trigger Enable** = X, where: \\Virtual Frame_Reset enabled

- If X = 1, External Frame Trigger is enabled
- If X = 0, External Frame Trigger is disabled

**External Frame Trigger Detection** = Y, where: \\ Frame_Reset edge select

- If Y= 4, External Frame Trigger is active on rising edge
- If Y= 8, External Frame Trigger is active on falling edge

**External Frame Trigger Level** = Z, where: \\ Frame_Reset signal type

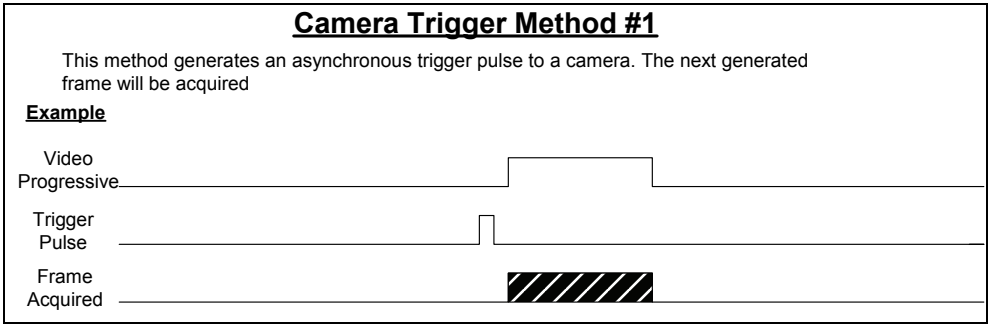
- If Z= 2, External Frame Trigger is a RS-422/LVDS signal

For information on camera files see the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Acquisition Methods

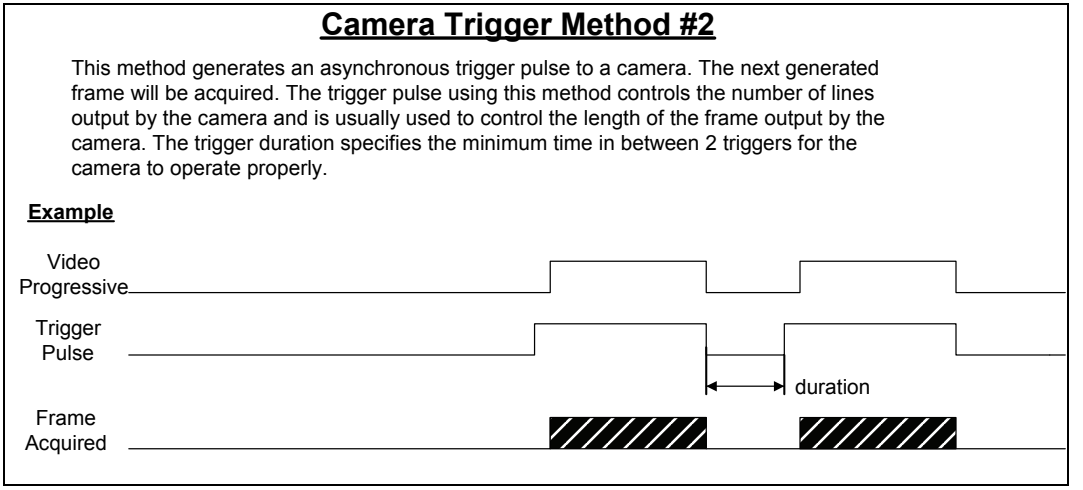
Refer to the Sapera LT Acquisition Parameters Reference manual (OC-SAPM-APR00) for additional information concerning camera control methods.

## Camera Trigger Method #1



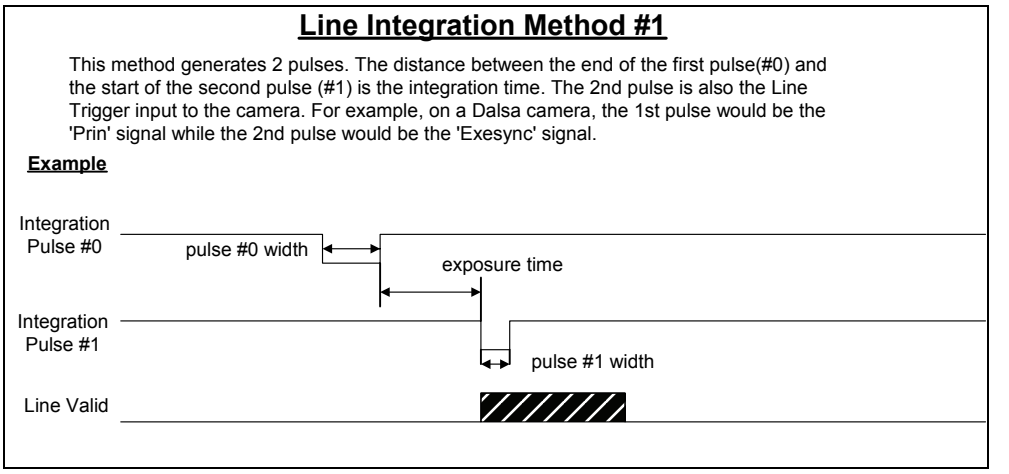

Refer to **CORACQ_VAL_CAM_TRIGGER_METHOD_1** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Camera Trigger Method #2



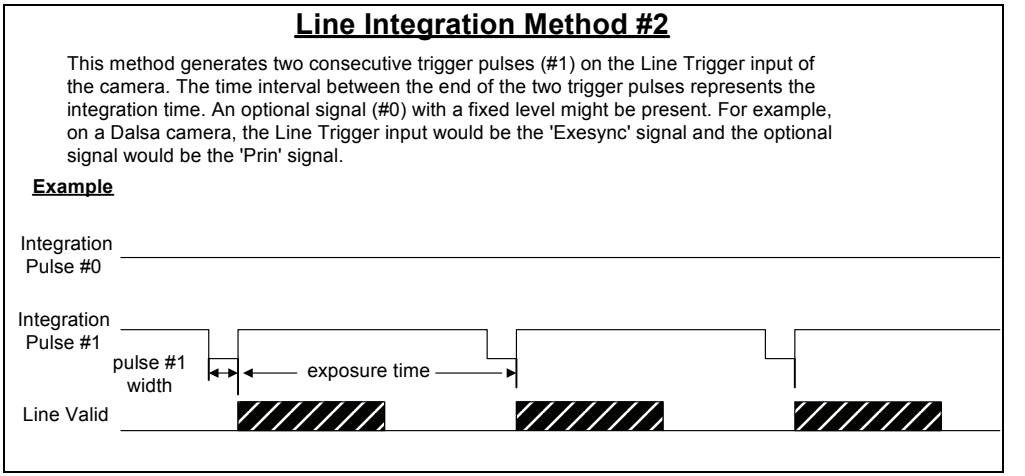

Refer to **CORACQ_VAL_CAM_TRIGGER_METHOD_2** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Line Integration Method #1



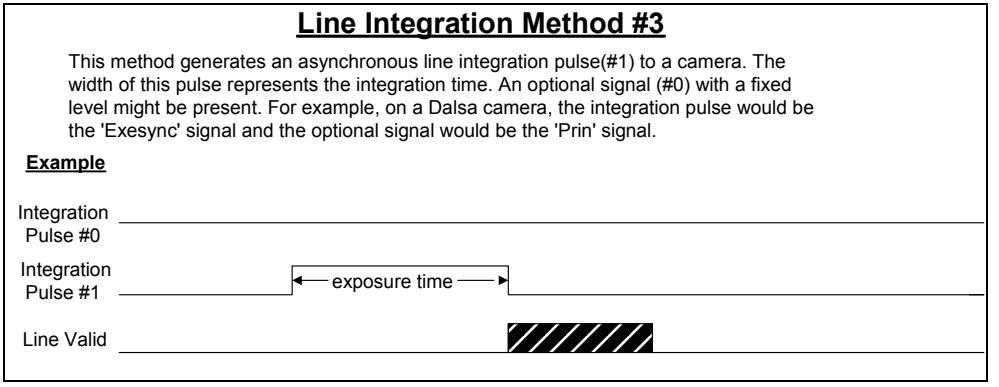

Refer to **CORACQ_VAL_LINE_INTEGRATE_METHOD_1** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Line Integration Method #2



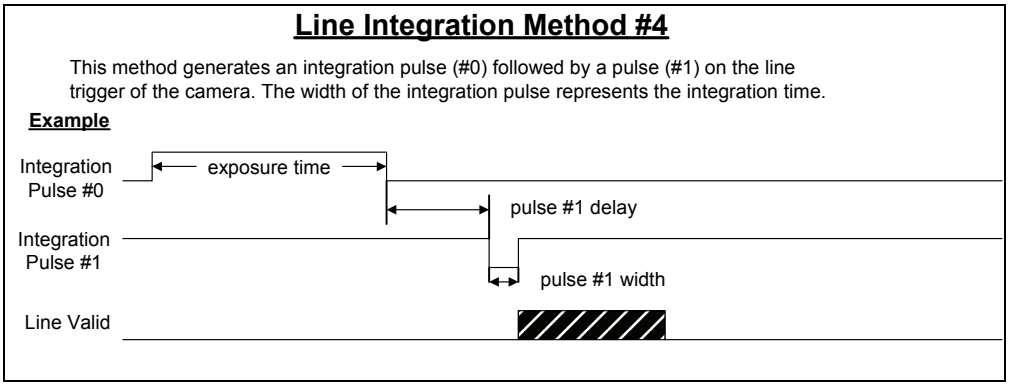

Refer to **CORACQ_VAL_LINE_INTEGRATE_METHOD_2** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Line Integration Method #3



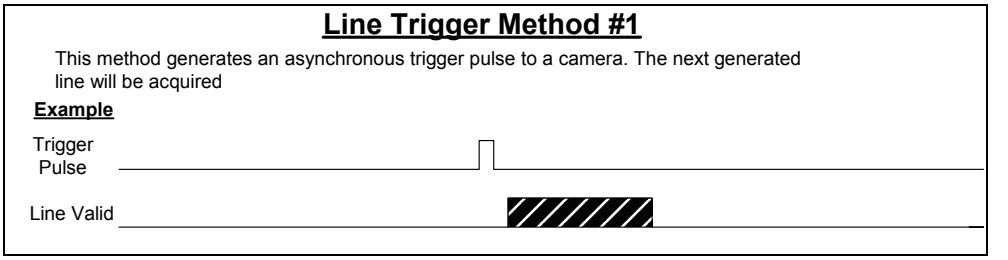

Refer to **CORACQ_VAL_LINE_INTEGRATE_METHOD_3** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Line Integration Method #4



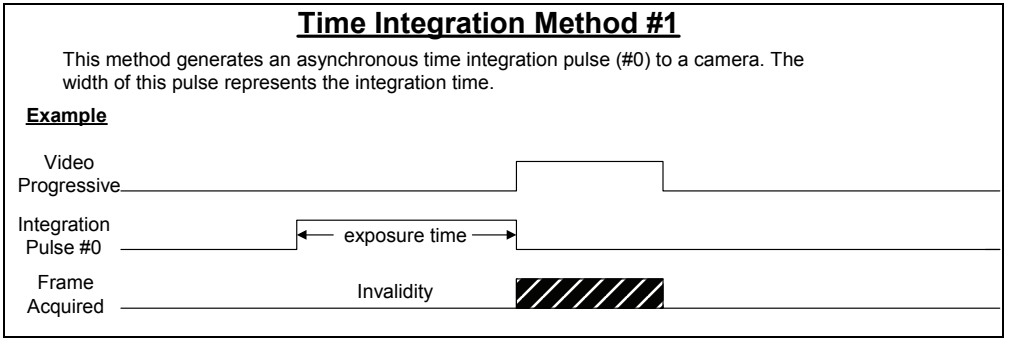

Refer to **CORACQ_VAL_LINE_INTEGRATE_METHOD_4** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Line Trigger Method #1



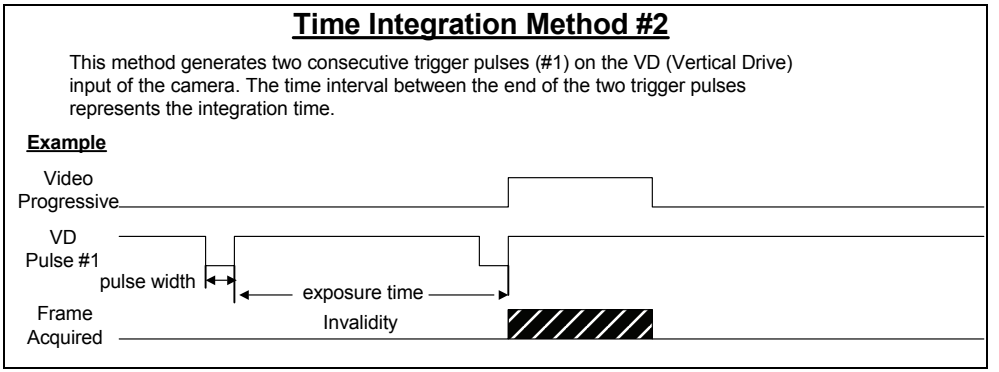

Refer to **CORACQ_VAL_LINE_TRIGGER_METHOD_1** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Time Integration Method #1



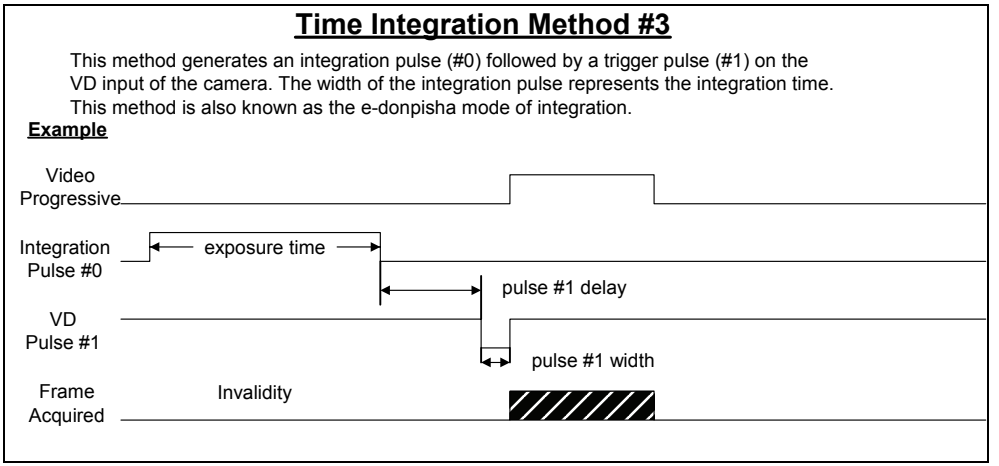

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_1** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Time Integration Method #2



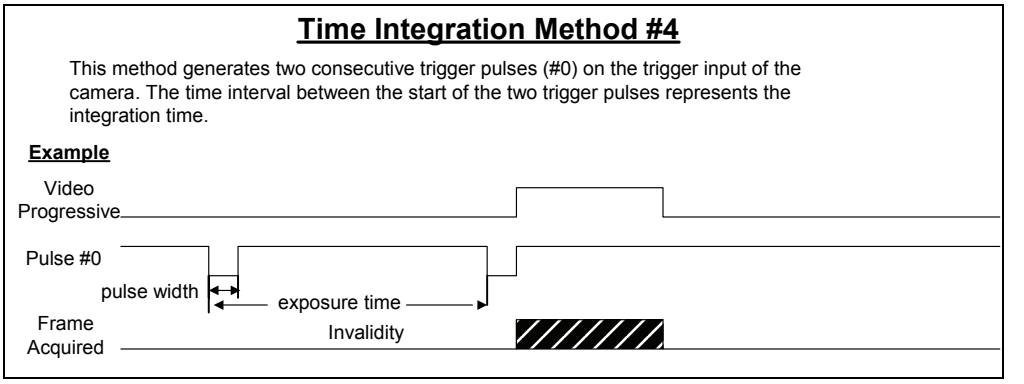

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_2** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Time Integration Method #3



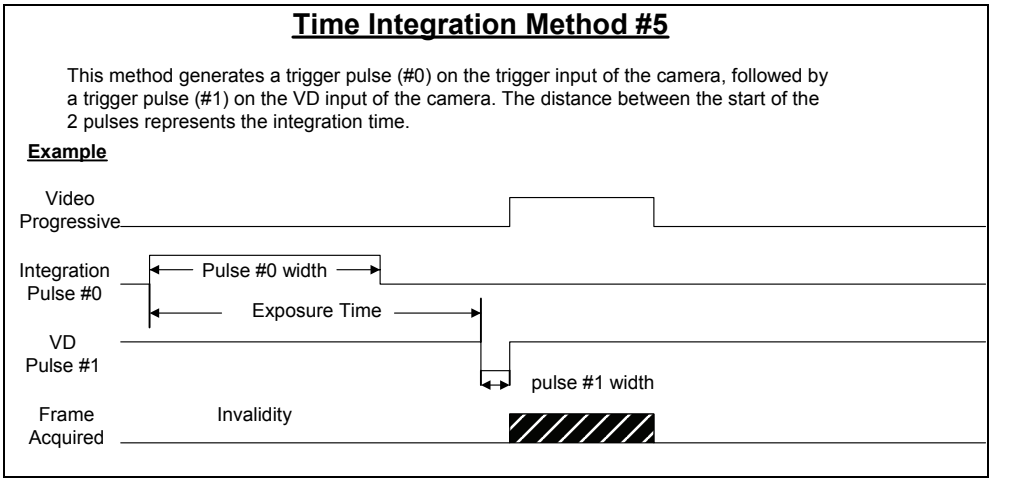

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_3** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Time Integration Method #4



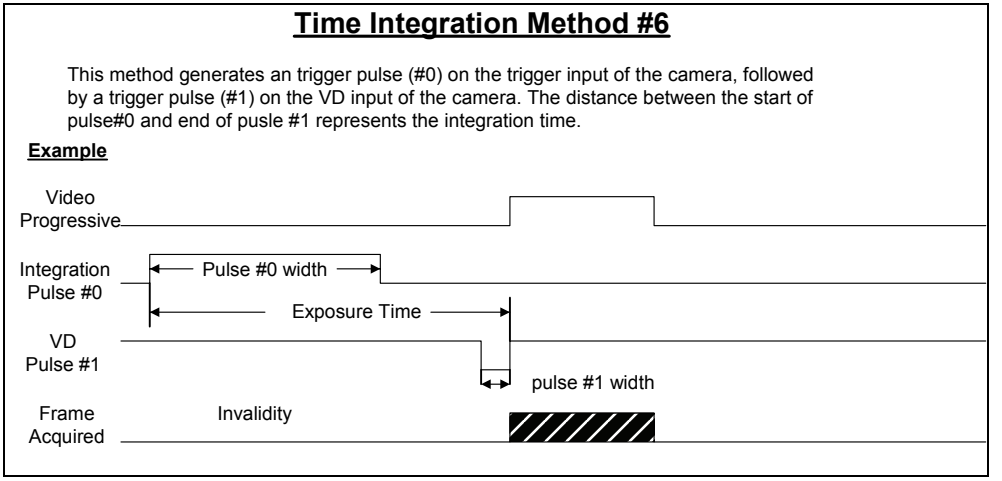

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_4** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Time Integration Method #5



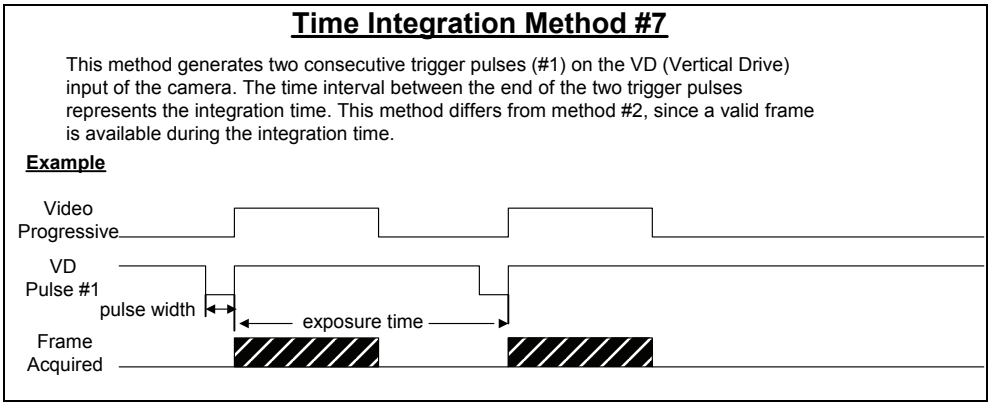

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_5** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Time Integration Method #6



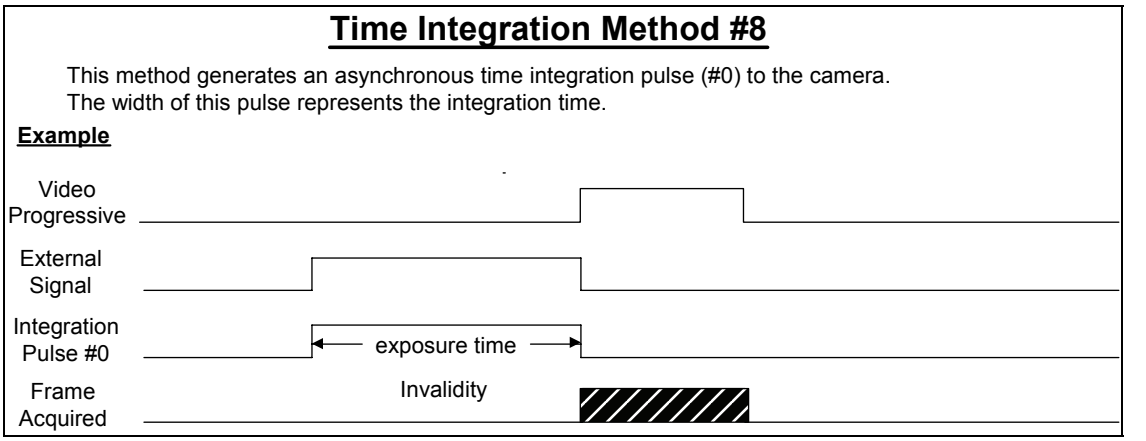

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_6** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# Time Integration Method #7



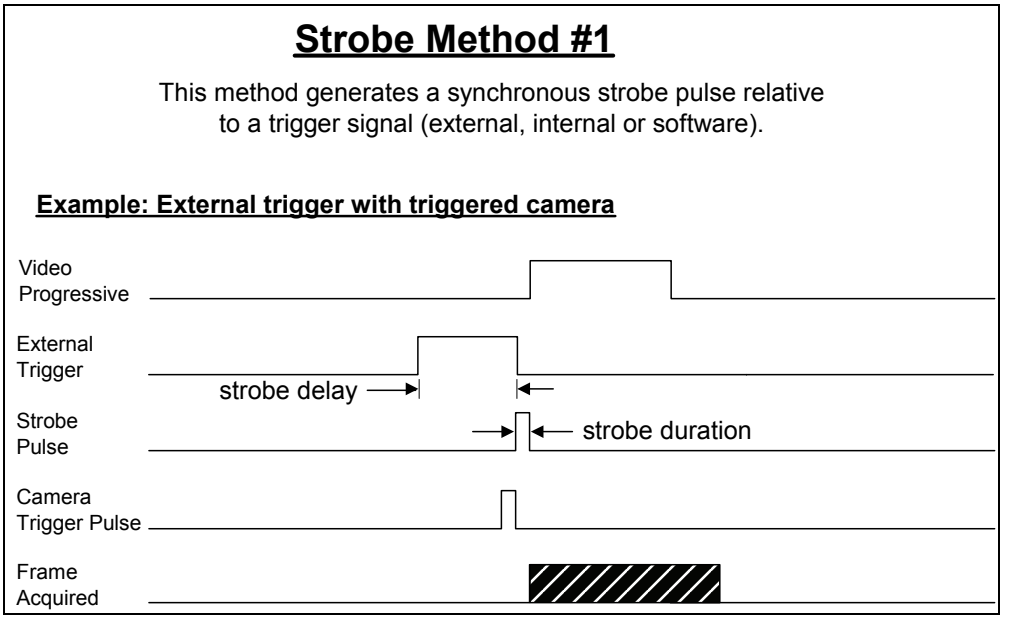

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_7** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Time Integration Method #8



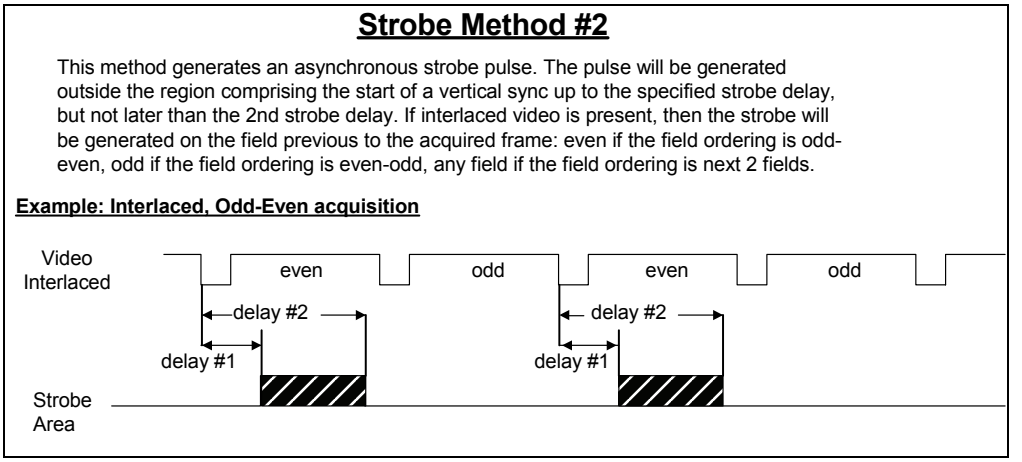

Refer to **CORACQ_VAL_TIME_INTEGRATE_METHOD_8** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Strobe Method #1



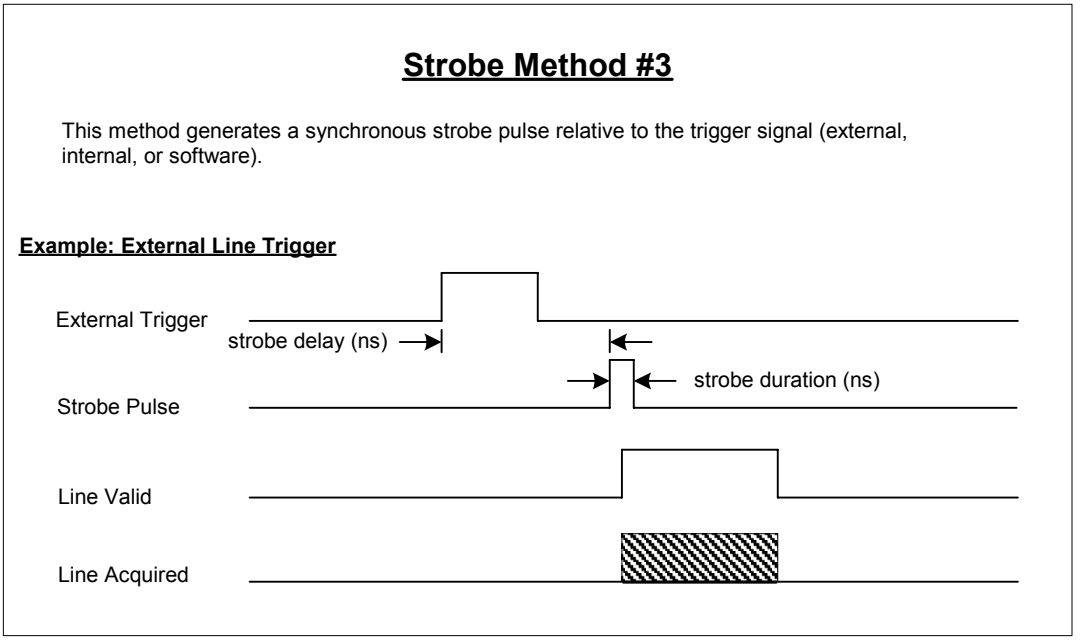

Refer to **CORACQ_VAL_STROBE_METHOD_1** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Strobe Method #2



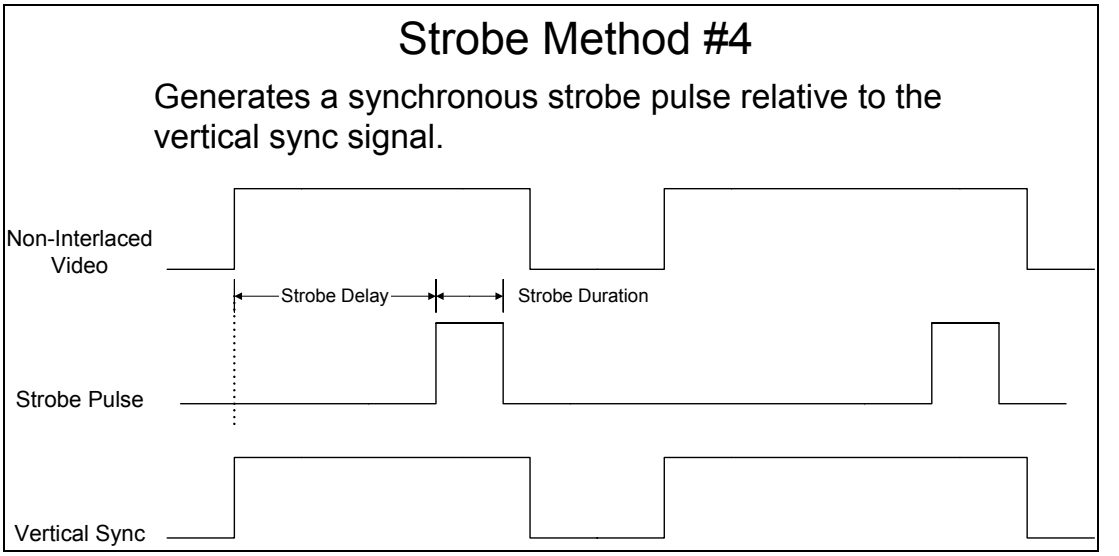

Refer to **CORACQ_VAL_STROBE_METHOD_2** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Strobe Method #3




Refer to **CORACQ_VAL_STROBE_METHOD_3** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

## Strobe Method #4




Refer to **CORACQ_VAL_STROBE_METHOD_4** in the Sapera Acquisition Parameters Reference Manual (OC-SAPM-APR00).

# X64-CL Express LUT availability

The following table defines the X64-CL Express input LUT availability.

| Number of Digital Bits | Number of Taps Medium | Number of Taps Dual Base | Output Pixel Format | LUT Format | Notes |
|---|---|---|---|---|---|
| 8 | 4 | 3 | MONO 8 | | Yes |
| 8 | 4 | 3 | MONO 16 | - | Not Supported |
| 10 | 4 | 2 | MONO 8 | | Yes |
| 10 | 4 | 2 | MONO 16 | | 10 bits in 10 LSBs of 16-bit |
| 12 | 4 | 2 | MONO 8 | 12-in, 8-out | 8 MSB |
| 12 | 4 | 2 | MONO 16 | 12-in, 12-out | 12 bits in 12 LSBs of 16-bit |
| 14 | - | - | MONO 8 | - | Not Supported |
| 14 | - | - | MONO 16 | - | Not Supported |
| 16 | - | - | MONO 8 | - | Not Supported |
| 16 | - | - | MONO 16 | - | Not Supported |
| 8 x 3 (RGB) | 2 | 1 | RGB8888 | 3 x 8 bit | Yes |
| 8 x 3 (RGB) | 1 | 1 | RGB101010 | - | Not Supported |
| 10 x 3 (RGB) | 1 | - | RGB888 | | Medium only |
| 10 x 3 (RGB) | 1 | - | RGB101010 | | Medium only |

# X64-CL Express Sapera Capabilities

The three tables below describe the Sapera capabilities supported by the X64-CL Express. Unless specified, each capability applies to both boards or all mode configurations and all acquisition modes. Sapera capabilities not used have been omitted for clarity.

The board capabilities listing is subject to change. Capabilities should be verified by the application because new board driver releases may change certain capabilities.

Specifically the X64-CL Express family is described in Sapera as:

- Board Server: X64-CL_Express_1
- Acquisition Device: Camera Link
- Modes (X64-CL Express): Medium Mono or Medium Color RGB with Flat Field correction
- Modes (X64-CL Express): Medium Mono with Bayer mosaic filter decoding
- Modes (X64-CL Express): Dual Base Mono with Flat Field correction

## Camera Related Capabilities

| Capability | Value or Limits or Bitfield | Capability Description / Details |
|---|---|---|
| CORACQ_CAP_CAM_RESET | 1 | (0x1) True; reset is supported |
| CORACQ_CAP_CAM_RESET_DURATION_MAX | 65535000 | 0x3e7fc18 |
| CORACQ_CAP_CAM_RESET_DURATION_MIN | 1 | 0x1 |
| CORACQ_CAP_CAM_RESET_METHOD | 1 | 0x1; reset method 1 |
| CORACQ_CAP_CAM_RESET_POLARITY | 00000011b | (0x1) active low<br>(0x2) active high |
| CORACQ_CAP_CAM_TRIGGER | 1 | (0x1) True; trigger is supported |
| CORACQ_CAP_CAM_TRIGGER_DURATION_MAX | 65535000 | 0x3e7fc18 |
| CORACQ_CAP_CAM_TRIGGER_DURATION_MIN | 1 | 0x1 |
| CORACQ_CAP_CAM_TRIGGER_METHOD | 00000011b | (0x1) camera trigger method 1<br>(0x2) camera trigger method 2 |
| CORACQ_CAP_CAM_TRIGGER_POLARITY | 00000011b | (0x1) active low<br>(0x2) active high |
| CORACQ_CAP_CAMLINK_CONFIGURATION | 00000001b | (0x1) Base configuration (mono or RGB) |
| CORACQ_CAP_CAMLINK_CONFIGURATION | 00000011b | (0x1) Base configuration<br>(0x2) Medium configuration<br>(mono or RGB) |
| CORACQ_CAP_CHANNEL | 1 | (0x1) CORACQ_VAL_CHANNEL_SINGLE |
| CORACQ_CAP_CHANNEL | 00000011 | (0x1) CORACQ_VAL_CHANNEL_SINGLE<br>(0x2) CORACQ_VAL_CHANNEL_DUAL |
| CORACQ_CAP_CHANNELS_ORDER | 00000011b | (0x1) CORACQ_VAL_CHANNELS_ORDER_NORMAL<br>(0x2) CORACQ_VAL_CHANNELS_ORDER_REVERSE |

| | | |
|---|---|---|
| CORACQ_CAP_DATA_VALID_ENABLE | 1 | (0x1) True<br>The camera data valid signal is supported |
| CORACQ_CAP_DATA_VALID_POLARITY | 00000010b | (0x2) Data valid signal active high |
| CORACQ_CAP_FIELD_ORDER | 00000100b | (0x4) CORACQ_VAL_FIELD_ORDER_NEXT_FIELD |
| CORACQ_CAP_FRAME | 00000010b | (0x2) CORACQ_VAL_FRAME_PROGRESSIVE |
| CORACQ_CAP_HACTIVE_MAX | 16777215 | (0xffffff) maximum pixels per tap |
| CORACQ_CAP_HACTIVE_MIN | 1 | (0x01) minimum pixel per tap |
| CORACQ_CAP_HACTIVE_MULT | 1 | 0x1 |
| CORACQ_CAP_HBACK_INVALID_MAX | 16777215 | 0xffffff |
| CORACQ_CAP_HBACK_INVALID_MIN | 0 | 0x0 |
| CORACQ_CAP_HBACK_INVALID_MULT | 1 | 0x1 |
| CORACQ_CAP_HFRONT_INVALID_MAX | 16777215 | 0xffffff |
| CORACQ_CAP_HFRONT_INVALID_MIN | 0 | 0x0 |
| CORACQ_CAP_HFRONT_INVALID_MULT | 1 | 0x1 |
| CORACQ_CAP_HSYNC_MAX | 4294967295 | 0xffffffff |
| CORACQ_CAP_HSYNC_MIN | 1 | (0x1) minimum pixel per tap |
| CORACQ_CAP_HSYNC_MULT | 1 | 0x1 |
| CORACQ_CAP_HSYNC_POLARITY | 1 | (0x1) Horizontal sync pulse is active low |
| CORACQ_CAP_INTERFACE | 00000010b | (0x2) CORACQ_VAL_INTERFACE_DIGITAL |
| CORACQ_CAP_LINE_INTEGRATE | 1 | (0x1) True<br>At least one method of line integration is supported |
| CORACQ_CAP_LINE_INTEGRATE_METHOD | 00001111b | (0x01) Line Integration Method #1<br>(0x02) Line Integration Method #2<br>(0x04) Line Integration Method #3<br>(0x08) Line Integration Method #4 |
| CORACQ_CAP_LINE_INTEGRATE_PULSE0_DELAY_MAX | 65535 | 0xffff (in pixels) |
| CORACQ_CAP_LINE_INTEGRATE_PULSE0_DELAY_MIN | 0 | 0x0 |
| CORACQ_CAP_LINE_INTEGRATE_PULSE0_DURATION_MAX | 65535000 | 0x3e7fc18 (in pixels) |
| CORACQ_CAP_LINE_INTEGRATE_PULSE0_DURATION_MIN | 1 | 0x1 |
| CORACQ_CAP_LINE_INTEGRATE_PULSE0_POLARITY | 00000011b | (0x1) Line integration trigger pulse is active low<br>(0x2) Line integration trigger pulse is active high. |
| CORACQ_CAP_LINE_INTEGRATE_PULSE1_DELAY_MAX | 65535000 | 0x3e7fc18 (in pixels) |
| CORACQ_CAP_LINE_INTEGRATE_PULSE1_DELAY_MIN | 0 | 0x0 |
| CORACQ_CAP_LINE_INTEGRATE_PULSE1_DURATION_MAX | 65535000 | 0x3e7fc18 (in pixels) |
| CORACQ_CAP_LINE_INTEGRATE_PULSE1_DURATION_MIN | 1 | 0x1 |
| CORACQ_CAP_LINE_INTEGRATE_PULSE1_POLARITY | 00000011b | (0x1) Line integration trigger pulse is active low<br>(0x2) Line integration trigger pulse is active high. |
| CORACQ_CAP_LINE_TRIGGER | 1 | (0x1) True<br>At least one method of line trigger is supported |
| CORACQ_CAP_LINE_TRIGGER_DELAY_MAX | 65535 | (0xffff) |
| CORACQ_CAP_LINE_TRIGGER_DELAY_MIN | 0 | (0x0) |

| | | |
|---|---|---|
| CORACQ_CAP_LINE_TRIGGER_DURATION_MAX | 65535 | (0xffff) |
| CORACQ_CAP_LINE_TRIGGER_DURATION_MIN | 0 | (0x0) |
| CORACQ_CAP_LINE_TRIGGER_METHOD | 1 | (0x1) Line Trigger Method #1 |
| CORACQ_CAP_LINE_TRIGGER_POLARITY | 00000011b | (0x01) CORACQ_VAL_ACTIVE_LOW<br>(0x02) CORACQ_VAL_ACTIVE_HIGH |
| CORACQ_CAP_LINESCAN_DIRECTION | 0 | False<br>Line scan direction signal can not be controlled |
| CORACQ_CAP_LINESCAN_DIRECTION_POLARITY | 00000010b | (0x2) Forward direction scan signal is active high |
| CORACQ_CAP_PIXEL_CLK_DETECTION | 00000100b | (0x4) CORACQ_VAL_RISING_EDGE |
| CORACQ_CAP_PIXEL_CLK_EXT_MAX | 85000000 | 0x510ff40 |
| CORACQ_CAP_PIXEL_CLK_EXT_MIN | 20000000 | 0x1312d00 |
| CORACQ_CAP_PIXEL_CLK_INT_MAX | 85000000 | 0x510ff40 |
| CORACQ_CAP_PIXEL_CLK_INT_MIN | 20000000 | 0x1312d00 |
| CORACQ_CAP_PIXEL_CLK_SRC | 00000010b | (0x2) External pixel clock |
| CORACQ_CAP_PIXEL_DEPTH | 8 bit | |
| CORACQ_CAP_PIXEL_DEPTH | 10 bit | |
| CORACQ_CAP_PIXEL_DEPTH | 12 bit | |
| CORACQ_CAP_PIXEL_DEPTH | 14 bit | available with Base or Medium monochrome |
| CORACQ_CAP_PIXEL_DEPTH | 16 bit | available with Base or Medium monochrome |
| CORACQ_CAP_PIXEL_DEPTH_PER_TAP | 8 | |
| CORACQ_CAP_PIXEL_DEPTH_PER_TAP | 10 | available with medium Bayer configuration |
| CORACQ_CAP_PIXEL_DEPTH_PER_TAP | 12 | available with medium RGB configuration |
| CORACQ_CAP_SCAN | 00000011b | (0x1) CORACQ_VAL_SCAN_AREA<br>(0x2) CORACQ_VAL_SCAN_LINE (not with Bayer configuration) |
| CORACQ_CAP_SIGNAL | 00000010b | (0x2) CORACQ_VAL_SIGNAL_DIFFERENTIAL |
| CORACQ_CAP_SYNC | 00000100b | (0x4) Separate horizontal and vertical sync source. |
| CORACQ_CAP_TAP_DIRECTION | 01111111b | (0x01)CORACQ_VAL_TAP_DIRECTION_LR<br>(0x02)CORACQ_VAL_TAP_DIRECTION_RL<br>(0x04)CORACQ_VAL_TAP_DIRECTION_UD<br>(0x08)CORACQ_VAL_TAP_DIRECTION_DU<br>(0x10)CORACQ_VAL_TAP_DIRECTION_FROM_TOP<br>(0x20)CORACQ_VAL_TAP_DIRECTION_FROM_MID<br>(0x40)CORACQ_VAL_TAP_DIRECTION_FROM_BOT |
| CORACQ_CAP_TAP_OUTPUT | 00000111b | (0x01) CORACQ_VAL_TAP_OUTPUT_ALTERNATE<br>(0x02) CORACQ_VAL_TAP_OUTPUT_SEGMENTED<br>(0x04) CORACQ_VAL_TAP_OUTPUT_PARALLEL |
| CORACQ_CAP_TAPS | 6 | Bayer and Medium monochrome configuration |
| CORACQ_CAP_TAPS | 1 | Base or Medium RGB configuration |
| CORACQ_CAP_TAPS | 3 | Dual Base monochrome configuration |
| CORACQ_CAP_TIME_INTEGRATE | 1 | (0x1) True<br>At least one method of time integration is supported |

| | | |
|---|---|---|
| CORACQ_CAP_TIME_INTEGRATE_METHOD | 11111111b | (0x01) Time Integration Method #1<br>(0x02) Time Integration Method #2<br>(0x04) Time Integration Method #3<br>(0x08) Time Integration Method #4<br>(0x10) Time Integration Method #5<br>(0x20) Time Integration Method #6<br>(0x40) Time Integration Method #7<br>(0x80) Time Integration Method #8 |
| CORACQ_CAP_TIME_INTEGRATE_PULSE0_DELAY_MAX | 65535000 | 0x3e7fc18 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_PULSE0_DELAY_MIN | 0 | 0x0 |
| CORACQ_CAP_TIME_INTEGRATE_PULSE0_DURATION_MAX | 65535000 | 0x3e7fc18 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_PULSE0_DURATION_MIN | 1 | 0x1 |
| CORACQ_CAP_TIME_INTEGRATE_PULSE0_POLARITY | 00000011b | (0x1) Time integration trigger pulse is active low<br>(0x2) Time integration trigger pulse is active high. |
| CORACQ_CAP_TIME_INTEGRATE_PULSE1_DELAY_MAX | 65535000 | 0x3e7fc18 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_PULSE1_DELAY_MIN | 0 | 0x0 |
| CORACQ_CAP_TIME_INTEGRATE_PULSE1_DURATION_MAX | 65535000 | 0x3e7fc18 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_PULSE1_DURATION_MIN | 1 | 0x1 |
| CORACQ_CAP_TIME_INTEGRATE_PULSE1_POLARITY | 00000011b | (0x1) Time integration trigger pulse is active low<br>(0x2) Time integration trigger pulse is active high. |
| | | |
| CORACQ_CAP_VACTIVE_MAX | 16777215 | 0xffffff lines |
| CORACQ_CAP_VACTIVE_MIN | 1 | 0x1 line |
| CORACQ_CAP_VACTIVE_MULT | 1 | 0x1 |
| CORACQ_CAP_VBACK_INVALID_MAX | 16777215 | 0xffffff |
| CORACQ_CAP_VBACK_INVALID_MIN | 0 | 0x0 |
| CORACQ_CAP_VBACK_INVALID_MULT | 1 | 0x1 |
| CORACQ_CAP_VFRONT_INVALID_MAX | 16777215 | 0xffffff |
| CORACQ_CAP_VFRONT_INVALID_MIN | 0 | 0x0 |
| CORACQ_CAP_VFRONT_INVALID_MULT | 1 | 0x1 |
| | | |
| CORACQ_CAP_VIDEO | 00000001b | (0x1) CORACQ_VAL_VIDEO_MONO |
| CORACQ_CAP_VIDEO | 00001000b | (0x8) CORACQ_VAL_VIDEO_RGB |
| CORACQ_CAP_VIDEO | 00010000b | (0x10) CORACQ_VAL_VIDEO_BAYER |
| CORACQ_CAP_VIDEO_STD | 1 | (0x1) CORACQ_VAL_VIDEO_STD_NON_STD |
| | | |
| CORACQ_CAP_VSYNC_MAX | 4294967295 | 0xffffffff |
| CORACQ_CAP_VSYNC_MIN | 0 | 0x0 |
| CORACQ_CAP_VSYNC_MULT | 1 | 0x1 |
| CORACQ_CAP_VSYNC_POLARITY | 1 | (0x1) Vertical sync pulse is active low |
| CORACQ_CAP_TIMESLOT | 3 | (0x04) CORACQ_VAL_TIMESLOT_3<br>3 clock cycles are needed to output 1 pixel from each tap |
| CORACQ_CAP_BAYER_ALIGNMENT | 00001111b | (0x1) CORACQ_VAL_BAYER_ALIGNMENT_GB_RG<br>(0x2) CORACQ_VAL_BAYER_ALIGNMENT_BG_GR<br>(0x4) CORACQ_VAL_BAYER_ALIGNMENT_RG_GB<br>(0x8) CORACQ_VAL_BAYER_ALIGNMENT_GR_BG |

# VIC Related Capabilities

| Capability | Value or Limits or Bitfield | Capability Description / Details |
|---|---|---|
| CORACQ_CAP_BIT_ORDERING | 1 | (0x1) Standard digital bit ordering |
| CORACQ_CAP_CAMSEL_MONO | 1 | (0x1) camera supported |
| CORACQ_CAP_CAMSEL_RGB | 1 | (0x1) camera supported |
| | | |
| CORACQ_CAP_CROP_HEIGHT_MAX | 16777215 | 0xffffff (in lines) |
| CORACQ_CAP_CROP_HEIGHT_MIN | 1 | 0x1 (lines) |
| CORACQ_CAP_CROP_HEIGHT_MULT | 1 | 0x1 (lines) |
| CORACQ_CAP_CROP_HORZ | 1 | (0x1) True, horizontal cropping is supported |
| CORACQ_CAP_CROP_LEFT_MAX | 16777215 | 0xffffff (in pixels) |
| CORACQ_CAP_CROP_LEFT_MIN | 0 | 0x0 (in pixels) |
| CORACQ_CAP_CROP_LEFT_MULT | 8 | 0x8 (pixels) |
| CORACQ_CAP_CROP_TOP_MAX | 16777215 | 0xffffff (in lines) |
| CORACQ_CAP_CROP_TOP_MIN | 0 | 0x0 (in lines) |
| CORACQ_CAP_CROP_TOP_MULT | 1 | 0x1 (lines) |
| CORACQ_CAP_CROP_VERT | 1 | (0x1) True, vertical cropping is supported |
| CORACQ_CAP_CROP_WIDTH_MAX | 16777215 | 0xffffff (in pixels) |
| CORACQ_CAP_CROP_WIDTH_MIN | 8 | 0x8 (pixels) |
| CORACQ_CAP_CROP_WIDTH_MULT | 8 | 0x8 (pixels) |
| | | |
| CORACQ_CAP_DECIMATE_METHOD | 00000000b | (0x00) No decimation |
| CORACQ_CAP_EXT_FRAME_TRIGGER | 1 | (0x1) True, external frame trigger is available |
| CORACQ_CAP_EXT_FRAME_TRIGGER_DETECTION | 01101111b | (0x01) Active low signal<br>(0x02) Active high signal<br>(0x04) Rising signal edge<br>(0x08) Falling signal edge<br>(0x20) Acquisition starts on rising edge of trigger 1 – ends on rising edge of trigger 2<br>(0x40) Acquisition starts on falling edge of trigger 1 – ends on falling edge of trigger 2 |
| CORACQ_CAP_EXT_FRAME_TRIGGER_LEVEL | 00000011b | (0x01), A TTL signal level<br>(0x02), A RS-422 signal level |
| CORACQ_CAP_EXT_LINE_TRIGGER | 1 | (0x1) True, external line trigger is available |
| CORACQ_CAP_EXT_LINE_TRIGGER_DETECTION | 00000100b | (0x04), Rising signal edge |
| CORACQ_CAP_EXT_LINE_TRIGGER_LEVEL | 00000010b | (0x02), A RS-422 signal level |
| CORACQ_CAP_EXT_LINE_TRIGGER_SOURCE | 4 | number of different external line trigger sources possible |
| | | |
| CORACQ_CAP_EXT_TRIGGER | 1 | (0x1) True, external trigger is available |
| CORACQ_CAP_EXT_TRIGGER_DETECTION | 00001111b | (0x01) An active low signal.<br>(0x02) An active high signal.<br>(0x04) The rising edge of the signal.<br>(0x08) The falling edge of the signal. |
| CORACQ_CAP_EXT_TRIGGER_DURATION_MAX | 65535 | 0xffff |

| | | |
|---|---|---|
| CORACQ_CAP_EXT_TRIGGER_DURATION_MIN | 0 | 0x0 |
| CORACQ_CAP_EXT_TRIGGER_FRAME_COUNT | 1 | (0x1) True, more than 1 frame can be acquired |
| CORACQ_CAP_EXT_TRIGGER_LEVEL | 00000011b | (0x01), A TTL signal level<br>(0x02), A RS-422 signal level |
| CORACQ_CAP_EXT_TRIGGER_SOURCE | 0 | (0x0) |
| CORACQ_CAP_EXT_TRIGGER_DELAY_MIN | 0 | 0x0 Medium Mono |
| CORACQ_CAP_EXT_TRIGGER_DELAY_MAX | 255 | 0xff Medium Mono |
| CORACQ_CAP_EXT_TRIGGER_DELAY_TIME_BASE | 13 | 0xd Medium Mono |
| | | |
| CORACQ_CAP_FRAME_LENGTH | 00000011b | (0x1) Fixed length images<br>(0x2) Variable length images |
| CORACQ_CAP_HSYNC_REF | 2 | (0x2) End of horizontal sync |
| CORACQ_CAP_INT_FRAME_TRIGGER | 1 | (0x1) True, internal frame trigger is available |
| CORACQ_CAP_INT_FRAME_TRIGGER_FREQ_MAX | 1073741823 | 0x3fffffff (in milli-Hz) |
| CORACQ_CAP_INT_FRAME_TRIGGER_FREQ_MIN | 1 | 0x1 (in milli-Hz) |
| CORACQ_CAP_INT_LINE_TRIGGER | 1 | (0x1) True, internal line trigger is available |
| | | |
| CORACQ_CAP_LINE_INTEGRATE_DURATION_MAX | 16777215 | 0xffffff (in pixels) |
| CORACQ_CAP_LINE_INTEGRATE_DURATION_MIN | 1 | 0x1 (in pixels) |
| CORACQ_CAP_LUT | 1 | (0x1) True, at least one LUT is available |
| CORACQ_CAP_LUT_ENABLE | 1 | (0x1) True, input LUT can be enabled/disabled |
| | | |
| CORACQ_CAP_OUTPUT_FORMAT<br>*(Medium Mono, Base Mono)* | | CORACQ_VAL_OUTPUT_FORMAT_MONO8 |
| CORACQ_CAP_OUTPUT_FORMAT<br>*(Medium Mono, Base Mono)* | | CORACQ_VAL_OUTPUT_FORMAT_MONO16 |
| CORACQ_CAP_OUTPUT_FORMAT<br>*(Medium Color RGB)* | | CORACQ_VAL_OUTPUT_FORMAT_RGB101010 |
| CORACQ_CAP_OUTPUT_FORMAT<br>*(Medium RGB, Base RGB)* | | CORACQ_VAL_OUTPUT_FORMAT_RGB8888 |
| CORACQ_CAP_OUTPUT_FORMAT_BYTE_MULT | 4 | 0x4 (in bytes) |
| | | |
| CORACQ_CAP_SHAFT_ENCODER | 1 | (0x1) True, shaft encoder option is available |
| CORACQ_CAP_SHAFT_ENCODER_DROP | 1 | (0x1) True, edge dropping is available. |
| CORACQ_CAP_SHAFT_ENCODER_DROP_MAX | 511 | (0x1ff ) |
| CORACQ_CAP_SHAFT_ENCODER_DROP_MIN | 0 | (0x0) |
| CORACQ_CAP_SHAFT_ENCODER_LEVEL | 00000010b | (0x2) a differential signal |
| | | |
| CORACQ_CAP_STROBE | 1 | (0x1) Supports at least one output strobe pulse method |
| CORACQ_CAP_STROBE_DELAY_2_MAX | 65535000 | 0x3e7fc18(in µs) |
| CORACQ_CAP_STROBE_DELAY_2_MIN | 0 | 0x0 (in µs) |
| CORACQ_CAP_STROBE_DELAY_MAX | 65535000 | 0x3e7fc18(in µs) |
| CORACQ_CAP_STROBE_DELAY_MIN | 0 | 0x0 (in µs) |
| CORACQ_CAP_STROBE_DURATION_MAX | 65535000 | 0x3e7fc18(in µs) |
| CORACQ_CAP_STROBE_DURATION_MIN | 0 | 0x0 (in µs) |
| CORACQ_CAP_STROBE_LEVEL | 1 | (0x1) A TTL signal. |

| | | |
|---|---|---|
| CORACQ_CAP_STROBE_METHOD | 00000111b | (0x01) Strobe Method #1<br>(0x02) Strobe Method #2<br>(0x04) Strobe Method #3 |
| CORACQ_CAP_STROBE_POLARITY | 00000011b | (0x1) Strobe pulse will be active low<br>(0x2) Strobe pulse will be active high |
| CORACQ_CAP_SYNC_CROP_HEIGHT_MAX | 16777215 | 0xffffff (in lines) |
| CORACQ_CAP_SYNC_CROP_HEIGHT_MIN | 0 | 0x0 (in lines) |
| CORACQ_CAP_SYNC_CROP_HEIGHT_MULT | 1 | 0x1 (line) |
| CORACQ_CAP_SYNC_CROP_LEFT_MAX | 16777215 | 0xffffff (in pixels) |
| CORACQ_CAP_SYNC_CROP_LEFT_MIN | 0 | 0x0 (in pixels) |
| CORACQ_CAP_SYNC_CROP_LEFT_MULT | 1 | 0x1 (pixel) |
| CORACQ_CAP_SYNC_CROP_TOP_MAX | 16777215 | 0xffffff (in lines) |
| CORACQ_CAP_SYNC_CROP_TOP_MIN | 0 | 0x0 (in lines) |
| CORACQ_CAP_SYNC_CROP_TOP_MULT | 1 | 0x1 (line) |
| CORACQ_CAP_SYNC_CROP_WIDTH_MAX | 16777215 | 0xffffff (in pixels) |
| CORACQ_CAP_SYNC_CROP_WIDTH_MIN | 0 | 0x0 (in pixels) |
| CORACQ_CAP_SYNC_CROP_WIDTH_MULT | 1 | 0x1 (pixel) |
| CORACQ_CAP_TIME_INTEGRATE_DELAY_MAX | 65535000 | 0x3e7fc18 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_DELAY_MIN | 0 | 0x0 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_DURATION_MAX | 65535000 | 0x3e7fc18 (in µs) |
| CORACQ_CAP_TIME_INTEGRATE_DURATION_MIN | 1 | 0x1 (in µs) |
| CORACQ_CAP_VSYNC_REF | 2 | (0x2) End of vertical sync |
| CORACQ_CAP_SHAFT_ENCODER_MULTIPLY | 1 | 0x1 |
| CORACQ_CAP_SHAFT_ENCODER_MULTIPLY_MIN | 1 | 0x1 |
| CORACQ_CAP_SHAFT_ENCODER_MULTIPLY_MAX | 32 | 0x20 |
| CORACQ_CAP_SHAFT_ENCODER_MULTIPLY_STEP | 536870914 | 0x20000002 |
| CORACQ_CAP_BAYER_DECODER_METHOD | 1 | True; with Bayer configuration |
| CORACQ_CAP_BAYER_DECODER_METHOD | 1 | method 1 supported (with Bayer configuration) |
| CORACQ_CAP_BAYER_DECODER_WB_GAIN_MIN | 100000 | 0x186a0 (with Bayer configuration) |
| CORACQ_CAP_BAYER_DECODER_WB_GAIN_MAX | 499609 | 0x79f99 (with Bayer configuration) |
| CORACQ_CAP_BAYER_DECODER_WB_OFFSET_MIN | 0 | 0x0 (with Bayer configuration) |
| CORACQ_CAP_BAYER_DECODER_WB_OFFSET_MAX | 0 | 0x0 (with Bayer configuration) |
| CORACQ_CAP_SERIAL_PORT_INDEX | 0 | 0x0 |

# Acquisition Related Capabilities

| Capability | Value or Limits or Bitfield | Capability Description / Details |
|---|---|---|
| CORACQ_CAP_DETECT_HACTIVE | 1 | True |
| CORACQ_CAP_DETECT_VACTIVE | 1 | True |
| CORACQ_CAP_EVENT_TYPE | | 0xc100e000<br>bit field defines supported modes |
| CORACQ_CAP_SOFTWARE_TRIGGER | 1 | True |
| CORACQ_CAP_SIGNAL_STATUS | 00000111b | (0x01) CORACQ_VAL_SIGNAL_HSYNC_PRESENT<br>(0x02) CORACQ_VAL_SIGNAL_VSYNC_PRESENT<br>(0x04) CORACQ_VAL_SIGNAL_PIXEL_CLK_PRESENT |
| CORACQ_CAP_FLAT_FIELD | 1 | True (with Flat Field Correction configuration) |
| CORACQ_CAP_FLAT_FIELD_OFFSET_MIN | 0 | 0x0 |
| CORACQ_CAP_FLAT_FIELD_OFFSET_MAX | 255 | 0xff |
| CORACQ_CAP_FLAT_FIELD_GAIN_MAX | 1 | 0x1 |
| CORACQ_CAP_FLAT_FIELD_GAIN_MAX | 255 | 0xff |
| CORACQ_CAP_FLAT_FIELD_GAIN_DIVISOR | 128 | 0x80 |
| CORACQ_CAP_FLAT_FIELD_PIXEL_REPLACEMENT | 1 | True; The flat field resource does support pixel replacement when the gain is zero. |
| CORACQ_CAP_SERIAL_PORT | 1 | 0x1 |

# X64-CL Express Memory Error with Area Scan Frame Buffer Allocation

The memory error message [ **Error: "CorXferConnect" <Xfer module> - No memory ()** ] may occur when loading a Sapera camera file, or when the application configures a frame buffer for area scan cameras. The problem is that the X64-CL Express does not have enough onboard memory for two frame buffers.

The X64-CL Express when used with area scan cameras, allocates two internal frame buffers in onboard memory, each equal in size to the acquisition frame buffer. This allocation is automatic at the driver level. The X64-CL Express driver allocates two buffers to ensure that the acquired video frame is complete and not corrupted in cases where the transfer to host system memory may be interrupted by other host system processes.

The total size of the two internal frame buffers must be somewhat smaller than the total onboard memory due to memory overhead required for image transfer management. Also note that the X64-CL Express dual configuration equally divides the onboard memory between the two acquisition modules, reducing the available memory for the two buffers by half.

# X64-CL Express Sapera Servers & Resources

## Servers and Resources

| Servers | | Resources | | | |
|---|---|---|---|---|---|
| **Name** | **Description** | **Type** | **Name** | **Index** | **Description** |
| X64-CL_Express_1 | X64-CL Express Medium | Acquisition | CamLink Medium Mono | 0 | CamLink Medium configuration, monochrome output, Camera #1 |
| | | | CamLink Medium RGB | 1 | CamLink Medium configuration, RGB output, Camera #1 |
| X64-CL_Express_1 | X64-CL Express Dual Base | Acquisition | CamLink Base Mono #1 | 0 | CamLink Base configuration, monochrome output, Camera #1 |
| | | | CamLink Base Mono #2 | 1 | CamLink Base configuration, monochrome output, Camera #2 |
| | | | CamLink Base Color RGB #1 | 2 | CamLink Base configuration, color output, Camera #1 |
| | | | CamLink Base Color RGB #2 | 3 | CamLink Base configuration, color output, Camera #2 |
| X64-CL_Express_1 | X64-CL Express Medium | Acquisition | CamLink Bayer Decoder #1 | 0 | CamLink Medium configuration, Camera #1, Bayer Decoder |

# Transfer Resource Locations

The following table illustrates all possible source/destination pairs in a transfer.

| Source | Transfer passing through | Destination |
|---|---|---|
| X64-CL Express Acquisition | 1 to $2^{17}$ internal buffers | 1 to $2^{17}$ Host Buffers |
| X64-CL Express Acquisition | 1 to $2^{17}$ internal buffers & the X64 internal processor | 1 to $2^{17}$ Host Buffers |

# Technical Specifications

## X64-CL Express Board Specifications

### X64-CL Express Dimensions

Approximately 6.5 in. (16.6 cm) wide by 4 in. (10 cm) high.

### Digital Video Input & Controls

| | |
|---|---|
| Input Type | Camera Link Specifications Rev 1.10 compliant 1 Medium or 2 Base |
| Common Pixel Formats | |
| Two Base Configuration Pixel Formats/Tap | 3 x 8-bit, 2 x 10-bit, 2 x 12-bit, 1 x 14-bit, 1 x 16-bit, 1 x 24-bit RGB |
| One Medium Configuration Pixel Formats/Tap | 4 x 8-bit, 4 x 10-bit, 4 x 12-bit, 1 x 14-bit, 1 x 16-bit, 1 x 30-bit RGB, 1 x 36-bit RGB |
| Scanning | Progressive, Multi-Tap, Multi-Channel, Four quadrant, Tap reversal |
| Input LUTs | Yes.<br>See section "X64-CL Express LUT availability" (page 59) for details. |
| Resolution<br><br>*note: these are X64-CL Express maximums, not Camera Link specifications* | Horizontal Minimum:<br>8 Pixels per tap<br>Horizontal Maximum:<br>8-bits/pixel x 256K Pixels/line<br>16-bits/pixel x 128K Pixels/line<br>32-bits/pixel x 64K Pixels/line<br>64-bits/pixel x 32K Pixels/line<br>Vertical Minimum:<br>1 line<br>Vertical Maximum:<br>up to 16,000,000 lines (for area scan sensors) |
| Pixel Clock Range | up to 85 MHz |
| Bandwidth to Host System | Approximately 160MB/s. Limited by PCI Express 1x plus additional X64-CL Express DMA management. |
| Serial Port | Supports communication speeds from 9600 to 115 kbps |
| Controls | Comprehensive event notification includes end/start-of-field/frame/transfer<br>Dual independent TTL/LVDS trigger input programmable as active high or low (edge or level trigger, where pulse width minimum is 100ns)<br>Dual independent TTL Strobe outputs<br>Quadrature (AB) shaft-encoder inputs for external web synchronization (maximum frequency for any shaft encoder input is 1 MHz) |

| | |
|---|---|
| Processing<br><br>*Dependant on user loaded firmware configuration* | **Bayer Mosaic Filter**:<br>Hardware Bayer Engine supports one 8, 10 or 12-bit Bayer camera input.<br>Bayer output format supports 8 or 10-bit RGB/pixel.<br>Zero host CPU utilization for Bayer conversion.<br><br>**Flat Field Correction** (Shading Correction):<br>Real-time Flat-line and Flat-field correction.<br>Compensates for sensor defects such as FPN, PRNU, defective pixels and variations between pixels due to the light refraction through a lens (Shading effect).<br><br>Supports two independent monochrome cameras concurrently.<br><br>PRNU (*Photo Response Non Uniformity*): PRNU is the variation in response between sensor pixels.<br><br>FPN (*Fixed Pattern Noise*): FPN is the unwanted static variations in response for all pixels in the image. |

# Host System Requirements

## General System Requirements for the X64-CL Express

- PCI Express 1x slot compatible

## Operating System Support

Windows 2000 SP1 and Windows XP

## Power Requirements

| | | |
|---|---|---|
| +5 Volt | 2 amp typical | *Note: other internal voltages are derived from +5V* |
| +12 Volt | | *As per camera connected and supplied by X64-CL Express PC power interface* |

## Environment

| | |
|---|---|
| Ambient Temperature: | 10° to 50° C (operation)<br>0° to 70° C (storage) |
| Relative Humidity: | 5% to 90% non-condensing (operating)<br>0% to 95% (storage) |

# EMI Certifications

Class B, both FCC and CE

CORECO

EXCELLENCE IN MACHINE VISION

**EC & FCC DECLARATION OF CONFORMITY**

We : CORECO INC.
7075 Place Robert-Joncas, Suite 142,
St. Laurent, Quebec, Canada H4M 2Z2

Declare under sole legal responsibility that the following products conform to the protection requirements of council directive 89/336 EEC on the approximation of the laws of member states relating to electromagnetic compatibility, as amended by directive 93/68/EEC :

FRAME GRABBER BOARD: X64-CL

The products to which this declaration relates are in conformity with the following relevant harmonised standards, the reference numbers of which have been published in the Official Journal of the European Communities :

EN55022 :1998- Residential, Commercial and Light Industry
EN50204: 1995
EN61000-4: 1995, 1996

Further declare under our sole legal responsibility that the product listed conforms to the code of federal regulations CFR 47 part 15 for a class B product.

St. Laurent, Canada — Location
May 13, 2003 — Date

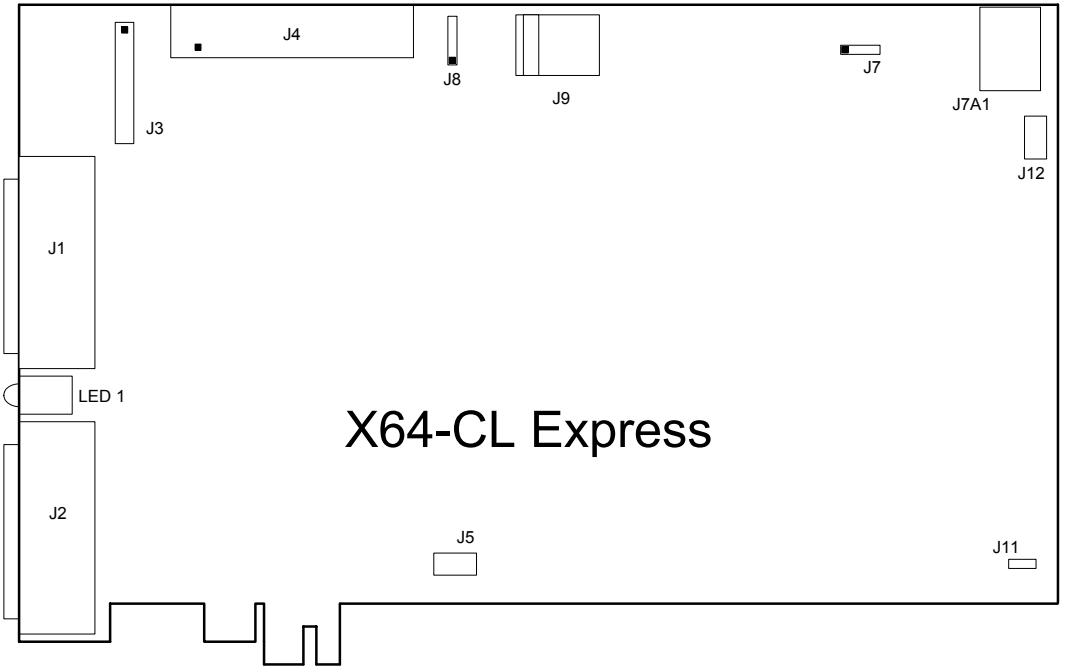
Ghislain Beaupré
Vice-President,
Research & Development

*State-of-the-art imaging products*

Coreco Imaging 7075 Place Robert-Joncas, Suite 142, Saint-Laurent, Quebec, Canada H4M 2Z2
Telephone: (514) 333-1301 Fax: (514) 333-1388 www.imaging.com

# Connector and Switch Locations

## X64-CL Express Board Layout Drawings

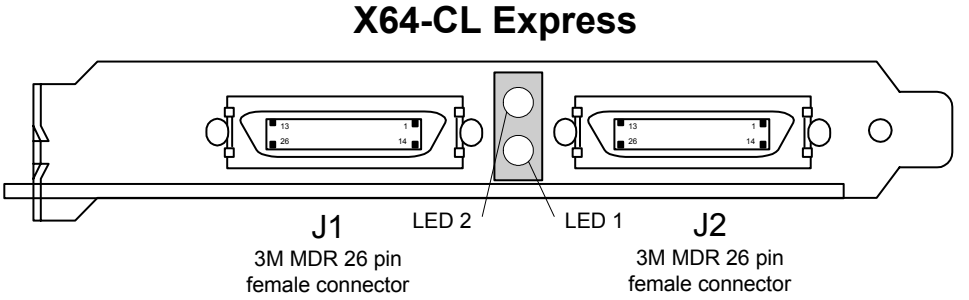



## Connector Description List

| Connector | Description | Connector | Description |
|---|---|---|---|
| J1 | Camera Link Connector | J8 | Camera Power Selector |
| J2 | Camera Link Connector | J9 | PC power to camera interface. |
| J3 | Reserved | J11 | Normal (jumper on)<br>Safe Start Mode (jumper off) |
| J4 | External Signals connector | J12 | Reserved |
| J5, J7, J7A1 | Reserved | | |

# Connector and Switch Specifications

## X64-CL Express End Bracket



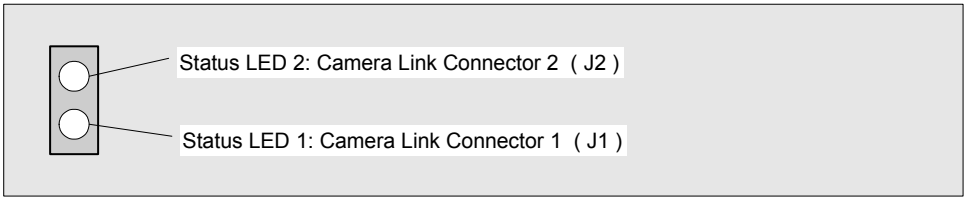

The hardware installation process is completed with the connection of a supported camera to the X64-CL Express board using Camera Link cables (see “Camera Link Cables” on page **93**).

- The X64-CL Express board supports a camera with one or two Camera Link MDR-26 connectors (two Base or one Medium – see “Data Port Summary” on page 92 for information on Camera Link configurations).
- Connect the camera to the J1 connector with a Camera Link cable. When using a Medium camera, connect the second camera connector to J2.

Caution: If the camera is powered by the X64-CL, it is very important that the correct power supply voltage is selected correctly. Refer to "J8: Power to Camera Voltage Selector" on page 87 for information on the selection jumper.

Contact DALSA Coreco or browse our web site http://www.imaging.com/camsearch for the latest information on X64-CL Express supported cameras.

# Status LEDs Functional Description




## Status LED Modes

- Red: No camera connected or camera has no power.
- Green: Camera connected and is ON. Camera clock detected. No line valid detected.
- Slow Flashing Green: Camera Line Valid signal detected.
- Fast Flashing Green: Acquisition in progress.
- Status LED 2 flashing red, X64-CL Express board with Medium Configuration only: Camera pixel clock incorrectly connected to J2 instead of J1. (Example - a Base camera is incorrectly connected to J2).

## J1: Camera Link Connector 1

| Name | Pin # | Type | Description |
|---|---|---|---|
| BASE_X0- | 25 | Input | Neg. Base Data 0 |
| BASE_X0+ | 12 | Input | Pos. Base Data 0 |
| BASE_X1- | 24 | Input | Neg. Base Data 1 |
| BASE_X1+ | 11 | Input | Pos. Base Data 1 |
| BASE_X2- | 23 | Input | Neg. Base Data 2 |
| BASE_X2+ | 10 | Input | Pos. Base Data 2 |
| BASE_X3- | 21 | Input | Neg. Base Data 3 |
| BASE_X3+ | 8 | Input | Pos. Base Data 3 |
| BASE_XCLK- | 22 | Input | Neg. Base Clock |
| BASE_XCLK+ | 9 | Input | Pos. Base Clock |
| SERTC+ | 20 | Output | Pos. Serial Data to Camera |
| SERTC- | 7 | Output | Neg. Serial Data to Camera |
| SERTFG- | 19 | Input | Neg. Serial Data to Frame Grabber |
| SERTFG+ | 6 | Input | Pos. Serial Data to Frame Grabber |
| CC1- | 18 | Output | Neg. Camera Control 1 |
| CC1+ | 5 | Output | Pos. Camera Control 1 |
| CC2+ | 17 | Output | Pos. Camera Control 2 |
| CC2- | 4 | Output | Neg. Camera Control 2 |
| CC3- | 16 | Output | Neg. Camera Control 3 |
| CC3+ | 3 | Output | Pos. Camera Control 3 |
| CC4+ | 15 | Output | Pos. Camera Control 4 |
| CC4- | 2 | Output | Neg. Camera Control 4 |
| GND | 1, 13, 14, 26 | | Ground |

## J2: Camera Link Connector 2 (on X64-CL Express with Dual Base Configuration)

The Camera Link connector J2 on the X64-CL Dual board is identical to Camera Link connector 1 (J1).

## J2: Camera Link Connector 2 (on X64-CL Express in Medium Configuration)

| Name | Pin # | Type | Description |
|---|---|---|---|
| MEDIUM _X0- | 25 | Input | Neg. Medium Data 0 |
| MEDIUM _X0+ | 12 | Input | Pos. Medium Data 0 |
| MEDIUM _X1- | 24 | Input | Neg. Medium Data 1 |
| MEDIUM _X1+ | 11 | Input | Pos. Medium Data 1 |
| MEDIUM _X2- | 23 | Input | Neg. Medium Data 2 |
| MEDIUM _X2+ | 10 | Input | Pos. Medium Data 2 |
| MEDIUM _X3- | 21 | Input | Neg. Medium Data 3 |
| MEDIUM _X3+ | 8 | Input | Pos. Medium Data 3 |
| MEDIUM _XCLK- | 22 | Input | Neg. Medium Clock |
| MEDIUM _XCLK+ | 9 | Input | Pos. Medium Clock |
| TERM | 20 | | Term Resistor |
| TERM | 7 | | Term Resistor |
| | 19 | Reserved | |
| | 6 | Reserved | |
| | 18 | Reserved | |
| | 5 | Reserved | |
| | 17 | Reserved | |
| | 4 | Reserved | |
| | 15 | Reserved | |
| | 2 | Reserved | |
| | 16 | Reserved | |
| | 3 | Reserved | |
| GND | 1, 13, 14, 26 | | Ground |

# Camera Link Camera Control Signal Overview

Four LVDS pairs are for general-purpose camera control, defined as camera inputs / frame grabber outputs by the Camera Link Base camera specification. These controls are on J1 and also on J2 for the second Base camera input of the X64-CL Express in two Base configuration.

- Camera Control 1 (CC1)
- Camera Control 2 (CC2)
- Camera Control 3 (CC3)
- Camera Control 4 (CC4)

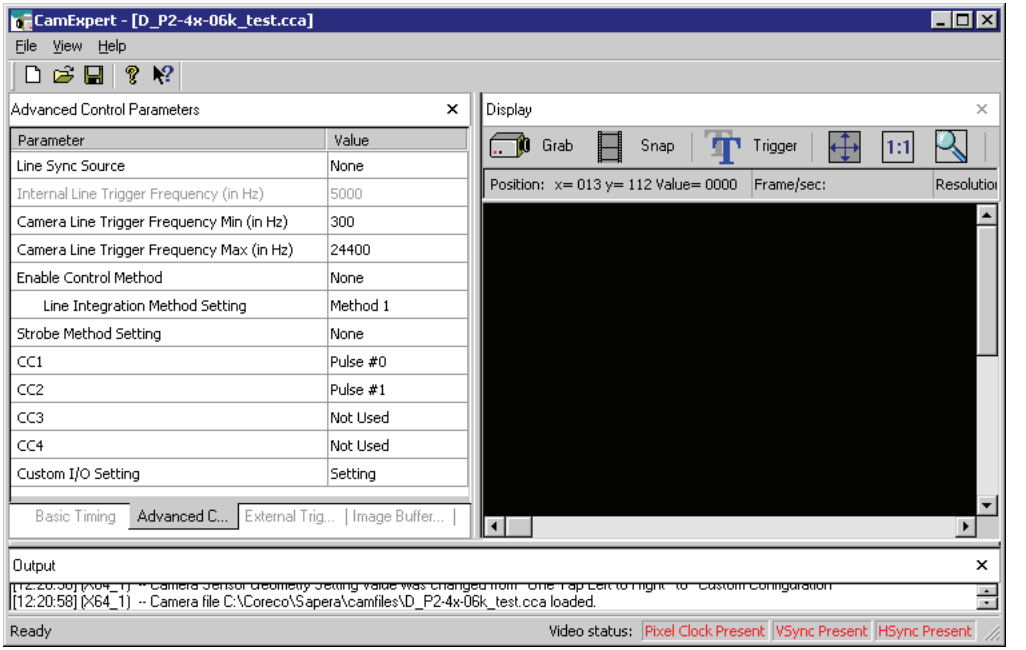
Each camera manufacture is free to define the signals input on any one or all four control signals. These control signals are used either as camera control pulses or as a static logic state. Control signals not required by the camera are simply assigned as not used. Refer to your camera's user manual for information on what control signals are required.

---

Note: The X64-CL Express pulse controller has a minimum resolution of 1 us. When configuring the Camera Link control signals, such as exposure control, etc. use values in increments of 1 us.

---

The X64-CL Express can assign any camera control signal to the appropriate Camera Link control. The following screen shot shows the Sapera CamExpert dialog where Camera Link controls are assigned.



# J4: External Signals Connector

## J4 Pin Header Numbering Detail

| | | | | |
|---|---|---|---|---|
| 2 | 4 | ... | 24 | 26 |
| 1 | 3 | ... | 23 | 25 |

## J4 Signal Descriptions

| Pin # | Signal | X64-CL Express Medium Description | X64-CL Express Dual Base Description |
|---|---|---|---|
| 1 | | Reserved | Reserved |
| 2, 4, 6 | GND | | |
| 3 | | Reserved | Reserved |
| 5 | | Reserved | Reserved |
| 7 | | Reserved | Reserved |
| 8 | | Reserved | Reserved |
| 9 | | Reserved | Reserved |
| 10 | | Reserved | Reserved |
| 11 | TrigIn 1 + (*input*) | LVDS Trigger In + or TTL Trigger In (*see note 1*) | *CamLink Base #1* LVDS + or TTL Trigger In (pulse width minimum 100ns) *see note 2* |
| 12 | TrigIn 1 - (*input*) | LVDS Trigger In - (or TTL Trigger In GND) | *CamLink Base #1* LVDS Trigger - (or TTL Trigger GND) |
| 13 | TrigIn 2 + (*input*) | LVDS Trigger In + or TTL Trigger In (*Used for two pulse external trigger with variable frame length linescan acquisition*) | *CamLink Base #2* LVDS + or TTL Trigger In (pulse width minimum 100ns) *see note 2* |
| 14 | TrigIn 2 - (*input*) | LVDS Trigger In - (or TTL Trigger In GND) | *CamLink Base #2* LVDS Trigger - (or TTL Trigger GND) |
| 15 | Phase A + (*input*) | LVDS/RS422 Shaft Encoder phase A + or line trigger used with linescan cameras (*see note 3 & 4 & 9*) | LVDS/RS422 Shaft Encoder + or line trigger used with linescan cameras (*see note 5 & 9*) |
| 16 | Phase A - (*input*) | LVDS/RS422 Shaft Encoder phase A - (*see note 9*) | LVDS/RS422 Shaft Encoder - (*see note 9*) |

| | | | |
|---|---|---|---|
| 17 | Phase B +<br>(*input*) | LVDS/RS422 Shaft Encoder phase B +<br>or line trigger<br>used with linescan cameras<br>(*see note 3 & 4 & 9*) | LVDS/RS422 Shaft Encoder +<br>or line trigger<br>used with linescan cameras<br>(*see note 5 & 9*) |
| 18 | Phase B -<br>(*input*) | LVDS/RS422 Shaft Encoder phase B -<br>(*see note 9*) | LVDS/RS422 Shaft Encoder -<br>(*see note 9*) |
| 19 | Strobe 2<br>(*output*) | *not used* | *CamLink Base #2*<br>TTL Strobe Output<br>(*see note 7*) |
| 20 | | Reserved | |
| 21 | Strobe 1<br>(*output*) | TTL Strobe Output<br>(*see note 6*) | *CamLink Base #1*<br>TTL Strobe Output<br>(*see note 7*) |
| 22, 24, 26 | GND | | |
| 23, 25 | DC Power<br>(*see note 8*) | Voltage selected (+12 or +5) via J8 (see "J8: Power to Camera Voltage Selector" on page 87) | Voltage selected (+12 or +5) via J8 (see "J8: Power to Camera Voltage Selector" on page 87) |

## Notes:

1. **X64-CL Express :**
   Refer to Sapera parameters CORACQ_PRM_EXT_TRIGGER_LEVEL
   CORACQ_PRM_EXT_FRAME_TRIGGER_LEVEL
   CORACQ_PRM_EXT_TRIGGER_ENABLE
   CORACQ_PRM_EXT_TRIGGER_DETECTION
   See also *.cvi file entries:
   External Trigger Level, External Frame Trigger Level, External Trigger Enable, External Trigger Detection.

2. **X64-CL Express Dual Base:**
   Sapera parameter CORACQ_PRM_EXT_TRIGGER_LEVEL,
   CORACQ_PRM_EXT_FRAME_TRIGGER_LEVEL is a common control to both CamLink Base #1 and #2.
   Parameters CORACQ_PRM_EXT_TRIGGER_ENABLE and CORACQ_PRM_EXT_TRIGGER_DETECTION are independent for each Camera Link input.
   When detection is CORACQ_VAL_DOUBLE _PULSE_RISING_EDGE or CORACQ_VAL_DOUBLE _PULSE_FALLING_EDGE the start trigger is "Trig In 1" and the end trigger is "Trig in 2".
   See also *.cvi file entries:
   External Trigger Level, External Frame Trigger Level, External Trigger Enable, External Trigger Detection.

3. **X64-CL Express:**
   See "Line Trigger Source Selection for Linescan Applications" on page 44 for more information.
   Refer to Sapera parameters CORACQ_PRM_SHAFT_ENCODER_ENABLE

CORACQ_PRM_SHAFT_ENCODER_DROP
or refer to CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE
CORACQ_PRM_EXT_LINE_TRIGGER_DETECTION
CORACQ_PRM_EXT_LINE_TRIGGER_LEVEL (fixed at LVDS)
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE
See also *.cvi file entries:
Shaft Encoder Enable, Shaft Encoder Pulse Drop
or see External Line Trigger Enable, External Line Trigger Detection, External Line Trigger Level, External Line Trigger Source.

4. **X64-CL Express:**
Important: When using only one shaft encoder input phase, say phase A, then the phase B inputs must be terminated by connecting phase B- to board ground available on any pin labeled GND and phase B+ to any DC source with a minimum of 100 mV positive relative to the phase B- input.

5. **X64-CL Express Dual Base:**
Same parameters as X64-CL Full (see note 3).
Parameters are independent for CamLink Base #1 and #2.

6. **X64-CL Express:**
Refer to Sapera parameters CORACQ_PRM_STROBE_ENABLE
CORACQ_PRM_STROBE_POLARITY, CORACQ_PRM_STROBE_LEVEL,
CORACQ_PRM_STROBE_METHOD, CORACQ_PRM_STROBE_DELAY
CORACQ_PRM_STROBE_DURATION
See also *.cvi file entries:
Strobe Enable, Strobe Polarity, Strobe Level, Strobe Method, Strobe Delay, Strobe Duration.

7. **X64-CL Express Dual Base:**
Same parameters as X64-CL Full (see note 6).
Parameters are independent for CamLink Base #1 and #2.

**8. DC Power Constraints:**
The supplied host PC voltage is selected (+12 or +5) via the shorting jumper J8 (see "J8: Power to Camera Voltage Selector" on page 87 for details).
A 1.5A resettable fuse is included on the board. If the fuse is tripped, power off the host computer to reset the fuse.

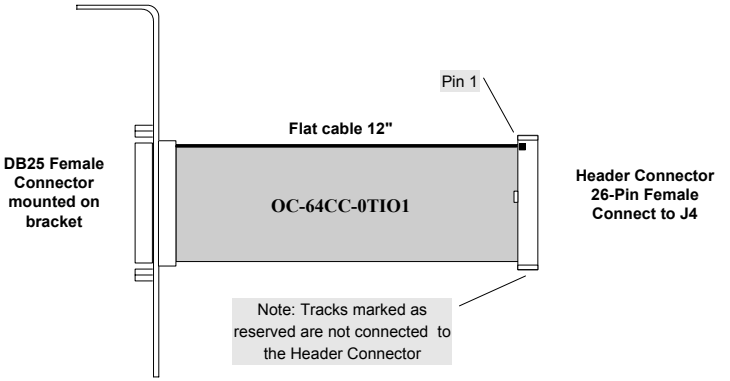
**9.** See "Connecting a TTL Shaft Encoder Signal to the LVDS/RS422 Input" on page 85 for details on using a TTL shaft encoder signal.

# X64-CL Express: External Signals Connector Bracket Assembly

The External Signals bracket (OC-64CC-0TIO1) provides a simple way to bring out the signals from the X64-CL Express External Signals ConnectorJ4 to a bracket mounted DB25. Install the bracket assembly into an adjacent PC expansion slot and connect the free cable end to the X64-CL Express J4 header. When connecting to J4, make sure that the cable pin 1 goes to J4 pin 1 (see layout drawings for your board revision: "X64-CL Express Board Layout Drawings" on page 75).

**Note:** When using the optional X-I/O module, this external signals cable is not used. Instead a 26 wire cable from the X-I/O J21 connects to X-I/O J4. All external signals described here plus the additional 8 input – 8 output general I/O controls are now on the X-I/O DB37 connector. See "Appendix: X-I/O Module Option" on page 95 for installation and pinout information.

## External Signals Connector Bracket Assembly Drawing




## External Signals Connector Bracket Assembly Signal Description

Refer to the table "J4: External Signals Connector " on page 81 for important signal descriptions.

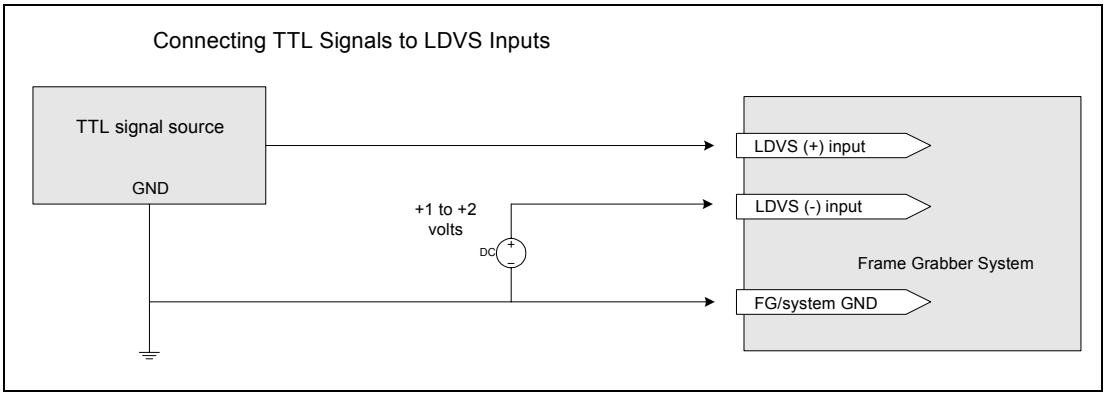
| DB25 Pin Number | Signal Names | Connector (to J4) |
|---|---|---|
| 1, 2, 3, 4, 17, 5, 18, 23 | Reserved | 1, 3, 5, 7, 8, 9, 10, 20 |
| 6 | TrigIn 1+ | 11 |
| 19 | TrigIn 1- | 12 |
| 7 | TrigIn 2+ | 13 |
| 20 | TrigIn 2- | 14 |
| 8 | Shaft Encoder phase A + | 15 |
| 21 | Shaft Encoder phase A - | 16 |
| 9 | Shaft Encoder phase B + | 17 |
| 22 | Shaft Encoder phase B - | 18 |

| | | |
|---|---|---|
| 11 | Strobe 1 | 21 |
| 10 | Strobe 2 | 19 |
| 12, 13 | DC power (+12 or +5 via J8) | 23, 25 |
| 14, 15, 16, 24, 25, | Ground | 2, 4, 6, 22, 24 |

# Connecting a TTL Shaft Encoder Signal to the LVDS/RS422 Input

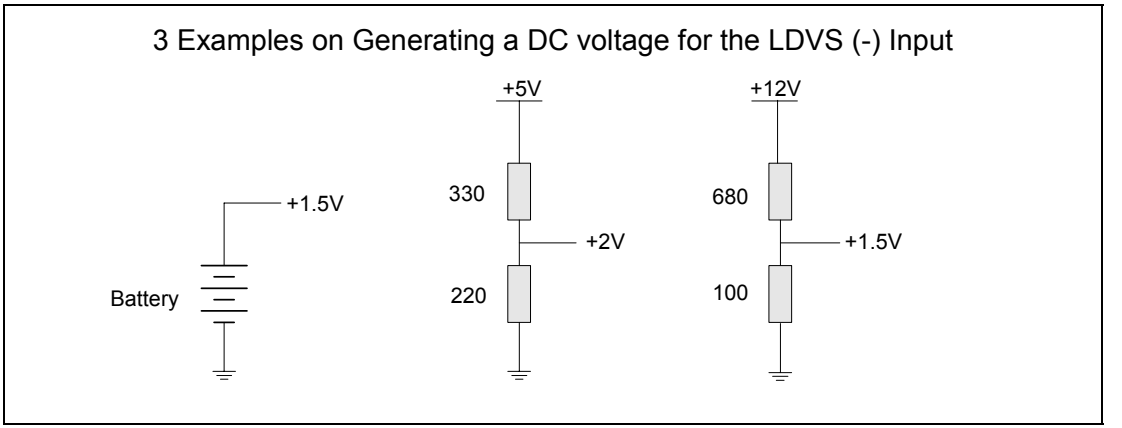
A TTL shaft encoder signal can be directly connected to the X64-CL Express LVDS/RS422 (+) input but the low side (-) input of the pair must be biased with a DC voltage to ensure reliable operation. This section shows the connection diagram along with suggestions as to how to generate the DC bias voltage. The actual physical wiring is left as an additional detail to interfacing a shaft encoder to the X64-CL Express to the imaging system.

## TTL Shaft Encoder to LVDS/RS422 Input Block Diagram




- LVDS/RS422 (-) input is biased to a DC voltage from +1 to +2 volts.
- This guarantees that the TTL signal connected to the LVDS/RS422 (+) input will be detected as a logic high or low relative to the (-) input.
- The TTL shaft encoder ground, the bias voltage ground, and the X64-CL Express computer system ground must be connected together.
- The maximum frequency for any shaft encoder input is 1 MHz.

### LVDS/RS422 (-) Input Bias Source Generation




- DC voltage for the LVDS/RS422 (-) input can be generated by a resister voltage divider.
- Use a single battery cell if this is more suitable to your system.
- A DC voltage (either +5 or +12) is available on External Signals ConnectorJ4. See "J8: Power to Camera Voltage Selector" on page 87 for information.

## External Trigger TTL Input Electrical Specification

The incoming trigger pulse is “debounced” to ensure that no voltage glitch would be detected as a valid trigger pulse. This debounce circuit time constant can be programmed from 0µs to 255µs. Any pulse smaller than the programmed value is blocked and therefore not seen by the acquisition circuitry.

| Electrical parameters | Description | Value |
|---|---|---|
| TrigIn low | Low logic level input | <= 0.8 V |
| TrigIn high | High logic level input | >= 2.0 V |
| TrigIn pulse width | Minimum trigger pulse width | 100 ns |

**Sapera parameters for External Trigger:**

CORACQ_PRM_EXT_TRIGGER_ENABLE = CORACQ_VAL_EXT_TRIGGER_ON

CORACQ_PRM_EXT_TRIGGER_SOURCE

CORACQ_PRM_EXT_TRIGGER_DETECTION = {CORACQ_VAL_RISING_EDGE, CORACQ_VAL_FALLING_EDGE, CORACQ_VAL_ACTIVE_LOW. CORACQ_VAL_ACTIVE_HIGH}

CORACQ_PRM_EXT_TRIGGER_DURATION: Debounce duration

## Strobe TTL Output Electrical Specification

| Electrical parameters | Description | Value |
|---|---|---|
| $V_{OH\ typ}$ | Typical high-level output voltage | 3.9V |
| $I_{OH\ max}$ | Maximum high-level output current | -8mA (sourcing) |
| $I_{OL\ max}$ | Maximum low-level output current | 8mA (sinking) |

**Sapera parameters for Strobe :**

Refer to Strobe Method in Sapera documentation

CORACQ_PRM_STROBE_ENABLE = TRUE

CORACQ_PRM_STROBE_METHOD={CORACQ_VAL_STROBE_METHOD_1, CORACQ_VAL_STROBE_METHOD_2}

CORACQ_PRM_STROBE_POLARITY={CORACQ_VAL_ACTIVE_LOW, CORACQ_VAL_ACTIVE_HIGH}

CORACQ_PRM_STROBE_DELAY: Pulse offset from trigger event

CORACQ_PRM_STROBE_DELAY_2: Duration of exclusion region

CORACQ_PRM_STROBE_DURATION: Pulse duration

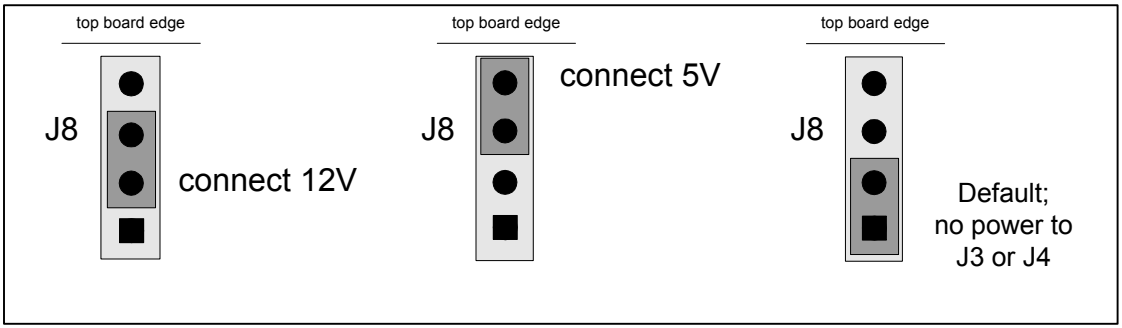
## J8: Power to Camera Voltage Selector

When the PC floppy drive power supply cable is connected to J9, a shorting jumper on J8 selects either 5 Vdc or 12 Vdc for the camera power supply. This supply voltage is available on the External Signals Connectorblock.

### Important:

For the X64-CL Express a 1.5A power on reset fuse is included on the board. If the fuse is tripped, power off (not just a soft reboot) the computer system to reset the fuse.

### J8 on X64-CL Express



## J9: PC Power to Camera Interface

Connect the PC floppy drive power connector to J9 so as to supply DC power to the camera. Place the J8 shorting jumper so as to select 5 Vdc or 12 Vdc for the camera.

## J11: Start Mode

- Default Mode: Shunt jumper is installed.
- Safe Mode: Shunt jumper is removed if any problems occurred while updating the X64-CL Express firmware. With the jumper off, reboot the computer and update the firmware again. When the update is complete, install the jumper and reboot the computer once again. (See "Recovering from a Firmware Update Error" on page 20).

## J3, J7, J12: Reserved

# Brief Description of Standards RS-232, RS-422, & RS-644 (LVDS)

## RS-232

Short for ***r****ecommended* ***s****tandard-232C,* a standard interface approved by the Electronic Industries Association (EIA) connecting serial devices.

The standards for RS-232 and similar interfaces usually restrict RS-232 to 256kbps or less and line lengths of 15M (50 ft) or less.

**Transmitted Data (TxD)** This signal is active when data is transmitted from the DTE device to the DCE device. When no data is transmitted, the signal is held in the mark condition (logic '1', negative voltage).

**Received Data (RxD)** This signal is active when the DTE device receives data from the DCE device. When no data is transmitted, the signal is held in the mark condition (logic '1', negative voltage).

DTE (Data Terminal Equipment)
DCE (Data Communication Equipment)

## RS-422

RS-422 uses a twisted-pair wire (i.e., 2 wires) for each signal. The differential drive voltage swing is 0 to +5V. RS-422 does not have tri-state capability (its driver is always enabled) and it is therefore usable only in point-to-point communications.

Although RS-422 is noise resistant, due to being differential data can still be damaged by EMI/RFI. A shielded cable can protect the transmitters/receivers from EMI/RFI.

## RS-644 (LVDS)

LVDS (Low-Voltage Differential Signaling): method to communicate data using a very low voltage swing (about 350mV) over two differential PCB traces or a balanced cable. LVDS allows single channel data transmission at hundreds of Megabits per second (Mbps).

# Camera Link Interface

## Camera Link Overview

Camera Link is a communication interface for vision applications developed as an extension of National Semiconductor's Channel Link technology. The advantages of the Camera Link interface are that it provides a standard digital camera connection specification, a standard data communication protocol, and simpler cabling between camera and frame grabber.

The Camera Link interface simplifies the usage of increasingly diverse cameras and high signal speeds without complex custom cabling. For additional information concerning Camera Link, see **http://www.pulnix.com/CameraLink.html**.

### Rights and Trademarks

Note: The following text is extracted from the Camera Link Specification (October 2000).

PULNiX America, Inc., as chair of this ad hoc Camera Link committee, has applied for U.S. trademark protection for the term "Camera Link" to secure it for the mutual benefit of industry members. PULNiX will issue a perpetual royalty-free license to any industry member (including competitors) for the use of the "Camera Link" trademark on the condition that it is used only in conjunction with products that are fully compliant to this standard. PULNiX will not require licensed users of the trademark to credit PULNiX with ownership.

3M™ is a trademark of the 3M Company.

Channel Link™ is a trademark of National Semiconductor.

Flatlink™ is a trademark of Texas Instruments.

Panel Link™ is a trademark of Silicon Image.

# Data Port Summary

The Camera Link interface has three configurations. A single Camera Link connection is limited to 28 bits requiring some cameras to have multiple connections or channels. The naming conventions for the three configurations are:

- Base: Single Channel Link interface, single cable connector.
- Medium: Two Channel Link interface, two cable connectors.
- Full: Three Channel Link interface, two cable connectors.

A single Camera Link port is defined as having an 8-bit data word. The "Full" specification supports 8 ports labeled as A to H.

# Camera Signal Summary

### Video Data

Four enable signals are defined as:

- FVAL Frame Valid (FVAL) is defined HIGH for valid lines.
- LVAL Line Valid (LVAL) is defined HIGH for valid pixels.
- DVAL Data Valid (DVAL) is defined HIGH when data is valid.
- Spare A spare has been defined for future use.

All four enables must be provided by the camera on each Channel Link. All unused data bits must be tied to a known value by the camera.

### Camera Controls

Four LVDS pairs are reserved for general-purpose camera control, defined as camera inputs and frame grabber outputs.

- Camera Control 1 (CC1)
- Camera Control 2 (CC2)
- Camera Control 3 (CC3)
- Camera Control 4 (CC4)

Note: the X64-CL by default implements the control lines as follows
(using DALSA Corporation terminology).

(CC1) EXYNC
(CC2) PRIN
(CC3) FORWARD
(CC4) HIGH

### Communication

Two LVDS pairs have been allocated for asynchronous serial communication to and from the camera and frame grabber. Cameras and frame grabbers should support at least 9600 baud.

- SerTFG Differential pair with serial communications to the frame grabber.
- SerTC Differential pair with serial communications to the camera.

The serial interface protocol is one start bit, one stop bit, no parity, and no handshaking.

# Camera Link Cables

For additional information on Camera Link cables and their specifications, visit the following web sites:

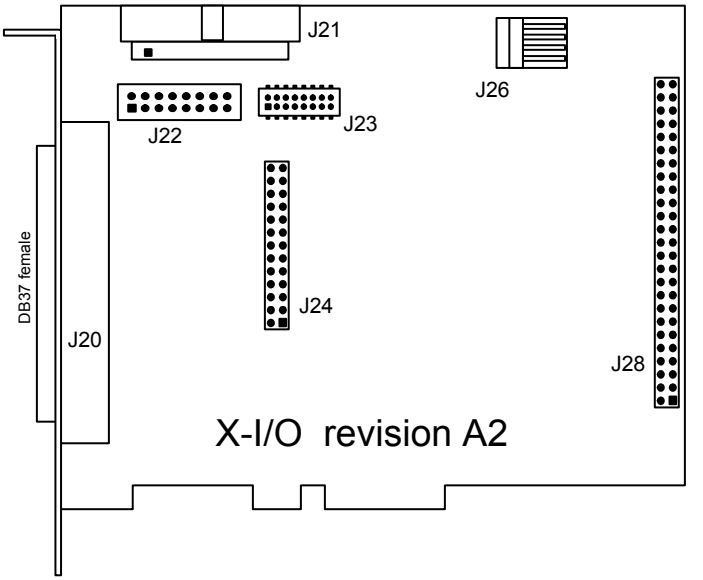
| | |
|---|---|
| 3 M | http://www.3m.com/us/electronics_mfg/interconnects/<br>*(enter Camera Link as the search keyword)* |
| Nortech Systems | http://www.nortechsys.com/intercon/CameraLinkMain.htm |

# Appendix: X-I/O Module Option

## X-I/O Module Overview

- The X-I/O module requires X64-CL Express board driver version 1.00 (or later) and Sapera LT version 5.30 (or later).
- Occupies an adjacent slot to the X64-CL Express. Slot can be either PCI-32 or PCI-64—no PCI signals or power are used.
- Connects to the X64-CL Express via a 26 pin flat ribbon cable. J21 on X-I/O to J4 on X64-CL Express. Note that the external signals connector bracket is removed.
- All X64-CL Express external signals, such as trigger, shaft encoder, strobe, are available on the X-I/O DB37. See "DB37 Pinout Description" on page 98.
- X-I/O provides 8 TTL outputs software selectable as NPN (current sink) or PNP (source driver) type drivers. See "TTL Output in NPN Mode: Electrical Details" on page 99 and "TTL Output in PNP Mode: Electrical Details" on page 100.
- X-I/O provides 2 opto-coupled inputs. See "Opto-coupled Input: Electrical Details" on page 101.
- X-I/O provides 6 TTL level inputs with software selectable transition point. See "TTL Input Electrical Details" on page 101.
- X-I/O provides both +5 volt and +12 volt power output pins on the DB37, where power comes directly from the host system power supply.
- Onboard flash memory to store user defined power up I/O states.

## X-I/O Module Connector List & Locations




| | |
|---|---|
| J20 | DB37 female external signals connector. |
| J21 | 26 pin header connector<br>(interconnect to the X64-CL Express via supplied ribbon cable). |
| J22, J23, J24, J28 | Reserved. |
| J26 | Connect PC power via floppy drive power cable. |

# X-I/O Module Installation

**Grounding Instructions:** Static electricity can damage electronic components. Please discharge any static electrical charge by touching a grounded surface, such as the metal computer chassis, before performing any hardware installation. If you do not feel comfortable performing the installation, please consult a qualified computer technician. **Never** remove or install any hardware component with the computer power on.

## Board Installation

Installing an X-I/O Module to an existing X64-CL Express installation takes only a few minutes. Install the X-I/O board into the host system as follows:

- Power off the computer system that has the installed X64-CL Express board.
- Disconnect the external signals cable (OC-64CC-0TIO1) if it was used. Remove that cable bracket from the computer.
- Insert the X-I/O module into any free PCI slot (no PCI electrical connections are used), securing the bracket.
- Connect the X-I/O module 26 pin ribbon cable from J21 to the X64-CL Express board J4.
- Power on the computer again.
- For new X64-CL Express and X-I/O module installations, simply follow the procedure to install Sapera and the X64-CL Express driver (start with "Sapera LT Library Installation" on page 10).

## X64-CL Express and X-I/O Driver Update

- If both Sapera 5.30 and X64-CL Express driver need to be installed, follow the procedure "Sapera and Board Driver Upgrades" on page 10. This procedure steps through the upgrade of both Sapera and the board driver—typically required when installing the X-I/O module in the field.
- If the X64-CL Express installation already has the required Sapera and board driver version, install the X-I/O module and perform a firmware update as described in "Executing the Firmware Loader from the Start Menu" on page 13.

# X-I/O Module External Connections to the DB37

Users can assemble their interface cable, using some or all of the signals available on the X-I/O module DB37. Use a male DB37 with thumb screws for a secure fit. Wiring type should meet the needs of the imaging environment.

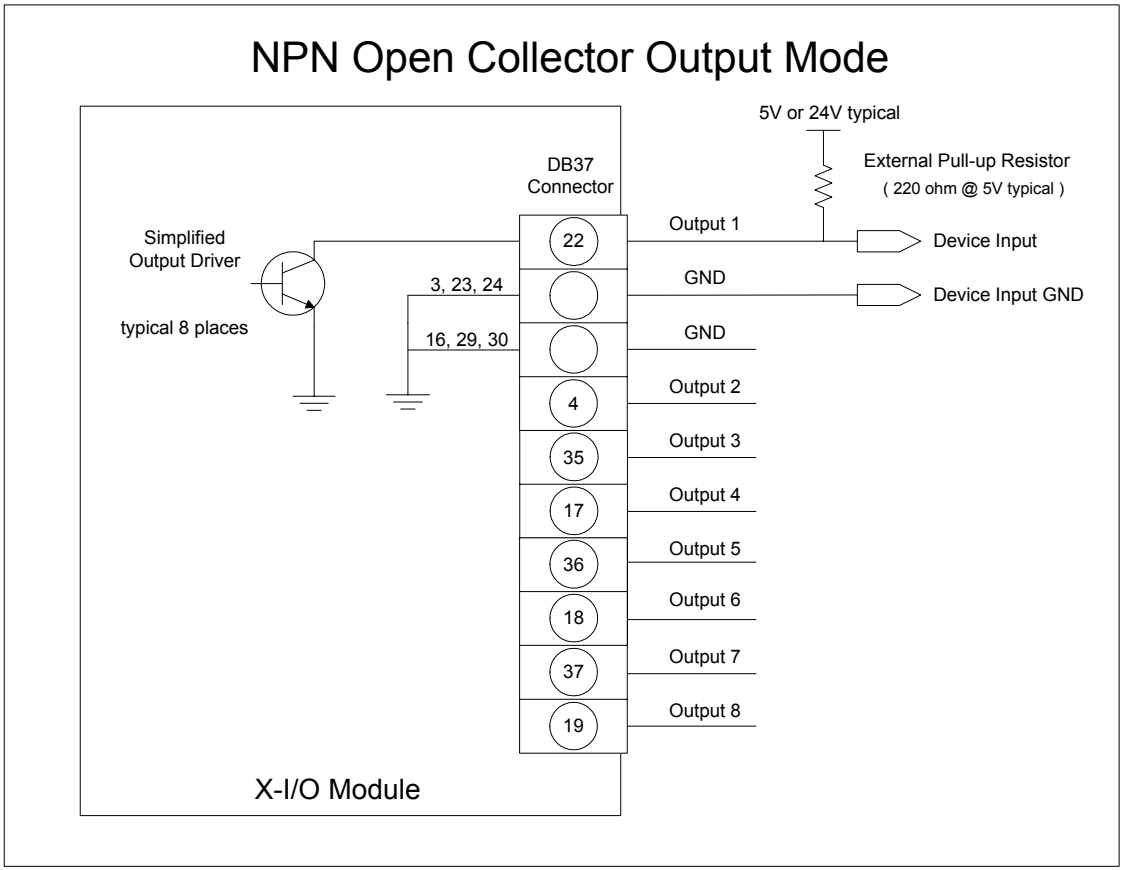
For the external signals Trigger Input, Shaft Encoder Input, and Strobe output, now available on the DB37, refer "J4: External Signals Connector" on page 81 to for signal details.

# DB37 Pinout Description

| Pin # | Signal | Description |
|---|---|---|
| 1 | IN_OPTO_1+ | Input #1 ( Opto-coupled ) |
| 20 | IN_OPTO_1- | |
| 2 | IN_OPTO_2+ | Input #2 ( Opto-coupled ) |
| 21 | IN_OPTO_2- | |
| 3, 23, 24 | Gnd | |
| 22 | OUT_TTL_1 | TTL output #1 |
| 4 | OUT_TTL_2 | TTL output #2 |
| 5 | USER_PWR | Power for the TTL Outputs in PNP mode |
| 6 | TrigIn 1+ | Trigger Input 1 + |
| 25 | TrigIn 1- | Trigger Input 1 - (TTL trigger GND) |
| 7 | TrigIn 2+ | Trigger Input 2 + |
| 26 | TrigIn 2- | Trigger Input 2 - (TTL trigger GND) |
| 8 | Phase A+ | Shaft Encoder Phase A+<br>(for TTL signals see "Connecting a TTL Shaft Encoder Signal to the LVDS/RS422 Input" on page 85) |
| 27 | Phase A- | Shaft Encoder Phase A- |
| 9 | Phase B+ | Shaft Encoder Phase B+ |
| 28 | Phase B- | Shaft Encoder Phase B- |
| 10 | Strobe 2 | TTL Strobe 2 output |
| 11 | Strobe 1 | TTL Strobe 1 output |
| 16, 29, 30 | Gnd | |
| 12 | Power | PC +5V (1A max) |
| 31 | Power | PC +12V (1A max) |
| 13 | IN_TTL_3 | Input #3 ( TTL ) |
| 32 | IN_TTL_4 | Input #4 ( TTL ) |
| 14 | IN_TTL_5 | Input #5 ( TTL ) |
| 33 | IN_TTL_6 | Input #6 ( TTL ) |
| 15 | IN_TTL_7 | Input #7 ( TTL ) |
| 34 | IN_TTL_8 | Input #8 ( TTL ) |
| 35 | OUT_TTL_3 | TTL output 3 |
| 17 | OUT_TTL_4 | TTL output 4 |
| 36 | OUT_TTL_5 | TTL output 5 |
| 18 | OUT_TTL_6 | TTL output 6 |
| 37 | OUT_TTL_7 | TTL output 7 |
| 19 | OUT_TTL_8 | TTL output 8 |

## TTL Output in NPN Mode: Electrical Details

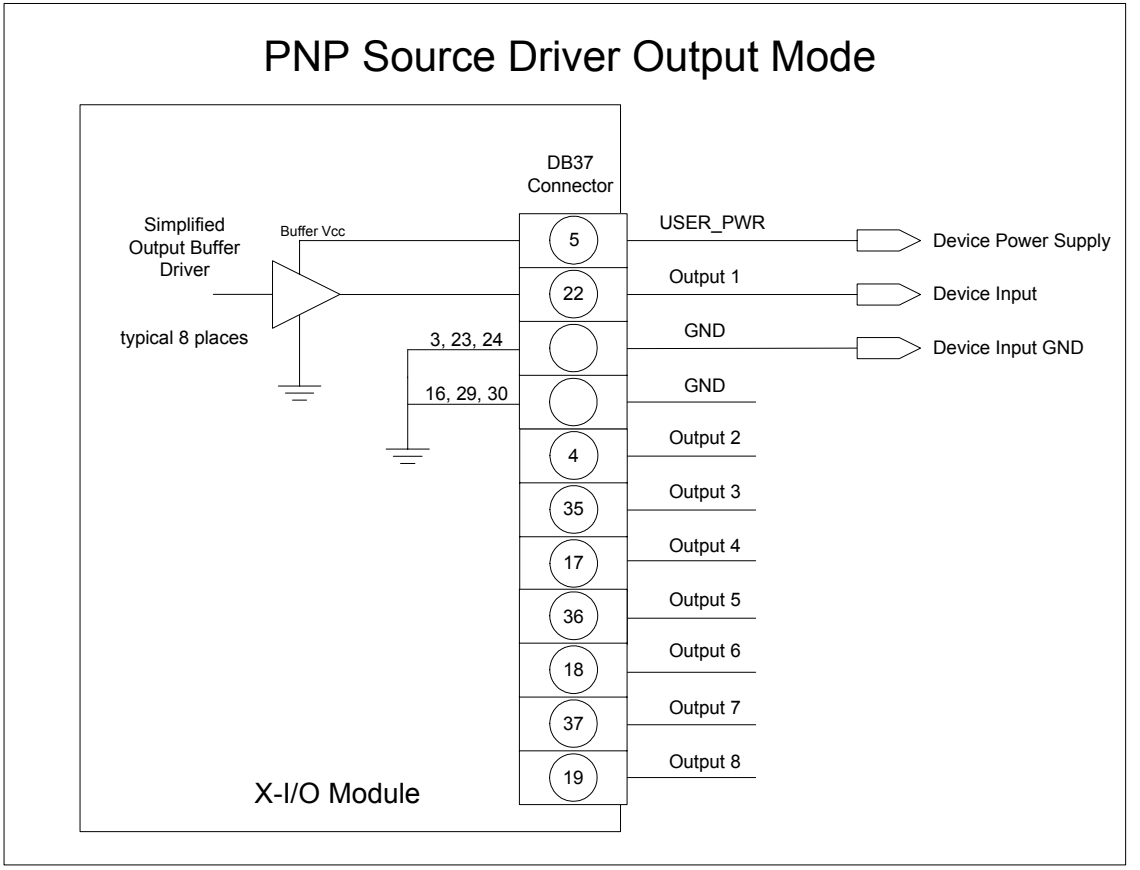
When the TTL outputs are configured for NPN mode (open collector - sink mode) the user is required to provide an external input pull-up resistor on the signal being controlled by the X-I/O output. A simplified schematic and important output specifications follow:




- Each output can sink 700 mA.
- Over-current thermal protection will automatically shut down the output device.

## TTL Output in PNP Mode: Electrical Details

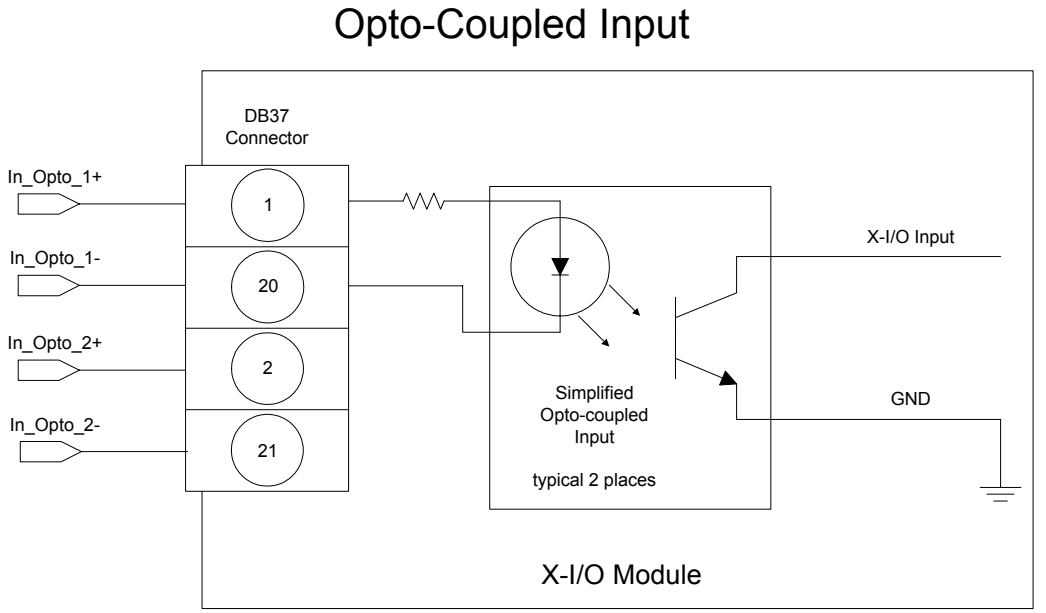
When the TTL outputs are configured for PNP mode (source driver) an external power supply is required to provide the buffer output supply voltage (USER_PWR). A simplified schematic and important output specifications follow:




- User provides the output power supply voltage ( 7 volts to 35 volts ).
- Source driver with over-current protection (all outputs will shut down simultaneously). The over-current fault circuit will protect the device from short-circuits to ground with supply voltages of up to 35V.

## Opto-coupled Input: Electrical Details

The two opto-coupled inputs can be used either with TTL or RS422 sources. A simplified input schematic and important electrical specifications are listed below.




| | |
|---|---|
| Input reverse breakdown voltage | 5 volts minimum |
| Maximum average forward input current | 25 mA |
| Maximum input frequency | 200 kHz |
| Maximum Sapera call-back rate | System processing dependent |

## TTL Input Electrical Details

The six TTL inputs are software configurable (see "Configuring User Defined Power-up I/O States" on page 102) for standard TTL logic levels or industrial logic systems (typically 24 volts). The design switch points are as follows:

- TTL level mode : trip point at 2V +/- 5%
- Industrial level mode: trip point at 16V +/- 5%

# X-I/O Module Sapera Interface

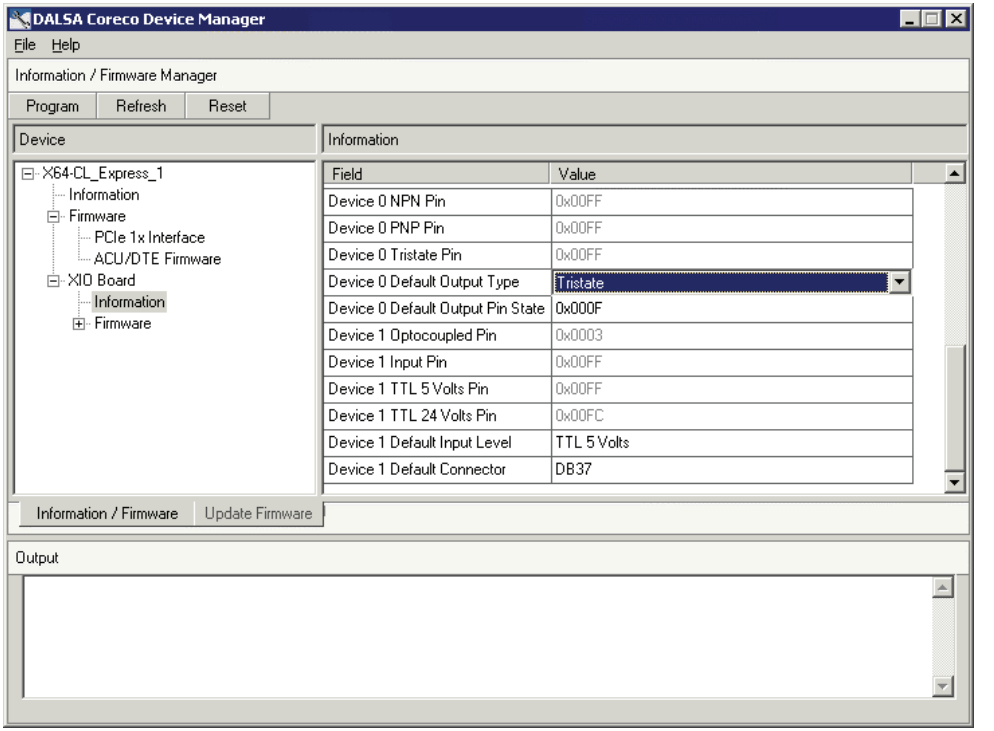
Sapera version 5.30 (or later) provides support for the X-I/O module via an I/O class and demonstration program. Users can use the demonstration program as is, or use the demo program source code to implement X-I/O controls within the custom imaging application.

This section describes configuring the X-I/O module power up state, using the X-I/O demo program, and describes the Sapera Class to program and read the X-I/O module along with sample code.

## Configuring User Defined Power-up I/O States

The X-I/O module power up state is stored onboard in flash memory. User configuration of this initial state is performed by the Coreco Device Manager program. Run the program via the windows start menu: (**Start • Programs • DALSA Coreco • X64-CL Express Device Driver • Coreco Device Manager**).

The Coreco Device Manager provides information on the installed X64-CL Express board and its firmware. With an X-I/O module installed, click on **XIO Board – Information**, as shown in the following figure.



The XIO information screen shows the current status of **Device 0**—the output device, and **Device 1**—the input device. A few items are user configurable for X-I/O board power up state. Click on the item to display a drop list of available capabilities, as described below.

- **Device 0 – Default Output Type**
  choose Tristate mode (i.e. output disconnected), or PNP mode, or NPN mode.
- **Device 0 – Default Output Pin State**
  A window is displayed to select a logic low or high state for each output pin. Click on each pin that should be logic high by default.
- **Device 1 – Default Input Level**
  Select the input logic level as TTL 5 Volts or TTL 24 Volts, dependent on the signal type being input to the X-I/O module.
- **Device 1 – Default Connector**
  DB37 is the supported output connector, as described in this section.

### Programming the User Configuration

After changing any user configurable X-I/O mode from the factory default state, click on the **Program** button (located on the upper left), to write the new default state to flash memory. The Coreco Device Manager message output window will display "Successfully updated EEProm". The program can now be closed.

## Using Sapera LT General I/O Demo

The Sapera General I/O demo program will control the I/O capabilities of any installed Sapera board product. The demo will present to the user only the controls pertaining to the selected hardware.

Run the demo via the windows start menu: (**Start • Programs • DALSA Coreco • Sapera LT • Demos • General I/O Demo**). The first menu presents a drop list of all installed Sapera Acquisition Devices with I/O capabilities. Select the X64-CL Express board is selected and click OK to continue.

### General I/O Module Control Panel

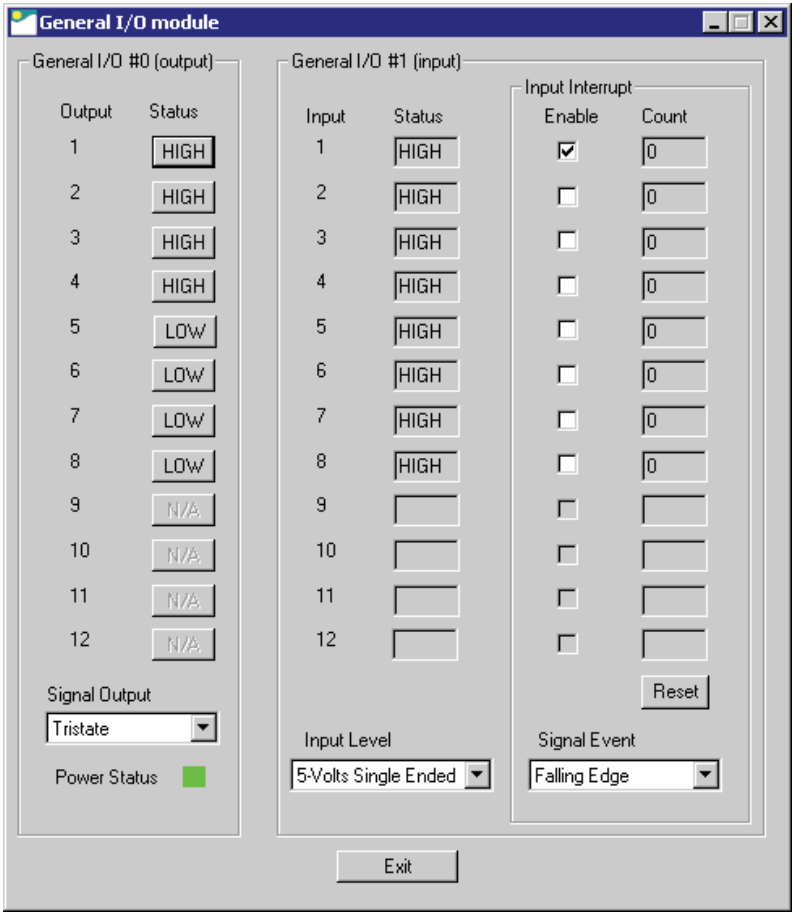
The I/O module control demo presents the I/O capabilities of the installed hardware. The following figure shows the X-I/O module connected to the X64-CL Express board.

**Output Pins**: The first column displays the current state of the eight output pins (I/O Device #0).

- The startup default state is user configured using the Coreco Device Manager program.
- The state of each output can be changed by clicking on its status button.
- Use the Signal Output drop menu to select the output mode (Tristate, PNP, NPN).

**Input Pins**: The second section provides input pin status (I/O device #1). Note that this program is a demo, therefore no action takes place on an input event.

- The first column reads the logic level present on each input. The Input Level drop menu changes the logic level from 5V TTL to 24V logic. Use the Coreco Device Manager program to select the default logic level type.
- The second column demonstrates activating interrupts on individual inputs. In this demo program, use the Enable box to activate the interrupt on an input. The Count box will tally detected input events. Use the Signal Event drop menu to select which input signal edge to detect. The Reset button clears all event counts.



# Sapera LT General I/O Demo Code Samples

The following source code was extracted from the General I/O demo program. The comments highlight the areas that an application developer needs for embedding X-I/O module controls within the imaging application.

### Main I/O Demo code

```
BOOL CGioMainDlg::OnInitDialog()
{

[ . . . ]

// some declarations
UINT32  m_gioCount;
int m_ServerIndex;
int m_ResourceIndex;


// Show the Server Dialog to select the acquisition device
CGioServer dlg(this);
if (dlg.DoModal() == IDOK)
{

    m_ServerIndex = dlg.GetServerIndex();
    m_ServerName = dlg.GetServerName();

    if ( m_ServerIndex != -1)
        {
        // Get the number of resources from SapManager for ResourceGio type by using
        // - the server index chosen in the dialog box
        // - the resource type to enquire for Gio
        m_gioCount=SapManager::GetResourceCount(m_ServerIndex,SapManager::ResourceGio);

        // Create all objects [see the function following]
        if (!CreateObjects()) { EndDialog(TRUE); return FALSE; }

        [ . . . ]

        //Loop for all resources
        for (UINT32 iDevice = 0; (iDevice < MAX_GIO_DEVICE) && (iDevice < m_gioCount);
                                  iDevice++)
        {
            [ . . . ]

            // direct read access to low-level Sapera C library capability to check
            // I/O Output module
            if (m_pGio[iDevice]->IsCapabilityValid(CORGIO_CAP_DIR_OUTPUT))
            status = m_pGio[iDevice]->GetCapability(CORGIO_CAP_DIR_OUTPUT,&capOutput);

            // direct read access to low-level Sapera C library capability to
            // check I/O Input module
            if (m_pGio[iDevice]->IsCapabilityValid(CORGIO_CAP_DIR_INPUT))
```

```
            status = m_pGio[iDevice]->GetCapability(CORGIO_CAP_DIR_INPUT,&capInput);

            [ . . . ]
            // Constructor used for I/O Output module dialog.
            if (capOutput)
            {
            m_pDlgOutput[iDevice] = new CGioOutputDlg(this, iDevice, m_pGio[iDevice]);
            }

            [ . . . ]

            // Constructor used for I/O Input module dialog.
            if (capInput)
            {
            m_pDlgInput[iDevice] = new CGioInputDlg(this, iDevice, m_pGio[iDevice]);
            }
        }   //end for
}   // end if

[ . . . ]
}
```

## Function CreateObjects()

```
BOOL CreateObjects()
{
CWaitCursor wait;

// Loop for all I/O resources
for (UINT32 iDevice = 0; (iDevice < MAX_GIO_DEVICE) && (iDevice < m_gioCount);
        iDevice++)
{
    // The SapLocation object specifying the server where the I/O resource is located
    SapLocation location(m_ServerIndex, iDevice);

        // The SapGio constructor is called for each resource found.
        m_pGio[iDevice] = new SapGio(location);

        // Creates all the low-level Sapera resources needed by the I/O object
        if (m_pGio[iDevice] && !*m_pGio[iDevice] && !m_pGio[iDevice]->Create())
        {
            DestroyObjects();
            return FALSE;
        }
    }
return TRUE;
}
```

### Output Dialog: CGioOutputDlg class (see Sapera Gui class)

```
void CGioOutputDlg::UpdateIO()
{
    UINT32 output=0;
    UINT32 state=0;
    BOOL status;
    [ . . . ]

    // We loop to get all I/O pins.
    for (UINT32 iIO=0; iIO < (UINT32)m_pGio->GetNumPins(); iIO++)
    {
        [ . . . ]

        // We set the current state of the current I/O pin by using
        // - the pin number on the current I/O resource
        // - the pointer to pin state
        //  ( SapGio ::PinLow if low and SapGio ::PinHigh if high)
        status = m_pGio->SetPinState(iIO, (SapGio::PinState)state);
    }

}
```

### Input Dialog: CGioInputDlg class. (see Sapera Gui class)

```
BOOL CGioInputDlg::Update()
{
    SapGio::PinState state = SapGio::PinState::PinLow;
    BOOL status = true;
    UINT32 iIO;
    UINT32 jIO;

    if (m_pGio == NULL)
        return FALSE;

    // We loop to get all I/O pins.
    for (iIO=0; iIO < (UINT32)m_pGio->GetNumPins(); iIO++)
    {
        m_pGio->SetDisplayStatusMode(SapManager::StatusLog, NULL);
        // We get the current state of the current I/O pin by using
        // the pin number on the current I/O resource
        // the pointer to pin state
        //  ( SapGio ::PinLow if low and SapGio ::PinHigh if high)

        status = m_pGio->GetPinState(iIO, &state);
        m_pGio->SetDisplayStatusMode(SapManager::StatusNotify, NULL);

        [ . . . ]
    }

    [ . . . ]
}
```

## I/O Event Handling

```
void CGioInputDlg::GioCallbackInfo(SapGioCallbackInfo *pInfo)
{
CGioInputDlg* pInputDlg;
CString strEventCount;

    // We get the application context associated with I/O events
    pInputDlg = (CGioInputDlg*)pInfo->GetContext();

    // We get the current count of I/O events
    strEventCount.Format("%d", pInfo->GetEventCount());

    // We get the I/O pin number that generated an I/O event and apply the changes.
    pInputDlg->m_GioEventCount[pInfo->GetPinNumber()]++;
}
```

# DALSA Coreco Contact Information

## Sales Information

**Visit our web site:** http://www.imaging.com/

**Email:** mailto:info@coreco.com

### Corporate Headquarters

DALSA Coreco
7075 Place Robert-Joncas Suite #142
St. Laurent, Quebec
H4M 2Z2
Canada

Tel: (514) 333-1301
Fax: (514) 333-1388

### US Sales Office

DALSA Coreco
Bldg. 8 2nd Floor
900 Middlesex Turnpike
Billerica, Ma. 01821

Tel: (978) 670-2000
Fax: (978) 670-2010

# Technical Support

## Any support question or request can be submitted via our web site:

| | |
|---|---|
| Technical support form via our web page:<br>Support requests for imaging product installations,<br>Support requests for imaging applications | http://www.imaging.com/support |
| Camera support information | http://www.imaging.com/camsearch |
| Product literature and driver updates | http://www.imaging.com/download |

# Glossary of Terms

**Bandwidth**
Describes the measure of data transfer capacity. PCI devices must share the maximum PCI bus bandwidth when transferring data to and from system memory or other devices.

**CAM**
Sapera camera file that uses the file extension CCA by default. Files using the CCA extension, also called CAM files (CORECO CAMERA files), contain all parameters which describe the camera video signal characteristics and operation modes (i.e. what the camera outputs).

**Channel**
Camera data path that includes all parts of a video line.

**Checksum**
A value used to ensure data is stored without error. It is created by calculating the binary values in a block of data using some algorithm and storing the results with the data.

**CMI**
**C**lient **M**odification **I**nstruction. A client requested engineering change applied to a DALSA Coreco board product to support either a non-standard function or custom camera.

**Contiguous memory**
A block of physical memory, occupying consecutive addresses.

**CRC**
Proprietary Sapera raw image data file format that supports any Sapera buffer type and utilizes an informative file header. Refer to the *Sapera Basic Modules Reference Manual* "Buffer File Formats" section.

**Firmware**
Software such as a board driver that is stored in nonvolatile memory mounted on that board.

**Frame buffer**
An area of memory used to hold a frame of image data. A frame buffer may exist on the acquisition hardware or be allocated by the acquisition hardware device driver in host system memory.

**Grab**
Acquiring an image frame by means of a frame grabber.

**Host**
Refers to the computer system that supports the installed frame grabber.

**Host buffer**
Refers to a frame buffer allocated in the physical memory of the host computer system.

**LSB**
Least Significant Bit in a binary data word.

**MSB**
Most Significant Bit in a binary data word.

**PCI 32**
Peripheral Component Interconnect. The PCI local bus is a 32-bit high-performance expansion bus intended for interconnecting add-in boards, controllers, and processor/memory systems.

**PCI 64**
A superset of the PCI specification providing a 64 bit data path and a 66 MHz clock.

**Pixel**
Picture Element. The number of pixels describes the number of digital samples taken of the analog video signal. The number of pixels per video line by the number of active video lines describes the acquisition image resolution. The binary size of each pixel (i.e., 8-bits, 15-bits, 24-bits) defines the number of gray levels or colors possible for each pixel.

**RAW**
A Sapera data file format where there is no header information and that supports any Sapera buffer type. Refer to the *Sapera Basic Modules Reference Manual* "Buffer File Formats" section.

**RISC**
(**R**educed **I**nstruction **S**et **C**omputer) A computer architecture that reduces chip complexity by using simpler instructions.

**Scatter Gather**

Host system memory allocated for frame buffers that is virtually contiguous but physically scattered throughout all available memory.

**Tap**

Data path from a camera that includes a part of or whole video line. When a camera tap outputs a partial video line, the multiple camera tap data must be constructed by combining the data in the correct order.

**VIC**

Sapera camera parameter definition file that uses the file extension CVI by default. Files using the CVI extension, also know as VIC files (CORECO VIDEO files), contain all operating parameters related to the frame grabber board (i.e. what the frame grabber can actually do with camera controls or incoming video).

# Index

## A



## B



## C

## D



## E



## F



## H



## I

## J



## L



## M



## N



## O



## P



## Q



## S



## T



## U



## V

## W



## X