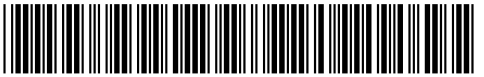
DALSA • 7075 Place Robert-Joncas, Suite 142 • St-Laurent, Quebec, H4M 2Z2 • Canada
http://www.imaging.com

X64 Xcelera-CL PX4™

User's Manual

Edition 1.10

Part number OC-X4CM-PUSR0



NOTICE

Edition 1.10 printed on: April 18, 2007

Document Number: OC-X4CM-PUSR0

Printed in Canada

Contents

X64 Xcelera-CL PX4 Overview

Product Part Numbers

X64 Xcelera-CL PX4 Board

Item	Product Number
All models have 85MHz Pixel Clock X64 Xcelera-CL PX4 Full with 128 MB of memory X64 Xcelera-CL PX4 Dual with 128 MB of memory	 OR-X4C0-XPF00 OR-X4C0-XPD00
X-I/O Module (optional): provides an additional 8 input & 8 output general I/Os (see "Appendix: X-I/O Module Option" on page 103)	OC-IO01-STD00
For OEM clients, this manual in printed form, is available on request	OC-X4CM-USER0

X64 Xcelera-CL PX4 Software

Item	Product Number
Sapera LT version 6.00 or later (required but sold separately) 1. Sapera LT: Provides everything you will need to build your imaging application 2. Current Sapera compliant board hardware drivers 3. Board and Sapera documentation (compiled HTML help, and Adobe Acrobat® (PDF) formats)	OC-SL00-0000000
(optional) Sapera Processing Imaging Development Library includes over 600 optimized image processing routines.	Contact Sales at **DALSA**

X64 Xcelera-CL PX4 Cables & Accessories

Item	Product Number
(optional) X64 Xcelera-CL PX4 can be shipped with an External Signals Connector Bracket Assembly, either with a **DB37 or DB25** connector (see the two product numbers below). Either cable, if required, should be specified at the time of order. Note: clients requiring more I/O connections must add the optional X-I/O Module.	
DB37 assembly see "External Signals Connector Bracket Assembly (Type 1)" on page 94. This cable assembly connects to J4.	OR-X4CC-IOCAB
DB25 assembly see "External Signals Connector Bracket Assembly (Type 2)" on page 97. Provides direct compatibility with external cables made for products such as the X64-CL iPro. This cable assembly connects to J4.	OR-X4CC-0TIO2
(optional) Power interface cable required when supplying power to cameras	OC-COMC-PCPWR
(optional) Camera Link Video Input Cable: 1 meter 2 meter	 OC-COMC-CLNK0 OC-COMC-CLNK6

About the X64 Xcelera-CL PX4 Frame Grabber

Series Key Features

- Monochrome or RGB Camera Link
- Flat Field Correction
- Bayer Filter Decoding
- Output lookup tables available for each mode
- Vertical Flip supported on board
- RoHS compliant

See “Technical Specifications” on page **79** for detailed information.

User Programmable Configurations

Use the X64 Xcelera-CL PX4 firmware loader function in the DALSA Device manager utility to select firmware for one of the supported modes. Firmware selection is made either during driver installation or manually later on (see "Firmware Update: Manual Mode" on page 14).

For the X64 Xcelera-CL PX4 Full board the firmware choices are:

- **Full Camera Link Input with Flat Field Correction** (*installation default selection*)
  Support for one Base or one Medium or one Full Camera Link port. Flat Field Correction (FFC) includes Fixed Pattern Noise (FPN), Pixel Replacement, Photo Response Non Uniformity (PRNU), and Shading Correction.
- **Base Camera Link Input** with Bayer Filter Decoding.
- **Camera Link Input with 10 Tap Format 2 support** with Flat Field Correction
  Supports camera such as the Basler A504. See "Supporting Non-Standard CameraLink Cameras" on page 52 for details.

For the X64 Xcelera-CL PX4 Dual board the firmware choices are:

- **Dual Base Camera Link Input with Flat Field Correction** (*installation default selection*)
  Support for one or two Base Camera Link ports. Flat Field Correction (FFC) includes Fixed Pattern Noise (FPN), Pixel Replacement, Photo Response Non Uniformity (PRNU), and Shading Correction.
- **Dual Base Camera Link Input** with Bayer Filter Decoding.
- **One Medium Camera Link Input** with Flat field correction.

ACUPlus: Acquisition Control Unit

ACUPlus consists of a grab controller, 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 64-bit/pixels. Additionally, alternate tap configurations can support up to 8 taps of 8-bits each or optionally 10 tap with alternate firmware.

DTE: Intelligent Data Transfer Engine

The X64 Xcelera-CL PX4 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 x4 Interface

The X64 Xcelera-CL PX4 is a universal PCI Express x4 board, compliant with the PCI Express 1.0a specification. The X64 Xcelera-CL PX4 board achieves transfer rates up to 680 Mbytes/sec. with all taps used when connected to a corresponding camera or sensor.

The X64 Xcelera-CL PX4 board occupies one PCI Express x4 expansion slot and one chassis opening (two slots with the optional X-I/O Module Option).

Important:

- Older computers may not support the maximum data transfer bandwidth defined for PCI Express x4.
- The X64 Xcelera-CL PX4 board can also be used in an PCI Express x8 slot typically without issue.
- If the computer only has a PCI Express x16 slot, direct installation tests or the computer documentation is required to know if the X64 Xcelera-CL PX4 is supported. It has been seen that many computer motherboards only support x16 products in x16 slots (commonly used with graphic video boards).

Advanced Controls Overview

Visual Indicators

X64 Xcelera-CL PX4 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

Trigger inputs and strobe signals are provided to precisely synchronize 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 present 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

An 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 independent general purpose software controllable I/O signals to the X64 Xcelera-CL PX4. 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 Xcelera-CL PX4 board or installed into the computer system at a later time. The module occupies one adjacent PCI slot and connects to the X64 Xcelera-CL PX4 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 Xcelera-CL PX4 board driver version 1.00 or later and Sapera LT version 6.0 or later.

See "Appendix: X-I/O Module Option" on page 103 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 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 X64 Xcelera-CL PX4

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 Board Driver

When installing a new version of Sapera or a DALSA 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 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 Technical Support (see "Technical Support" on page 118).

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 • X64 Xcelera-CL PX4 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 Xcelera-CL PX4 Hardware and Driver" on page 13.
- Note that you can not install a DALSA board driver without Sapera LT installed on the computer.

Sapera and Board Driver Upgrades

When both Sapera and the 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 • X64 Xcelera-CL PX4 Driver • Modify-Repair-Remove**.
- Click on **Remove** to uninstall the board driver.
- From the Windows start menu select **Start • Programs • DALSA • 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 12 and "Installing X64 Xcelera-CL PX4 Hardware and Driver" on page 13 for installation procedures.

Sapera LT Library Installation

Note: to install Sapera LT and the X64 Xcelera-CL PX4 device driver, logon to the workstation as 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.

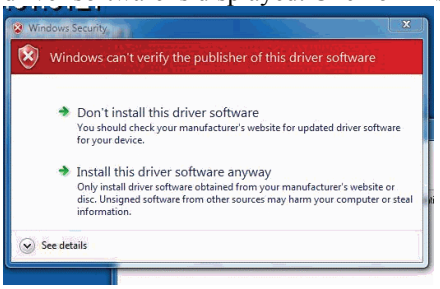
- Insert the DALSA Sapera CD-ROM. If **AUTORUN** is enabled on your computer, the installation menu is presented.
- If **AUTORUN** is not enabled, use Windows Explorer and browse to the root directory of the CD-ROM. Execute **launch.exe** to start the installation menu and install the required Sapera 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 Xcelera-CL PX4 Hardware and Driver

In a Windows 2000/XP/Vista System

- Turn the computer off and open the computer chassis to allow access to the expansion slot area.
- Install the X64 Xcelera-CL PX4 into a free PCI Express x4 expansion slot. The X64 Xcelera-CL PX4 could also be installed in a PCI Express x8. Note that some computer's x16 slot may support the X64 Xcelera-CL PX4. The user needs to test each computer to verify support of a x4 product.
- 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 Xcelera-CL PX4 and start its **Found New Hardware Wizard**. Click on the **Cancel** button to close the Wizard.
- Insert the DALSA Sapera CD-ROM. If **AUTORUN** is enabled on your computer, the installation menu is presented. Install the X64 Xcelera-CL PX4 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 installation menu and install the X64 Xcelera-CL PX4 driver.
- Reboot the computer when prompted. During the early stages of the Windows reboot, the X64 Xcelera-CL PX4 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**, if the **Digital Signature Not Found** message is displayed, click on Yes to continue the X64 Xcelera-CL PX4 driver installation. Reboot the computer when prompted.
- When using **Windows XP**, if a message stating that the X64 Xcelera-CL PX4 software has not passed **Windows Logo testing** is displayed, click on **Continue Anyway** to finish the X64 Xcelera-CL PX4 driver installation. Reboot the computer when prompted.
- When using **Windows Vista**, a message stating that Windows can not verify the publisher of the driver software is displayed. Click on **Install this driver software anyway**.



X64 Xcelera-CL PX4 Firmware Loader

After Windows boots, the Device Manager-Firmware Loader program automatically executes. It will determine if the X64 Xcelera-CL PX4 requires a firmware update. If firmware is required, the dialog displays and it also allows the user to load firmware for alternate operational modes of the X64 Xcelera-CL PX4.

***Important*:** In the vary rare case of firmware loader errors please see "Recovering from a Firmware Update Error" on page 22.

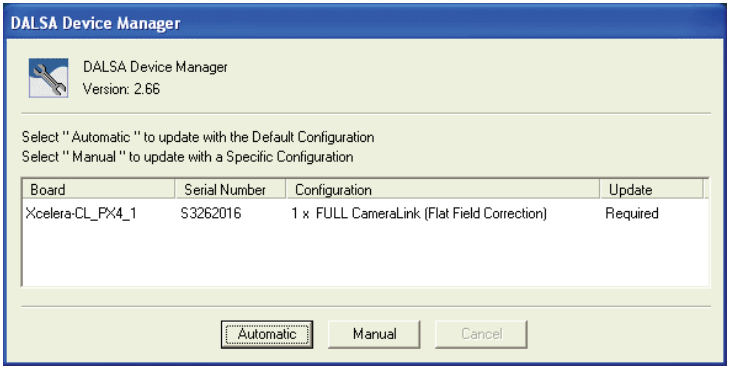
Firmware Update: Automatic Mode

Click **Automatic** to update the X64 Xcelera-CL PX4 firmware. The **X64 Xcelera-CL PX4 Full** supports three firmware configurations with the default being a Full, Medium, or Base camera with Flat Field correction.

The **X64 Xcelera-CL PX4 Dual** board supports three firmware configurations with the default being dual Base cameras with Flat Field correction.

See “Series Key Features” on page 7 and “User Programmable Configurations” on page 7 for details on all supported modes, which can be selected via a manual firmware update.

If there are multiple X64 Xcelera-CL PX4 boards in the system, all will be updated with new firmware. If any installed X64 Xcelera-CL PX4 board installed in a system already has the correct firmware version, an update is not required. In the following screen shot, a single X64 Xcelera-CL PX4 Full board is installed in the system and the default configuration is ready to be programmed.



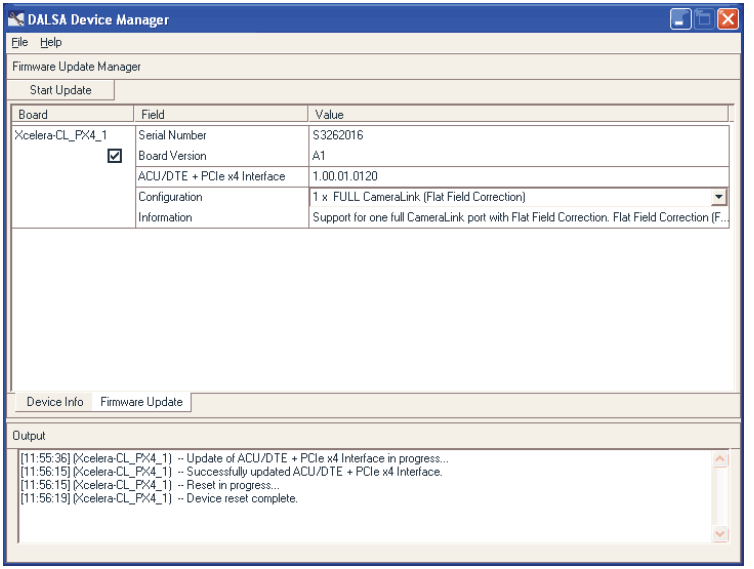
Firmware Update: Manual Mode

Select **Manual** mode to load firmware other then the default version or when, in the case of multiple X64 Xcelera-CL PX4 boards in the same system, each requires different firmware.

The figure below shows the Device Manager manual firmware screen. Information on all installed X64 Xcelera-CL PX4 boards, their serial numbers, and their firmware components are shown.

A manual firmware update is made as follows:

- Select the X64 Xcelera-CL PX4 to update 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 device 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 • X64 Xcelera-CL PX4 Driver • Firmware Update**. A firmware change after installation would be required to select a different Camera Link configuration mode. See "User Programmable Configurations" on page 7.

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 "J2: Camera Link Connector 1 " on page 86). The X64 Xcelera-CL PX4 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 Xcelera-CL PX4 serial port supports communication speeds from 9600 to 115 kbps.

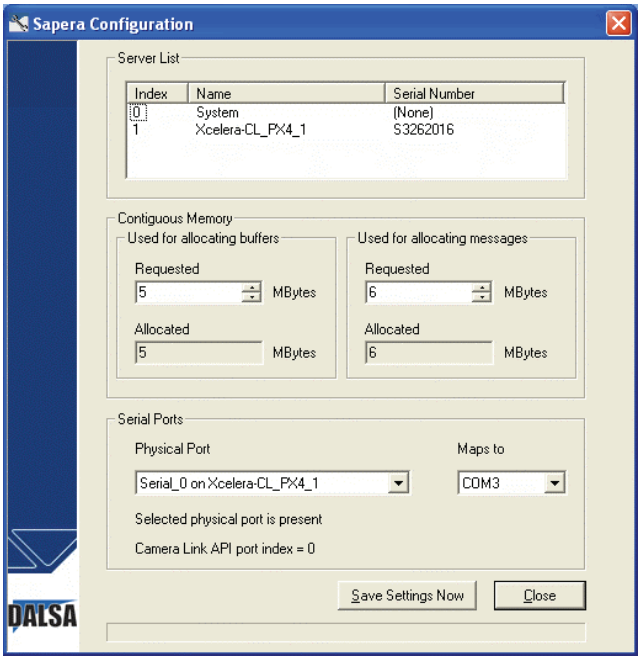
Note: if your serial communication program can directly select the X64 Xcelera-CL PX4 serial port then mapping to a system COM port is not necessary.

The X64 Xcelera-CL PX4 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 • Sapera LT • Sapera Configuration**.

COM Port Assignment

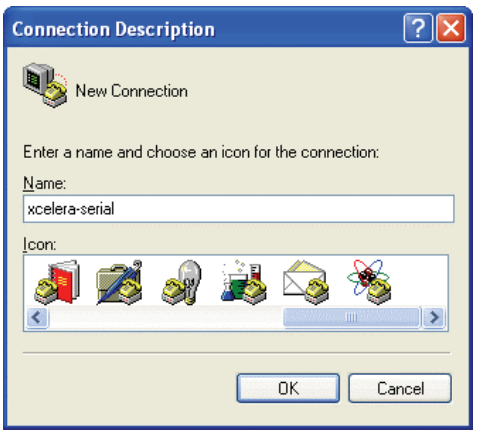
The lower section of the Sapera Configuration program screen contains the serial port configuration menu. Configure as follows:

- 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 Xcelera-CL PX4 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.

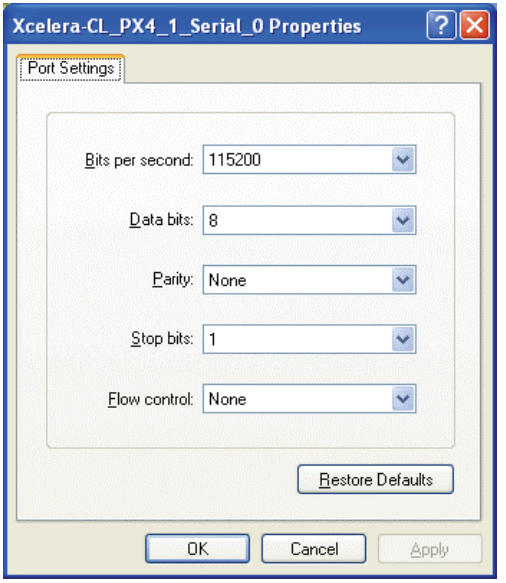


Setup Example with Windows HyperTerminal

- Run HyperTerminal and type a name for the new connection when prompted. Then click OK.
- On the following dialog screen select the port to connect with. The port could be the COM port mapped to the X64 Xcelera-CL PX4 or the COM device as shown in this example.



- 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 Xcelera-CL PX4 serial port does not support hardware flow control.



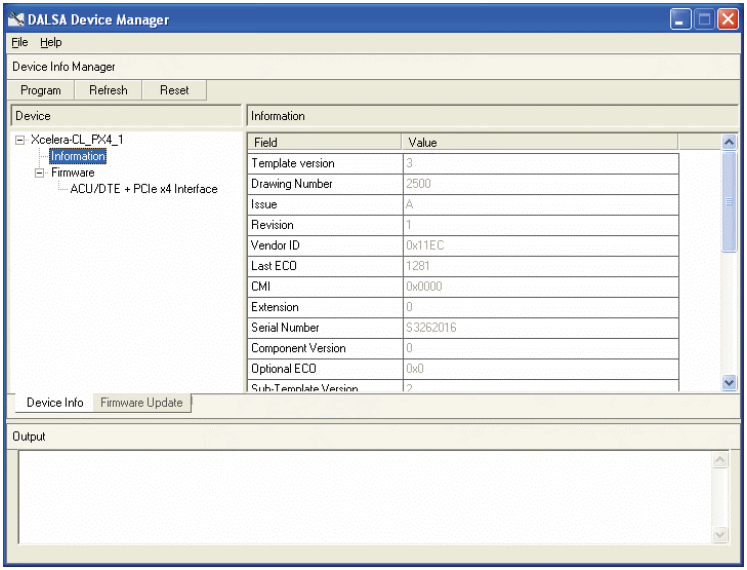
Displaying X64 Xcelera-CL PX4 Board Information

The Device Manager program also displays information about the X64 Xcelera-CL PX4 boards installed in the system. To view board information run the program via the Windows Start Menu shortcut **Start • Programs • DALSA • X64 Xcelera-CL PX4 Device Driver • CorDeviceManager**.

Device Manager – Board Viewer

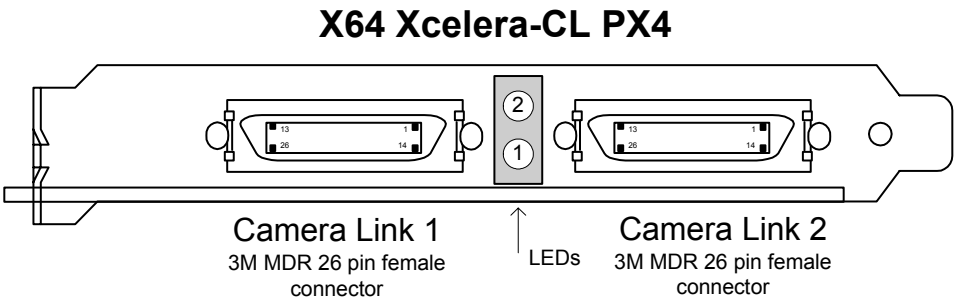
The following screen image shows the Device Manager program with the Information/Firmware tab active. The left window displays all X64 Xcelera-CL PX4 boards in the system and their individual device components. The right window displays the information stored in the selected board device. This example screen shows the X64 Xcelera-CL PX4 information contained in the EEProm component.

The X64 Xcelera-CL PX4 device manager report file (BoardInfo.txt) is generated by clicking **File • Save Device Info**. This report file may be requested by DALSA Technical Support to aid in troubleshooting installation or operational problems.



Camera to Camera Link Connections

X64 Xcelera-CL PX4 End Bracket



The hardware installation process is completed with the connection of a supported camera to the X64 Xcelera-CL PX4 board using Camera Link cables (see "Camera Link Cables" on page **101**).

- The X64 Xcelera-CL PX4 Full board supports a camera with one or two Camera Link MDR-26 connectors (one Base or one Medium or one Full– see "Data Port Summary" on page 100 for information on Camera Link configurations).
- Connect the camera to Camera Link 1 with a Camera Link cable. When using a Medium camera, connect the second camera cable to Camera Link 2.
- The X64 Xcelera-CL PX4 Dual board supports one or two cameras (one or two Base, or one Medium – see "Data Port Summary" on page 100 for information on Camera Link configurations).
- Connect camera 1 to the Camera Link 1 connector with a Camera Link cable. Connect camera 2 to the Camera Link 2. When using a Medium camera, connect the second camera cable to Camera Link 2.

Refer to section "Connector and Switch Specifications" on page **83** for details on the Camera Link connectors.

Note: If the camera is powered by the X64 Xcelera-CL PX4, refer to "External Signals Connector Bracket Assembly" on page 94 for power connections.

Contact DALSA or browse our web site http://www.imaging.com/camsearch for the latest information on X64 Xcelera-CL PX4 supported cameras.

Configuring Sapera

Viewing Installed Sapera Servers

The Sapera configuration program (**Start • Programs • DALSA • 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 "Grab Demo Overview" on page 41) 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 Xcelera-CL PX4 (and the X64 family of products) has been tested by DALSA 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 Technical Support. Note that information provided within this section will be updated with the latest information DALSA can provide for each manual version released.

If you require help and need to contact DALSA Technical Support, make detailed notes on your installation and/or test results for our technical support to review. See "Technical Support" on page 118 for contact information.

Recovering from a Firmware Update Error

This procedure is required if any failure occurred while updating the X64 Xcelera-CL PX4 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 Xcelera-CL PX4 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.

- Note that this procedure may require removing the X64 Xcelera-CL PX4 board several times from the computer.

- ***Important*:** Referring to the board's user manual (in the connectors and jumpers reference section), identify the configuration switch location. The Boot Recovery Mode switch for the X64 Xcelera-CL PX4 is SW3-1 (see "SW3: Additional Controls" on page 85).
- Shut down Windows and power OFF the computer.
- Move the switch SW3-1 to ON, for the boot recovery mode position. (The default position is SW3-1 to OFF for normal operation).
- 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 15).
- When the update is complete, shut down Windows and power off the computer.
- Set the SW3-1 switch back to the OFF position (i.e. default position) and power on the computer once again.
- Verify that the frame grabber is functioning by running a Sapera application such as CamExpert. The Sapera application will now be able to communicate with the X64 Xcelera-CL PX4 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.

Device Manager Program

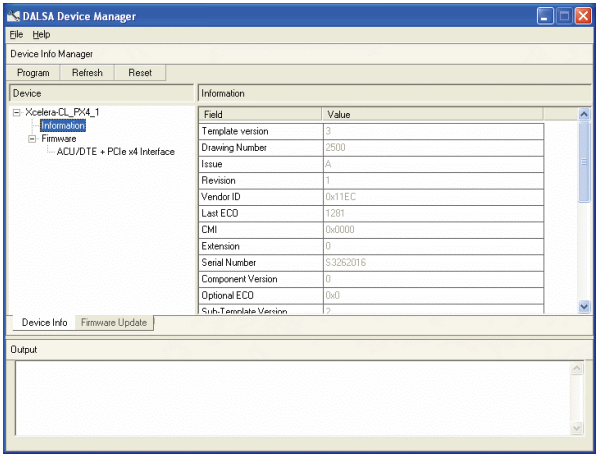
The Device Manager program provides a convenient method of collecting information about the installed X64 Xcelera-CL PX4 Full. System information such as operating system, computer CPU, system memory, PCI configuration space, plus X64 Xcelera-CL PX4 firmware information can be displayed or written to 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 Xcelera-CL PX4 Full.

Execute the program via the Windows Start Menu shortcut **Start • Programs • DALSA • X64 Xcelera-CL PX4 Device Driver • CorDeviceManager**. If the Device Manager program does not run, it will exit with a message that the board was not found. Since the X64 Xcelera-CL PX4 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 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.

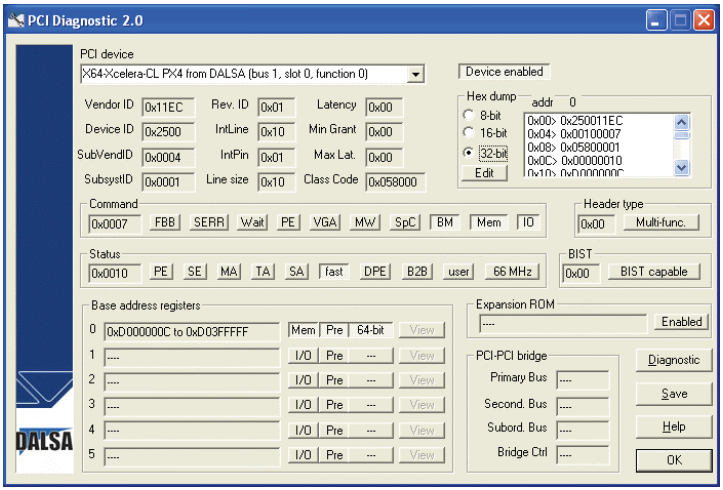


- Select **Information** to display identification and information stored in the X64 Xcelera-CL PX4 firmware.
- Select **Firmware** to display version information for the firmware components.
- Select one of the firmware components to load *custom* firmware when supplied by DALSA engineering for a future feature.
- Click on **File • Save Device Info** to save all information to a text file. Email this file when requested by Technical Support.

PCI Configuration

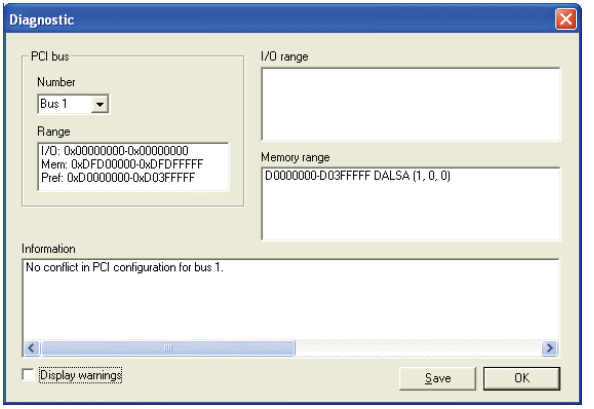
One of the first items to check when there is a problem with any PCI board is to examine the system PCI configuration and ensure that there are no conflicts with other PCI or system devices. The *PCI Diagnostic* program (**cpcidiag.exe**) allows examination of the PCI configuration registers and can save this information to a text file. Run the program via the Windows Start Menu shortcut **Start • Programs • DALSA • 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. 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 Xcelera-CL PX4 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 dump of the PCI configuration registers. Email this file when requested by the DALSA 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 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 Xcelera-CL PX4**.

Device	Description	Type	Started
Corx64Expre4x	X64 Xcelera-CL PX4 messaging	Kernel Driver	Yes
CorLog	Sapera Log viewer	Kernel Driver	Yes
CorMem	Sapera Memory manager	Kernel Driver	Yes
CorPci	Sapera PCI configuration	Kernel Driver	Yes
CorSerial	Sapera Serial Port manager	Kernel Driver	Yes

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

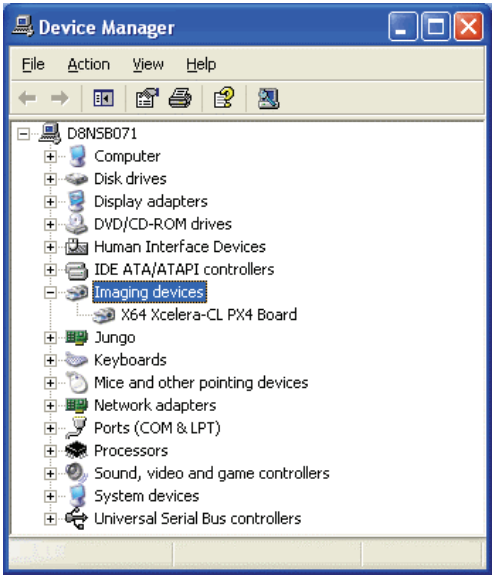
Log Viewer

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

The Log Viewer lists information about the installed DALSA 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 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 Xcelera-CL PX4* 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 Xcelera-CL PX4 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 Xcelera-CL PX4 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 Xcelera-CL PX4 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 6.00 or later:**
  When starting CamExpert, if no DALSA 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 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 before contacting Technical Support.
- Try the board in a different PCIe slot if available.

Symptoms: X64 Xcelera-CL PX4 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 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 before contacting Technical Support.

Symptoms: Card acquisition bandwidth is less than expected

The X64 Xcelera-CL PX4 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. 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.
- Is the X64 Xcelera-CL PX4 installed in a PCI Express x16 slot?
  Note that some computer's x16 slot may only support non x16 boards at x1 or not at all. Check the computer documentation or test an X64 Xcelera-CL PX4 installation. Note that the X64 Xcelera-CL PX4 board is not designed to function at x1 speeds.

CamExpert Quick Start

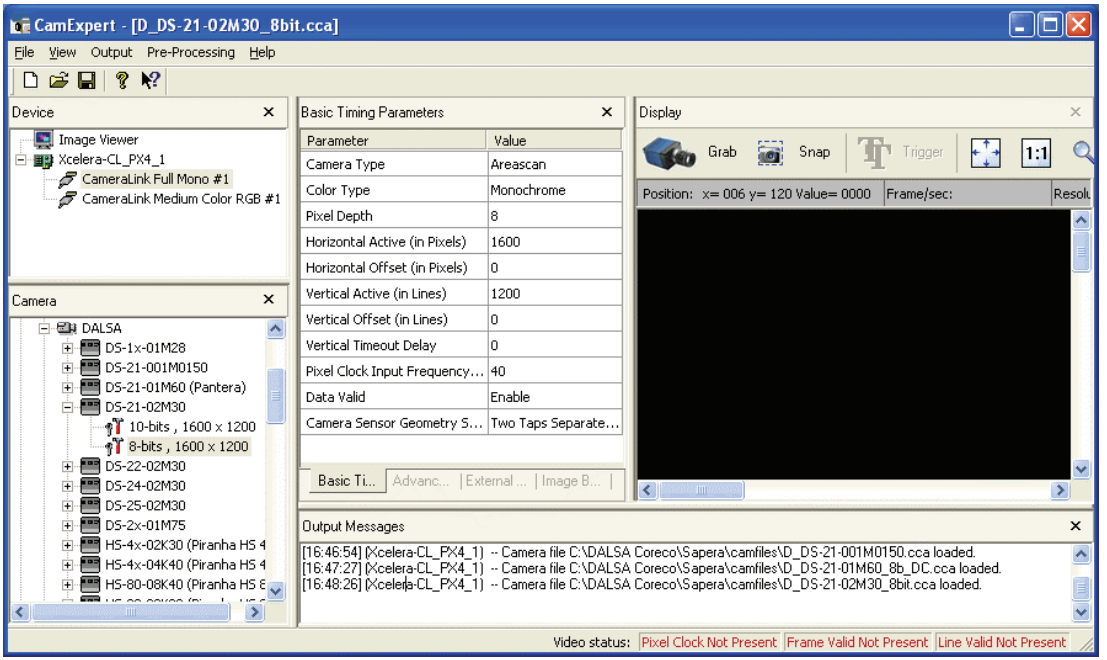
Interfacing Cameras with CamExpert

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 Xcelera-CL PX4 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 Xcelera-CL PX4 Full. The camera outputs monochrome 8-bit video on a 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 Xcelera-CL PX4 was installed with firmware for Full support for monochrome or Medium 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 Xcelera-CL PX4 Full. 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 36), 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 38).

Camera Types & Files Applicable to the X64 Xcelera-CL PX4

The X64 Xcelera-CL PX4 supports digital area scan or linescan cameras using the Camera Link interface standard. See "Camera to Camera Link Connections" on page 20 for information on connecting a Camera Link camera.

Contact DALSA or browse our web site [http://www.imaging.com/camsearch] for the latest information and application notes on X64 Xcelera-CL PX4 supported cameras.

Camera Files Distributed with Sapera

The Sapera distribution CDROM includes camera files for a selection of X64 Xcelera-CL PX4 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 continually updates a camera application library composed of application information and prepared camera files. Along with the camera search utility on the DALSA web site, as described above, a number of camera files are ready to download from the DALSA 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 Xcelera-CL PX4 does not have enough onboard memory for two frame buffers.

The X64 Xcelera-CL PX4 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 Xcelera-CL PX4 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 Xcelera-CL PX4 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, (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 distributes camera files using the ".CCA" extension, (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 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 interfacing a camera from scratch with CamExpert:

- Confirm that DALSA 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 Xcelera-CL PX4 is used. Confirm that the required firmware is loaded into the X64 Xcelera-CL PX4 .
- 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 Xcelera-CL PX4 Flat Field Support

The X64 Xcelera-CL PX4 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 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.

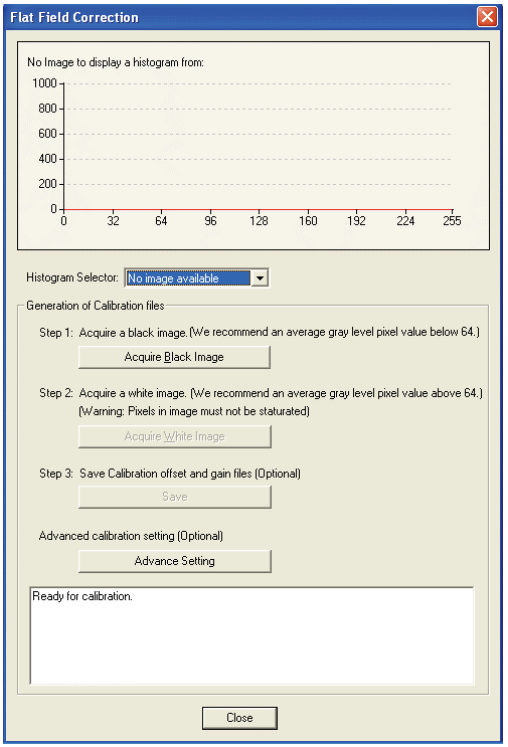
Flat Field Correction Calibration Procedure

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

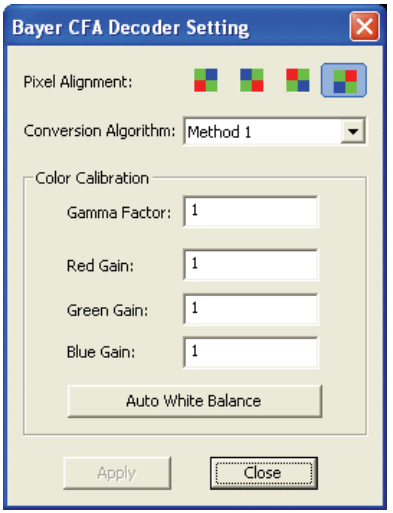
CamExpert supports the use of Bayer Filter cameras by providing a tool to select the Bayer filter mosaic pattern and to perform an auto white balance. Color calibration can then be manually fine tuned with RGB gain and gamma adjustments.

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 Xcelera-CL PX4 (requires loading the Bayer Decoder firmware), 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 processor.

Bayer Filter White Balance Calibration Procedure

The following procedure uses an X64 Xcelera-CL PX4 with hardware Bayer filter support (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 Xcelera-CL PX4 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 •Sapera LT•Demos•Grab Demo**
Program file	**\DALSA \Sapera\Demos\Classes\vc\GrabDemo\Release\GrabDemo.exe**
Workspace	**\DALSA \Sapera\Demos\Classes\vc\SapDemos.dsw**
.NET Solution	**\DALSA \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 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

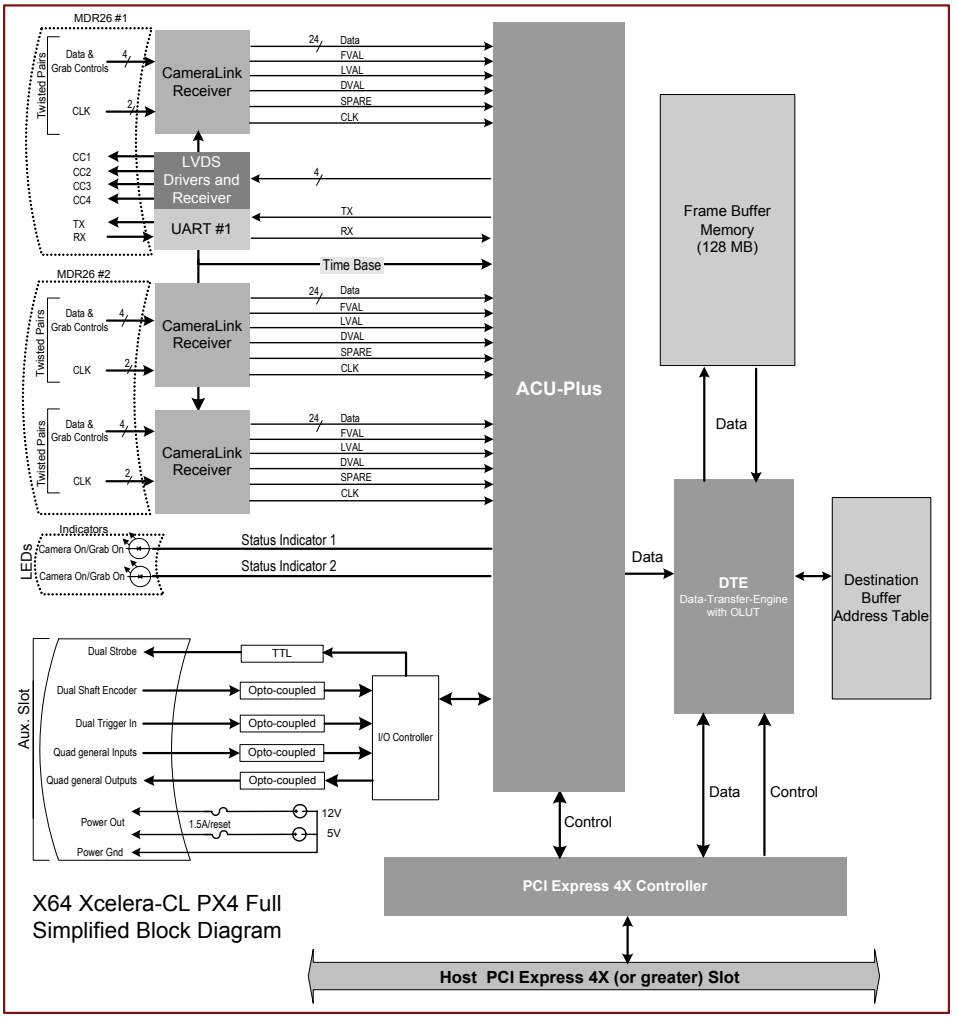
Program	**Start•Programs•Sapera LT•Demos•Sapera++•Flat Field Demo**
Program file	**\\Sapera\Demos\Classes\vc\FlatFieldDemo\Release\FlatfieldDemo.exe**
Workspace	**\\Sapera\Demos\Classes\vc\SapDemos.dsw**
Description	This program demonstrates Flat Field or Flat Line processing, either performed by supporting DALSA 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.

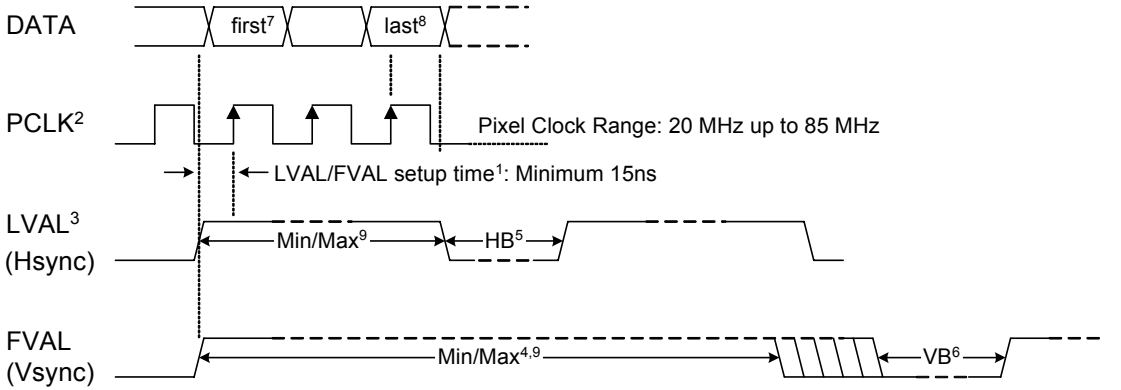
X64 Xcelera-CL PX4 Reference

Full Block Diagram



X64 Xcelera-CL PX4 Full Simplified Block Diagram

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 Xcelera-CL PX4 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 Xcelera-CL PX4 Full. 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 Xcelera-CL PX4 configuration used:

PRM Value	X64 Xcelera-CL PX4 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
	Full - 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
	Full - 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
	Full - 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
	Full - Camera #1	n/a	n/a – use parameter value = 0

See "J4: External Signals Connector " on page **89** 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

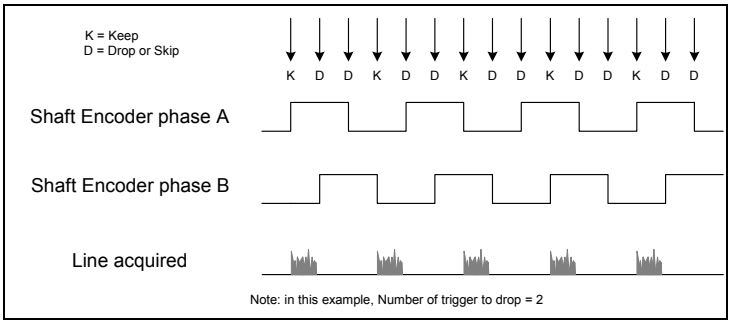
Shaft Encoder Interface Timing

Connector J4, Dual Balanced Shaft Encoder Inputs:

- Input 1: Pin 23 (Phase A +) & Pin 24 (Phase A -)
  (see "J4: External Signals Connector " on page **89** for complete connector signal details)
- Input 2: Pin 25 (Phase B +) & Pin 26 (Phase B -)
- See "External Signals Connector Bracket Assembly" on page **94** for pinout information about the DB37 used for external connections.

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 Xcelera-CL PX4 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.

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.

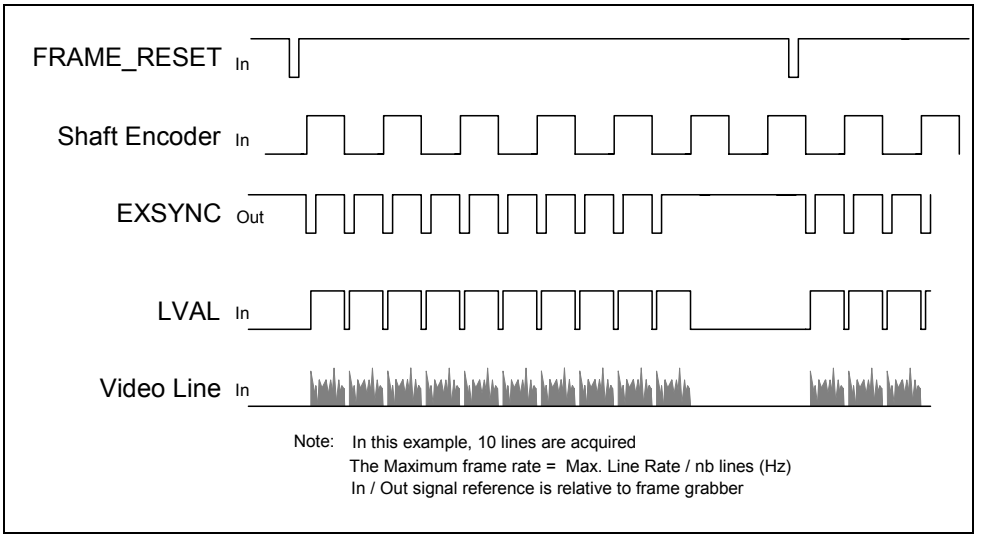
Virtual Frame_Reset Timing Diagram

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 Xcelera-CL PX4 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 Xcelera-CL PX4 via the Trigger In 1 balanced inputs on connector J4 pin 11 (+) and 12 (-).
- After the X64 Xcelera-CL PX4 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 Xcelera-CL PX4 shaft encoder inputs (see "J4: External Signals Connector " on page **89** pinout) or an internal X64 Xcelera-CL PX4 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

Sapera acquisition methods define the control and timing of the camera and frame grabber board. Various methods are available, grouped as:

- Camera Trigger Methods (method 1 and 2 supported)
- Camera Reset Methods (method 1 supported)
- Line Integration Methods (method 1 through 4 supported)
- Time Integration Methods (method 1 through 8 supported)
- Strobe Methods (method 1 through 4 supported)

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

Supported Events

The following acquisition and transfer events are supported. Event monitoring is a major component to the Trigger-to-Image Reliability framework.

Acquisition Events

Acquisition events are related to the acquisition module. They provide feedback on the image capture phase.

- **External Trigger** (Used/Ignored)
  Generated when the external trigger pin is asserted, usually indicating the start of the acquisition process. There are 2 types of external trigger events: 'Used' or 'Ignored'. Following an external trigger, if the event generates a captured image, an External Trigger Used event will be generated (CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER).
  If there is no captured image, an External Trigger Ignored event will be generated (CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED). An external trigger event will be ignored if the rate at which the events are received are higher than the possible frame rate of the camera.
- **Start of Frame**
  Event generated, during acquisition, when the connected sensor video frame start is detected by the board acquisition hardware. The Sapera event value is CORACQ_VAL_EVENT_TYPE_START_OF_FRAME.
- **End of Frame**
  Event generated, during acquisition, when the connected sensor video frame end is detected by the board acquisition hardware. The Sapera event value is CORACQ_VAL_EVENT_TYPE_END_OF_FRAME.
- **Data Overflow**
  The Data Overflow event indicates that there is not enough bandwidth for the acquired data to be transferred without loss. This is usually caused by limitations of the acquisition module and should never occur.
  The Sapera event value is CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW.

- **Frame Valid**
  Event generated when the connected sensor video frame start is detected by the board acquisition hardware. Acquisition does not need to be started, therefore this event can verify a valid signal is connected. The Sapera event value is CORACQ_VAL_EVENT_TYPE_VERTICAL_SYNC.
- **Pixel Clock** (Present/Absent)
  Event generated on the transition from detecting or not detecting a pixel clock signal. The Sapera event values are CORACQ_VAL_EVENT_TYPE_NO_PIXEL_CLK and CORACQ_VAL_EVENT_TYPE_PIXEL_CLK.
- **Frame Lost**
  The Frame Lost event indicates that an acquired image could not be transferred to on-board memory. An example of this case would be if there are no free on-board buffers available for the new image. This may be the case if the image transfer from onboard buffers to host PC memory cannot be sustained due to bus bandwidth issues.
  The Sapera event value is CORACQ_VAL_EVENT_TYPE_FRAME_LOST.
- **Vertical Timeout**
  This event indicates a timeout situation where a camera fails to output a video frame after a trigger. The Sapera event value is CORACQ_VAL_EVENT_VERTICAL_TIMEOUT.

Transfer Events

Transfer events are the ones related to the transfer module. Transfer events provide feedback on image transfer from onboard memory frame buffers to PC memory frame buffers.

- **Start of Frame**
  The Start of Frame event is generated when the first image pixel is transferred from onboard memory into PC memory.
  The Sapera event value is CORXFER_VAL_EVENT_TYPE_START_OF_FRAME.
- **End of Frame**
  The End of Frame event is generated when the last image pixel is transferred from onboard memory into PC memory.
  The Sapera event value is CORXFER_VAL_EVENT_TYPE_END_OF_FRAME.
- **End of Line**
  The End of Line event is generated after a video line is transferred to a PC buffer.
  The Sapera event value is CORXFER_VAL_EVENT_TYPE_END_OF_LINE.
- **End of N Lines**
  The End of N Lines event is generated after a set number of video lines are transferred to a PC buffer. The Sapera event value is CORXFER_VAL_EVENT_TYPE_END_OF_NLINES.
- **End of Transfer**
  The End of Transfer event is generated at the completion of the last image being transferred from onboard memory into PC memory. To complete a transfer, a stop must be issued to the transfer module (if transfers are already in progress). If a transfer of a fixed number of frames was requested, the transfer module will stop transfers automatically. The Sapera event value is CORXFER_VAL_EVENT_TYPE_END_OF_TRANSFER.

LUT Availability

The following table defines the X64 Xcelera-CL PX4 Output LUT availability.

Number of Digital Bits	Number of Taps Full	Number of Taps Dual Base	Output Pixel Format	LUT Format	Notes
8	8	3	MONO 8		Yes
8	8	3	MONO 16	-	Not Supported
8	10	-	MONO 8		Yes
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	1	1	MONO 8		Not Supported
14	1	1	MONO 16		Not Supported
16	1	1	MONO 8		Not Supported
16	1	1	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

Supporting Non-Standard CameraLink Cameras

High performance cameras that output 10 taps can not be interfaced with a standard CameraLink full specification frame grabber. The X64 Xcelera-CL PX4 Full provides support for a non-standard 10-tap format by using specific firmware easily uploaded when required. This format is described below along with an example camera requiring this non-standard format.

Firmware: X64-CL 10-Tap Format 2

- Supports 10-tap Format 2 cameras only such as Basler A504K
- This CameraLink utilization is not compatible with the standard 8 tap full specification
- Output LUT and Flat Field Correction are available
- The following table describes the Bit assignment of Format 2
- Tap 1 Bits are D0_x … Tap 10 Bits are D9_x

Connector 1 Channel Link No. X		Connector 2 Channel Link No. Y		Connector 2 Channel Link No. Z	
Bit Name	Input/Output Pin	Bit Name	Input/Output Pin	Bit Name	Input/Output Pin
D0_0	Tx0/Rx0	D3_2	Tx0/Rx0	D6_5	Tx0/Rx0
D0_1	Tx1/Rx1	D3_3	Tx1/Rx1	D6_6	Tx1/Rx1
D0_2	Tx2/Rx2	D3_4	Tx2/Rx2	D6_7	Tx2/Rx2
D0_3	Tx3/Rx3	D3_5	Tx3/Rx3	D7_0	Tx3/Rx3
D0_4	Tx4/Rx4	D3_6	Tx4/Rx4	D7_1	Tx4/Rx4
D0_5	Tx5/Rx5	D3_7	Tx5/Rx5	D7_2	Tx5/Rx5
D0_6	Tx6/Rx6	D4_0	Tx6/Rx6	D7_3	Tx6/Rx6
D0_7	Tx7/Rx7	D4_1	Tx7/Rx7	D7_4	Tx7/Rx7
D1_0	Tx8/Rx8	D4_2	Tx8/Rx8	D7_5	Tx8/Rx8
D1_1	Tx9/Rx9	D4_3	Tx9/Rx9	D7_6	Tx9/Rx9
D1_2	Tx10/Rx10	D4_4	Tx10/Rx10	D7_7	Tx10/Rx10
D1_3	Tx11/Rx11	D4_5	Tx11/Rx11	D8_0	Tx11/Rx11
D1_4	Tx12/Rx12	D4_6	Tx12/Rx12	D8_1	Tx12/Rx12
D1_5	Tx13/Rx13	D4_7	Tx13/Rx13	D8_2	Tx13/Rx13
D1_6	Tx14/Rx14	D5_0	Tx14/Rx14	D8_3	Tx14/Rx14
D1_7	Tx15/Rx15	D5_1	Tx15/Rx15	D8_4	Tx15/Rx15
D2_0	Tx16/Rx16	D5_2	Tx16/Rx16	D8_5	Tx16/Rx16
D2_1	Tx17/Rx17	D5_3	Tx17/Rx17	D8_6	Tx17/Rx17
D2_2	Tx18/Rx18	D5_4	Tx18/Rx18	D8_7	Tx18/Rx18
D2_3	Tx19/Rx19	D5_5	Tx19/Rx19	D9_0	Tx19/Rx19
D2_4	Tx20/Rx20	D5_6	Tx20/Rx20	D9_1	Tx20/Rx20
D2_5	Tx21/Rx21	D5_7	Tx21/Rx21	D9_2	Tx21/Rx21
D2_6	Tx22/Rx22	D6_0	Tx22/Rx22	D9_3	Tx22/Rx22
D2_7	Tx23/Rx23	D6_1	Tx23/Rx23	D9_4	Tx23/Rx23
LVAL	Tx24/Rx24	D6_2	Tx24/Rx24	D9_5	Tx24/Rx24
FVAL	Tx25/Rx25	D6_3	Tx25/Rx25	D9_6	Tx25/Rx25
D3_0	Tx26/Rx26	D6_4	Tx26/Rx26	D9_7	Tx26/Rx26
D3_1	Tx27/Rx27	LVAL	Tx27/Rx27	LVAL	Tx27/Rx27

X64 Xcelera-CL PX4 Supported Parameters

The tables below describe the Sapera capabilities supported by the X64 Xcelera-CL PX4 Full (i.e. default firmware is loaded). Unless specified, each capability applies to both boards or all mode configurations and all acquisition modes.

The information here is subject to change. Capabilities should be verified by the application because new board driver releases may change product specifications.

Specifically the X64 Xcelera-CL PX4 family is described in Sapera as:

- Board Server: Xcelera-CL_PX4_1
- Acquisition Module: *dependent on firmware used*

Camera Related Capabilities - CameraLink Full Mono

Acq Device (0):

Capability	Values
CORACQ_CAP_CONNECTOR_TYPE	CORACQ_VAL_CONNECTOR_TYPE_CAMLINK (0x2)
CORACQ_CAP_CONNECTOR_CAMLINK (Pin – 01)	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1) CORACQ_VAL_SIGNAL_NAME_PULSE0 (0x8) CORACQ_VAL_SIGNAL_NAME_PULSE1 (0x10) CORACQ_VAL_SIGNAL_NAME_GND (0x4000)
CORACQ_CAP_CONNECTOR_CAMLINK (Pin – 02)	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1) CORACQ_VAL_SIGNAL_NAME_PULSE0 (0x8) CORACQ_VAL_SIGNAL_NAME_PULSE1 (0x10) CORACQ_VAL_SIGNAL_NAME_GND (0x4000)
CORACQ_CAP_CONNECTOR_CAMLINK (Pin – 03)	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1) CORACQ_VAL_SIGNAL_NAME_PULSE0 (0x8) CORACQ_VAL_SIGNAL_NAME_PULSE1 (0x10) CORACQ_VAL_SIGNAL_NAME_GND (0x4000)
CORACQ_CAP_CONNECTOR_CAMLINK (Pin – 04)	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1)

Camera Related Parameters - CameraLink Full Mono

Acq Device (0):

Parameter	Values
CORACQ_PRM_CHANNEL	CORACQ_VAL_CHANNEL_SINGLE (0x1) CORACQ_VAL_CHANNEL_DUAL (0x2)
CORACQ_PRM_FRAME	CORACQ_VAL_FRAME_PROGRESSIVE (0x2)
CORACQ_PRM_INTERFACE	CORACQ_VAL_INTERFACE_DIGITAL (0x2)
CORACQ_PRM_SCAN	CORACQ_VAL_SCAN_AREA (0x1) CORACQ_VAL_SCAN_LINE (0x2)
CORACQ_PRM_SIGNAL	CORACQ_VAL_SIGNAL_DIFFERENTIAL (0x2)
CORACQ_PRM_VIDEO	CORACQ_VAL_VIDEO_MONO (0x1)
CORACQ_PRM_PIXEL_DEPTH	8 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO8 10 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO10 10 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO8 12 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO12 12 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO8 14 bits, # LUT = 0, LUT format = CORDATA_FORMAT_MONO14 16 bits, # LUT = 0, LUT format = CORDATA_FORMAT_MONO16
CORACQ_PRM_VIDEO_STD	CORACQ_VAL_VIDEO_STD_NON_STD (0x1)
CORACQ_PRM_FIELD_ORDER	CORACQ_VAL_FIELD_ORDER_NEXT_FIELD (0x4)
CORACQ_PRM_HACTIVE	min = 1 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_HSYNC	min = 4 pixel max = 4294967295 pixel step = 1 pixel
CORACQ_PRM_VACTIVE	min = 1 line max = 16777215 line step = 1 line
CORACQ_PRM_VSYNC	min = 0 line max = 4294967295 line step = 1 line
CORACQ_PRM_HFRONT_PORCH	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_HBACK_PORCH	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_COUPLING	Not available
CORACQ_PRM_VFRONT_PORCH	min = 0 line max = 0 line step = 1 line
CORACQ_PRM_VBACK_PORCH	min = 0 line max = 0 line step = 1 line
CORACQ_PRM_HFRONT_INVALID	min = 0 pixel max = 16777215 pixel step = 1 pixel

CORACQ_PRM_HBACK_INVALID	min = 0 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_VFRONT_INVALID	min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_VBACK_INVALID	min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_PIXEL_CLK_SRC	CORACQ_VAL_PIXEL_CLK_SRC_EXT (0x2)
CORACQ_PRM_PIXEL_CLK_INT	min = 20000000 Hz max = 85000000 Hz step = 1 Hz
CORACQ_PRM_PIXEL_CLK_11	20000000 Hz
CORACQ_PRM_PIXEL_CLK_EXT	min = 20000000 Hz max = 85000000 Hz step = 1 Hz
CORACQ_PRM_SYNC	CORACQ_VAL_SYNC_SEP_SYNC (0x4)
CORACQ_PRM_HSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_VSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_DETECT_HACTIVE	0 active pixel per line
CORACQ_PRM_DETECT_PIXEL_CLK	Not available
CORACQ_PRM_DETECT_VACTIVE	0 lines per field
CORACQ_PRM_FRAME_INTEGRATE_METHOD	Not available
CORACQ_PRM_FRAME_INTEGRATE_POLARITY	Not available
CORACQ_PRM_TIME_INTEGRATE_METHOD	CORACQ_VAL_TIME_INTEGRATE_METHOD_1 (0x1) CORACQ_VAL_TIME_INTEGRATE_METHOD_2 (0x2) CORACQ_VAL_TIME_INTEGRATE_METHOD_3 (0x4) CORACQ_VAL_TIME_INTEGRATE_METHOD_4 (0x8) CORACQ_VAL_TIME_INTEGRATE_METHOD_5 (0x10) CORACQ_VAL_TIME_INTEGRATE_METHOD_6 (0x20) CORACQ_VAL_TIME_INTEGRATE_METHOD_7 (0x40) CORACQ_VAL_TIME_INTEGRATE_METHOD_8 (0x80)
CORACQ_PRM_CAM_TRIGGER_METHOD	CORACQ_VAL_CAM_TRIGGER_METHOD_1 (0x1) CORACQ_VAL_CAM_TRIGGER_METHOD_2 (0x2)
CORACQ_PRM_CAM_TRIGGER_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_CAM_TRIGGER_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_RESET_METHOD	CORACQ_VAL_CAM_RESET_METHOD_1 (0x1)
CORACQ_PRM_CAM_RESET_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_CAM_RESET_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_NAME	Default Area Scan
CORACQ_PRM_LINE_INTEGRATE_METHOD	CORACQ_VAL_LINE_INTEGRATE_METHOD_1 (0x1) CORACQ_VAL_LINE_INTEGRATE_METHOD_2 (0x2) CORACQ_VAL_LINE_INTEGRATE_METHOD_3 (0x4) CORACQ_VAL_LINE_INTEGRATE_METHOD_4 (0x8)
CORACQ_PRM_LINE_TRIGGER_METHOD	CORACQ_VAL_LINE_TRIGGER_METHOD_1 (0x1)

CORACQ_PRM_LINE_TRIGGER_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_LINE_TRIGGER_DELAY	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_LINE_TRIGGER_DURATION	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_TAPS	min = 1 tap max = 16 tap step = 1 tap
CORACQ_PRM_TAP_OUTPUT	CORACQ_VAL_TAP_OUTPUT_ALTERNATE (0x1) CORACQ_VAL_TAP_OUTPUT_SEGMENTED (0x2) CORACQ_VAL_TAP_OUTPUT_PARALLEL (0x4)
CORACQ_PRM_TAP_1_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_2_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_3_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_4_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_5_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_6_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)

CORACQ_PRM_TAP_7_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_8_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_PIXEL_CLK_DETECTION	CORACQ_VAL_RISING_EDGE (0x4)
CORACQ_PRM_CHANNELS_ORDER	CORACQ_VAL_CHANNELS_ORDER_NORMAL (0x1) CORACQ_VAL_CHANNELS_ORDER_REVERSE (0x2)
CORACQ_PRM_LINESCAN_DIRECTION	Not available
CORACQ_PRM_LINESCAN_DIRECTION_POLARITY	Not available
CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MIN	1 Hz
CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MAX	16777215 Hz
CORACQ_PRM_CAM_TIME_INTEGRATE_DURATION_MIN	1 µs
CORACQ_PRM_CAM_TIME_INTEGRATE_DURATION_MAX	65535000 µs
CORACQ_PRM_CONNECTOR_HD_INPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_VD_INPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_RESET_TRIGGER_INPUT (*)	Default = 0
CORACQ_PRM_TIME_INTEGRATE_PULSE1_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_TIME_INTEGRATE_PULSE1_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_PULSE1_DURATION	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_IO_CONTROL (*)	?
CORACQ_PRM_CONNECTOR_EXPOSURE_INPUT (*)	Default = 0
CORACQ_PRM_TIME_INTEGRATE_PULSE0_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_TIME_INTEGRATE_PULSE0_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_PULSE0_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE1_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_LINE_INTEGRATE_PULSE1_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE1_DURATION	min = 1 µs max = 65535000 µs step = 1 µs

CORACQ_PRM_LINE_INTEGRATE_PULSE0_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_LINE_INTEGRATE_PULSE0_DELAY	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE0_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_VIDEO_LEVEL_MIN	Default = 0 µV
CORACQ_PRM_VIDEO_LEVEL_MAX	Default = 0 µV
CORACQ_PRM_CONNECTOR_LINE_TRIGGER_INPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_LINE_INTEGRATE_INPUT (*)	Connector #1, type 2, pin #1
CORACQ_PRM_CONNECTOR_LINESCAN_DIRECTION_INPUT (*)	Default = 0
CORACQ_PRM_CAMLINK_CONFIGURATION	CORACQ_VAL_CAMLINK_CONFIGURATION_BASE (0x1)
CORACQ_PRM_DATA_VALID_ENABLE	TRUE FALSE
CORACQ_PRM_DATA_VALID_POLARITY	CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_CONNECTOR_PIXEL_CLK_OUTPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_WEN_OUTPUT (*)	Default = 0
CORACQ_PRM_WEN_POLARITY	Not available
CORACQ_PRM_TAP_9_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_10_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_11_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_12_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_13_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)

CORACQ_PRM_TAP_14_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_15_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_16_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TIMESLOT	CORACQ_VAL_TIMESLOT_1 (0x1) CORACQ_VAL_TIMESLOT_2 (0x2)
CORACQ_PRM_BAYER_ALIGNMENT	Not available
CORACQ_PRM_CAM_CONTROL_DURING_READOUT	TRUE FALSE

VIC Related Parameters - CameraLink Full Mono

Acq Device (0):

Parameter	Values
CORACQ_PRM_CAMSEL	CAMSEL_MONO = from 0 to 0 CAMSEL_COLOR not available CAMSEL_YC not available CAMSEL_RGB = from 0 to 0
CORACQ_PRM_PIXEL_MASK	Not available
CORACQ_PRM_DC_REST_MODE	CORACQ_VAL_DC_REST_MODE_AUTO (0x1)
CORACQ_PRM_BRIGHTNESS	Not available
CORACQ_PRM_BRIGHTNESS_RED	Not available
CORACQ_PRM_BRIGHTNESS_GREEN	Not available
CORACQ_PRM_BRIGHTNESS_BLUE	Not available
CORACQ_PRM_CONTRAST	Not available
CORACQ_PRM_CONTRAST_RED	Not available
CORACQ_PRM_CONTRAST_GREEN	Not available
CORACQ_PRM_CONTRAST_BLUE	Not available
CORACQ_PRM_HUE	Not available
CORACQ_PRM_SATURATION	Not available
CORACQ_PRM_FIX_FILTER_ENABLE	Not available
CORACQ_PRM_FIX_FILTER_SELECTOR (*)	Not available

CORACQ_PRM_PROG_FILTER_ENABLE	Not available
CORACQ_PRM_PROG_FILTER_FREQ	Not available
CORACQ_PRM_CROP_LEFT	min = 0 pixel max = 16777215 pixel step = 16 pixel
CORACQ_PRM_CROP_TOP	min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_CROP_WIDTH	min = 16 pixel max = 16777215 pixel step = 16 pixel
CORACQ_PRM_CROP_HEIGHT	min = 1 line max = 16777215 line step = 1 line
CORACQ_PRM_SCALE_HORZ	Not available
CORACQ_PRM_SCALE_VERT	Not available
CORACQ_PRM_SCALE_HORZ_METHOD	Not available
CORACQ_PRM_SCALE_VERT_METHOD	Not available
CORACQ_PRM_DECIMATE_METHOD	CORACQ_VAL_DECIMATE_DISABLE (0x1)
CORACQ_PRM_DECIMATE_COUNT	Default = 0
CORACQ_PRM_LUT_ENABLE	TRUE FALSE
CORACQ_PRM_LUT_NUMBER	Default = 0
CORACQ_PRM_STROBE_ENABLE	TRUE FALSE
CORACQ_PRM_STROBE_METHOD	CORACQ_VAL_STROBE_METHOD_1 (0x1) CORACQ_VAL_STROBE_METHOD_2 (0x2) CORACQ_VAL_STROBE_METHOD_3 (0x4) CORACQ_VAL_STROBE_METHOD_4 (0x8)
CORACQ_PRM_STROBE_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_STROBE_DURATION	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_STROBE_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_FRAME_INTEGRATE_ENABLE	Not available
CORACQ_PRM_FRAME_INTEGRATE_COUNT	Not available
CORACQ_PRM_TIME_INTEGRATE_ENABLE	TRUE FALSE
CORACQ_PRM_TIME_INTEGRATE_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_CAM_RESET_ENABLE	TRUE FALSE
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8 CORACQ_VAL_OUTPUT_FORMAT_MONO16
CORACQ_PRM_EXT_TRIGGER_ENABLE	CORACQ_VAL_EXT_TRIGGER_OFF (0x1)

	CORACQ_VAL_EXT_TRIGGER_ON (0x8)
CORACQ_PRM_VIC_NAME	Default Area Scan
CORACQ_PRM_LUT_MAX	1
CORACQ_PRM_EXT_TRIGGER_DETECTION	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2) CORACQ_VAL_RISING_EDGE (0x4) CORACQ_VAL_FALLING_EDGE (0x8)
CORACQ_PRM_DC_REST_START	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_DC_REST_WIDTH	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_LUT_FORMAT	Default = CORACQ_VAL_OUTPUT_FORMAT_MONO8
CORACQ_PRM_VSYNC_TIMEOUT	Not available
CORACQ_PRM_VSYNC_REF	CORACQ_VAL_SYNC_REF_END (0x2)
CORACQ_PRM_HSYNC_REF	CORACQ_VAL_SYNC_REF_END (0x2)
CORACQ_PRM_LINE_INTEGRATE_ENABLE	TRUE FALSE
CORACQ_PRM_LINE_INTEGRATE_DURATION	min = 1 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_LINE_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_FRAME_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_FRAME_TRIGGER_DETECTION	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2) CORACQ_VAL_RISING_EDGE (0x4) CORACQ_VAL_FALLING_EDGE (0x8) CORACQ_VAL_DOUBLE_PULSE_RISING_EDGE (0x20) CORACQ_VAL_DOUBLE_PULSE_FALLING_EDGE (0x40)
CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_LINE_TRIGGER_DETECTION	CORACQ_VAL_RISING_EDGE (0x4)
CORACQ_PRM_SNAP_COUNT	Default = 1 frame
CORACQ_PRM_INT_LINE_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_INT_LINE_TRIGGER_FREQ	Default = 5000 Hz
CORACQ_PRM_LINESCAN_DIRECTION_OUTPUT (*)	Not available
CORACQ_PRM_BIT_ORDERING	CORACQ_VAL_BIT_ORDERING_STD (0x1)
CORACQ_PRM_EXT_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1) CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_STROBE_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1)
CORACQ_PRM_EXT_FRAME_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1) CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_EXT_LINE_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_INT_LINE_TRIGGER_FREQ_MIN	245 Hz
CORACQ_PRM_INT_LINE_TRIGGER_FREQ_MAX	500000 Hz
CORACQ_PRM_MASTER_MODE	Not available

CORACQ_PRM_MASTER_MODE_HSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_MASTER_MODE_VSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_SHAFT_ENCODER_DROP	min = 0 tick max = 255 tick step = 1 tick
CORACQ_PRM_SHAFT_ENCODER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_TRIGGER_FRAME_COUNT	Default = 1 frame
CORACQ_PRM_INT_FRAME_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_INT_FRAME_TRIGGER_FREQ	min = 1 milli-Hz max = 1073741823 milli-Hz step = 1 milli-Hz
CORACQ_PRM_SHARED_EXT_TRIGGER	Not available
CORACQ_PRM_SHARED_CAM_RESET	Not available
CORACQ_PRM_SHARED_CAM_TRIGGER	Not available
CORACQ_PRM_SHARED_TIME_INTEGRATE	Not available
CORACQ_PRM_SHARED_FRAME_INTEGRATE	Not available
CORACQ_PRM_SHARED_STROBE	Not available
CORACQ_PRM_STROBE_DELAY_2	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_FRAME_LENGTH	CORACQ_VAL_FRAME_LENGTH_FIX (0x1) CORACQ_VAL_FRAME_LENGTH_VARIABLE (0x2)
CORACQ_PRM_FLIP	Not available
CORACQ_PRM_SHARPNESS	min = 0 max = 0 step = 1
CORACQ_PRM_EXT_TRIGGER_DURATION	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_RESET_DELAY	min = 0 µs max = 0 µs step = 1 µs
CORACQ_PRM_CAM_TRIGGER_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_SHAFT_ENCODER_LEVEL	CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_WEN_ENABLE	Not available
CORACQ_PRM_LUT_NENTRIES	256 entries
CORACQ_PRM_EXT_FRAME_TRIGGER_SOURCE (*)	min = 0 max = 5 step = 1
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE (*)	min = 0 max = 7 step = 1
CORACQ_PRM_EXT_TRIGGER_SOURCE (*)	min = 0 max = 5

	step = 1
CORACQ_PRM_SHAFT_ENCODER_MULTIPLY	min = 1 max = 32 step = (2**N)
CORACQ_PRM_PLANAR_INPUT_SOURCES	Not available
CORACQ_PRM_CAM_TRIGGER_DELAY	min = 0 max = 255 step = 1
CORACQ_PRM_EXT_TRIGGER_DELAY_TIME_BASE	CORACQ_VAL_TIME_BASE_LINE (0x4)
CORACQ_PRM_BAYER_DECODER_ENABLE	Not available
CORACQ_PRM_BAYER_DECODER_METHOD	Not available
CORACQ_PRM_BAYER_DECODER_WB_GAIN_RED	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_GAIN_GREEN	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_GAIN_BLUE	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_OFFSET_RED	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_OFFSET_GREEN	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_OFFSET_BLUE	min = 0 max = 0 step = 1
CORACQ_PRM_CAM_CONTROL_PULSE0_HD_ALIGN	Not available
CORACQ_PRM_CAM_CONTROL_PULSE1_HD_ALIGN	Not available
CORACQ_PRM_EXT_TRIGGER_IGNORE_DELAY	min = 0 max = 65535000 step = 1
CORACQ_PRM_CONTROL_SIGNAL_OUTPUT1	Not available
CORACQ_PRM_CONTROL_SIGNAL_OUTPUT2	Not available
CORACQ_PRM_EXT_TRIGGER_SOURCE_STR	[0] = Automatic [1] = From Trigg in #1 [2] = From Trigg in #2 [3] = From Board Sync [4] = To Board Sync [5] = From GIO
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE_STR	[0] = Automatic [1] = From Shaft Encoder Phase A [2] = From Shaft Encoder Phase B [3] = From Shaft Encoder Phase A & B [4] = From Board Sync [5] = To Board Sync [6] = Pulse to Board Sync [7] = To Board Sync When Grabbing
CORACQ_PRM_VERTICAL_TIMEOUT_DELAY	min = 0 max = 16383000 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_FACTOR	min = 0 max = 0

	step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_WEIGHT_RED	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_WEIGHT_GREEN	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_WEIGHT_BLUE	min = 0 max = 0 step = 1

ACQ Related Parameters - CameraLink Full Mono

Acq Device (0):

Parameter	Values
CORACQ_PRM_LABEL	CameraLink Full Mono #1
CORACQ_PRM_EVENT_TYPE	CORACQ_VAL_EVENT_TYPE_START_OF_FRAME CORACQ_VAL_EVENT_TYPE_END_OF_FRAME CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER CORACQ_VAL_EVENT_TYPE_NO_PIXEL_CLK CORACQ_VAL_EVENT_TYPE_PIXEL_CLK CORACQ_VAL_EVENT_TYPE_FRAME_LOST CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED CORACQ_VAL_EVENT_TYPE_VERTICAL_TIMEOUT
CORACQ_PRM_SIGNAL_STATUS	CORACQ_VAL_SIGNAL_HSYNC_PRESENT CORACQ_VAL_SIGNAL_VSYNC_PRESENT CORACQ_VAL_SIGNAL_PIXEL_CLK_PRESENT
CORACQ_PRM_DETECT_PIXEL_CLK	Not available
CORACQ_PRM_DETECT_HACTIVE	0
CORACQ_PRM_DETECT_VACTIVE	0
CORACQ_PRM_FLAT_FIELD_SELECT	0
CORACQ_PRM_FLAT_FIELD_ENABLE	TRUE FALSE
CORACQ_CAP_SERIAL_PORT_INDEX	Supported

Camera Related Capabilities - CameraLink Medium Color RGB

Acquisition Device (1)

Capability	Values
CORACQ_CAP_CONNECTOR_TYPE	CORACQ_VAL_CONNECTOR_TYPE_CAMLINK (0x2)
CORACQ_CAP_CONNECTOR_CAMLINK	
Pin - 01	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1) CORACQ_VAL_SIGNAL_NAME_PULSE0 (0x8) CORACQ_VAL_SIGNAL_NAME_PULSE1 (0x10) CORACQ_VAL_SIGNAL_NAME_GND (0x4000)

Pin - 02	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1) CORACQ_VAL_SIGNAL_NAME_PULSE0 (0x8) CORACQ_VAL_SIGNAL_NAME_PULSE1 (0x10) CORACQ_VAL_SIGNAL_NAME_GND (0x4000)
Pin - 03	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1) CORACQ_VAL_SIGNAL_NAME_PULSE0 (0x8) CORACQ_VAL_SIGNAL_NAME_PULSE1 (0x10) CORACQ_VAL_SIGNAL_NAME_GND (0x4000)
Pin - 04	CORACQ_VAL_SIGNAL_NAME_NO_CONNECT (0x1)

Camera Related Parameters - CameraLink Medium Color RGB

Acquisition Device (1)

Parameter	Values
CORACQ_PRM_CHANNEL	CORACQ_VAL_CHANNEL_SINGLE (0x1)
CORACQ_PRM_FRAME	CORACQ_VAL_FRAME_PROGRESSIVE (0x2)
CORACQ_PRM_INTERFACE	CORACQ_VAL_INTERFACE_DIGITAL (0x2)
CORACQ_PRM_SCAN	CORACQ_VAL_SCAN_AREA (0x1) CORACQ_VAL_SCAN_LINE (0x2)
CORACQ_PRM_SIGNAL	CORACQ_VAL_SIGNAL_DIFFERENTIAL (0x2)
CORACQ_PRM_VIDEO	CORACQ_VAL_VIDEO_RGB (0x8)
CORACQ_PRM_PIXEL_DEPTH	8 bits, # LUT = 1, LUT format = CORDATA_FORMAT_COLORNI8 10 bits, # LUT = 1, LUT format = CORDATA_FORMAT_COLORNI10 12 bits, # LUT = 1, LUT format = CORDATA_FORMAT_COLORNI12
CORACQ_PRM_VIDEO_STD	CORACQ_VAL_VIDEO_STD_NON_STD (0x1)
CORACQ_PRM_FIELD_ORDER	CORACQ_VAL_FIELD_ORDER_NEXT_FIELD (0x4)
CORACQ_PRM_HACTIVE	min = 1 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_HSYNC	min = 4 pixel max = 4294967295 pixel step = 1 pixel
CORACQ_PRM_VACTIVE	min = 1 line max = 16777215 line step = 1 line
CORACQ_PRM_VSYNC	min = 0 line max = 4294967295 line step = 1 line
CORACQ_PRM_HFRONT_PORCH	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_HBACK_PORCH	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_COUPLING	Not available
CORACQ_PRM_VFRONT_PORCH	min = 0 line max = 0 line step = 1 line
CORACQ_PRM_VBACK_PORCH	min = 0 line

	max = 0 line step = 1 line
CORACQ_PRM_HFRONT_INVALID	min = 0 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_HBACK_INVALID	min = 0 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_VFRONT_INVALID	min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_VBACK_INVALID	min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_PIXEL_CLK_SRC	CORACQ_VAL_PIXEL_CLK_SRC_EXT (0x2)
CORACQ_PRM_PIXEL_CLK_INT	min = 20000000 Hz max = 85000000 Hz step = 1 Hz
CORACQ_PRM_PIXEL_CLK_11	20000000 Hz
CORACQ_PRM_PIXEL_CLK_EXT	min = 20000000 Hz max = 85000000 Hz step = 1 Hz
CORACQ_PRM_SYNC	CORACQ_VAL_SYNC_SEP_SYNC (0x4)
CORACQ_PRM_HSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_VSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_DETECT_HACTIVE	0 active pixel per line
CORACQ_PRM_DETECT_PIXEL_CLK	Not available
CORACQ_PRM_DETECT_VACTIVE	0 lines per field
CORACQ_PRM_FRAME_INTEGRATE_METHOD	Not available
CORACQ_PRM_FRAME_INTEGRATE_POLARITY	Not available
CORACQ_PRM_TIME_INTEGRATE_METHOD	CORACQ_VAL_TIME_INTEGRATE_METHOD_1 (0x1) CORACQ_VAL_TIME_INTEGRATE_METHOD_2 (0x2) CORACQ_VAL_TIME_INTEGRATE_METHOD_3 (0x4) CORACQ_VAL_TIME_INTEGRATE_METHOD_4 (0x8) CORACQ_VAL_TIME_INTEGRATE_METHOD_5 (0x10) CORACQ_VAL_TIME_INTEGRATE_METHOD_6 (0x20) CORACQ_VAL_TIME_INTEGRATE_METHOD_7 (0x40) CORACQ_VAL_TIME_INTEGRATE_METHOD_8 (0x80)
CORACQ_PRM_CAM_TRIGGER_METHOD	CORACQ_VAL_CAM_TRIGGER_METHOD_1 (0x1) CORACQ_VAL_CAM_TRIGGER_METHOD_2 (0x2)
CORACQ_PRM_CAM_TRIGGER_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_CAM_TRIGGER_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_RESET_METHOD	CORACQ_VAL_CAM_RESET_METHOD_1 (0x1)
CORACQ_PRM_CAM_RESET_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_CAM_RESET_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_NAME	Default Area Scan

CORACQ_PRM_LINE_INTEGRATE_METHOD	CORACQ_VAL_LINE_INTEGRATE_METHOD_1 (0x1) CORACQ_VAL_LINE_INTEGRATE_METHOD_2 (0x2) CORACQ_VAL_LINE_INTEGRATE_METHOD_3 (0x4) CORACQ_VAL_LINE_INTEGRATE_METHOD_4 (0x8)
CORACQ_PRM_LINE_TRIGGER_METHOD	CORACQ_VAL_LINE_TRIGGER_METHOD_1 (0x1)
CORACQ_PRM_LINE_TRIGGER_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_LINE_TRIGGER_DELAY	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_LINE_TRIGGER_DURATION	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_TAPS	min = 1 tap max = 4 tap step = 1 tap
CORACQ_PRM_TAP_OUTPUT	CORACQ_VAL_TAP_OUTPUT_ALTERNATE (0x1) CORACQ_VAL_TAP_OUTPUT_SEGMENTED (0x2) CORACQ_VAL_TAP_OUTPUT_PARALLEL (0x4)
CORACQ_PRM_TAP_1_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_2_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_3_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_TAP_4_DIRECTION	CORACQ_VAL_TAP_DIRECTION_LR (0x1) CORACQ_VAL_TAP_DIRECTION_RL (0x2) CORACQ_VAL_TAP_DIRECTION_UD (0x4) CORACQ_VAL_TAP_DIRECTION_DU (0x8) CORACQ_VAL_TAP_DIRECTION_FROM_TOP (0x10) CORACQ_VAL_TAP_DIRECTION_FROM_MID (0x20) CORACQ_VAL_TAP_DIRECTION_FROM_BOT (0x40)
CORACQ_PRM_PIXEL_CLK_DETECTION	CORACQ_VAL_RISING_EDGE (0x4)
CORACQ_PRM_CHANNELS_ORDER	CORACQ_VAL_CHANNELS_ORDER_NORMAL (0x1) CORACQ_VAL_CHANNELS_ORDER_REVERSE (0x2)
CORACQ_PRM_LINESCAN_DIRECTION	Not available
CORACQ_PRM_LINESCAN_DIRECTION_POLARITY	Not available
CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MIN	1 Hz
CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MAX	16777215 Hz
CORACQ_PRM_CAM_TIME_INTEGRATE_DURATION_MIN	1 µs
CORACQ_PRM_CAM_TIME_INTEGRATE_DURATION_MAX	65535000 µs

CORACQ_PRM_CONNECTOR_HD_INPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_VD_INPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_RESET_TRIGGER_INPUT (*)	Default = 0
CORACQ_PRM_TIME_INTEGRATE_PULSE1_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_TIME_INTEGRATE_PULSE1_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_PULSE1_DURATION	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_IO_CONTROL (*)	?
CORACQ_PRM_CONNECTOR_EXPOSURE_INPUT (*)	Default = 0
CORACQ_PRM_TIME_INTEGRATE_PULSE0_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_TIME_INTEGRATE_PULSE0_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_PULSE0_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE1_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_LINE_INTEGRATE_PULSE1_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE1_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE0_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_LINE_INTEGRATE_PULSE0_DELAY	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_LINE_INTEGRATE_PULSE0_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_VIDEO_LEVEL_MIN	Default = 0 µV
CORACQ_PRM_VIDEO_LEVEL_MAX	Default = 0 µV
CORACQ_PRM_CONNECTOR_LINE_TRIGGER_INPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_LINE_INTEGRATE_INPUT (*)	Connector #1, type 2, pin #1
CORACQ_PRM_CONNECTOR_LINESCAN_DIRECTION_INPUT (*)	Default = 0
CORACQ_PRM_CAMLINK_CONFIGURATION	CORACQ_VAL_CAMLINK_CONFIGURATION_BASE (0x1)
CORACQ_PRM_DATA_VALID_ENABLE	TRUE FALSE
CORACQ_PRM_DATA_VALID_POLARITY	CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_CONNECTOR_PIXEL_CLK_OUTPUT (*)	Default = 0
CORACQ_PRM_CONNECTOR_WEN_OUTPUT (*)	Default = 0
CORACQ_PRM_WEN_POLARITY	Not available
CORACQ_PRM_TIMESLOT	CORACQ_VAL_TIMESLOT_1 (0x1)

	CORACQ_VAL_TIMESLOT_2 (0x2)
CORACQ_PRM_BAYER_ALIGNMENT	Not available
CORACQ_PRM_CAM_CONTROL_DURING_READOUT	TRUE FALSE

VIC Related Parameters - CameraLink Medium Color RGB

Acquisition Device (1)

Parameter	Values
CORACQ_PRM_CAMSEL	CAMSEL_MONO = from 0 to 0 CAMSEL_COLOR not available CAMSEL_YC not available CAMSEL_RGB = from 0 to 0
CORACQ_PRM_PIXEL_MASK	Not available
CORACQ_PRM_DC_REST_MODE	CORACQ_VAL_DC_REST_MODE_AUTO (0x1)
CORACQ_PRM_BRIGHTNESS	Not available
CORACQ_PRM_BRIGHTNESS_RED	Not available
CORACQ_PRM_BRIGHTNESS_GREEN	Not available
CORACQ_PRM_BRIGHTNESS_BLUE	Not available
CORACQ_PRM_CONTRAST	Not available
CORACQ_PRM_CONTRAST_RED	Not available
CORACQ_PRM_CONTRAST_GREEN	Not available
CORACQ_PRM_CONTRAST_BLUE	Not available
CORACQ_PRM_HUE	Not available
CORACQ_PRM_SATURATION	Not available
CORACQ_PRM_FIX_FILTER_ENABLE	Not available
CORACQ_PRM_FIX_FILTER_SELECTOR (*)	Not available
CORACQ_PRM_PROG_FILTER_ENABLE	Not available
CORACQ_PRM_PROG_FILTER_FREQ	Not available
CORACQ_PRM_CROP_LEFT	min = 0 pixel max = 16777215 pixel step = 16 pixel
CORACQ_PRM_CROP_TOP	min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_CROP_WIDTH	min = 16 pixel max = 16777215 pixel step = 16 pixel
CORACQ_PRM_CROP_HEIGHT	min = 1 line max = 16777215 line step = 1 line
CORACQ_PRM_SCALE_HORZ	Not available
CORACQ_PRM_SCALE_VERT	Not available
CORACQ_PRM_SCALE_HORZ_METHOD	Not available
CORACQ_PRM_SCALE_VERT_METHOD	Not available

CORACQ_PRM_DECIMATE_METHOD	CORACQ_VAL_DECIMATE_DISABLE (0x1)
CORACQ_PRM_DECIMATE_COUNT	Default = 0
CORACQ_PRM_LUT_ENABLE	TRUE FALSE
CORACQ_PRM_LUT_NUMBER	Default = 0
CORACQ_PRM_STROBE_ENABLE	TRUE FALSE
CORACQ_PRM_STROBE_METHOD	CORACQ_VAL_STROBE_METHOD_1 (0x1) CORACQ_VAL_STROBE_METHOD_2 (0x2) CORACQ_VAL_STROBE_METHOD_3 (0x4) CORACQ_VAL_STROBE_METHOD_4 (0x8)
CORACQ_PRM_STROBE_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_STROBE_DURATION	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_STROBE_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_FRAME_INTEGRATE_ENABLE	Not available
CORACQ_PRM_FRAME_INTEGRATE_COUNT	Not available
CORACQ_PRM_TIME_INTEGRATE_ENABLE	TRUE FALSE
CORACQ_PRM_TIME_INTEGRATE_DURATION	min = 1 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_CAM_RESET_ENABLE	TRUE FALSE
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_RGB8888 CORACQ_VAL_OUTPUT_FORMAT_RGB101010 CORACQ_VAL_OUTPUT_FORMAT_RGB16161616
CORACQ_PRM_EXT_TRIGGER_ENABLE	CORACQ_VAL_EXT_TRIGGER_OFF (0x1) CORACQ_VAL_EXT_TRIGGER_ON (0x8)
CORACQ_PRM_VIC_NAME	Default Area Scan
CORACQ_PRM_LUT_MAX	1
CORACQ_PRM_EXT_TRIGGER_DETECTION	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2) CORACQ_VAL_RISING_EDGE (0x4) CORACQ_VAL_FALLING_EDGE (0x8)
CORACQ_PRM_DC_REST_START	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_DC_REST_WIDTH	min = 0 pixel max = 0 pixel step = 1 pixel
CORACQ_PRM_LUT_FORMAT	Default = CORACQ_VAL_OUTPUT_FORMAT_RGB101010
CORACQ_PRM_VSYNC_TIMEOUT	Not available
CORACQ_PRM_VSYNC_REF	CORACQ_VAL_SYNC_REF_END (0x2)
CORACQ_PRM_HSYNC_REF	CORACQ_VAL_SYNC_REF_END (0x2)

CORACQ_PRM_LINE_INTEGRATE_ENABLE	TRUE FALSE
CORACQ_PRM_LINE_INTEGRATE_DURATION	min = 1 pixel max = 16777215 pixel step = 1 pixel
CORACQ_PRM_LINE_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_FRAME_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_FRAME_TRIGGER_DETECTION	CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2) CORACQ_VAL_RISING_EDGE (0x4) CORACQ_VAL_FALLING_EDGE (0x8) CORACQ_VAL_DOUBLE_PULSE_RISING_EDGE (0x20) CORACQ_VAL_DOUBLE_PULSE_FALLING_EDGE (0x40)
CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_LINE_TRIGGER_DETECTION	CORACQ_VAL_RISING_EDGE (0x4)
CORACQ_PRM_SNAP_COUNT	Default = 1 frame
CORACQ_PRM_INT_LINE_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_INT_LINE_TRIGGER_FREQ	Default = 5000 Hz
CORACQ_PRM_LINESCAN_DIRECTION_OUTPUT (*)	Not available
CORACQ_PRM_BIT_ORDERING	CORACQ_VAL_BIT_ORDERING_STD (0x1)
CORACQ_PRM_EXT_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1) CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_STROBE_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1)
CORACQ_PRM_EXT_FRAME_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1) CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_EXT_LINE_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_422 (0x2
CORACQ_PRM_INT_LINE_TRIGGER_FREQ_MIN	245 Hz
CORACQ_PRM_INT_LINE_TRIGGER_FREQ_MAX	500000 Hz
CORACQ_PRM_MASTER_MODE	Not available
CORACQ_PRM_MASTER_MODE_HSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_MASTER_MODE_VSYNC_POLARITY	CORACQ_VAL_ACTIVE_LOW (0x1)
CORACQ_PRM_SHAFT_ENCODER_DROP	min = 0 tick max = 511 tick step = 1 tick
CORACQ_PRM_SHAFT_ENCODER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_TRIGGER_FRAME_COUNT	Default = 1 frame
CORACQ_PRM_INT_FRAME_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_INT_FRAME_TRIGGER_FREQ	min = 1 milli-Hz max = 1073741823 milli-Hz step = 1 milli-Hz
CORACQ_PRM_SHARED_EXT_TRIGGER	Not available
CORACQ_PRM_SHARED_CAM_RESET	Not available
CORACQ_PRM_SHARED_CAM_TRIGGER	Not available
CORACQ_PRM_SHARED_TIME_INTEGRATE	Not available

CORACQ_PRM_SHARED_FRAME_INTEGRATE	Not available
CORACQ_PRM_SHARED_STROBE	Not available
CORACQ_PRM_STROBE_DELAY_2	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_FRAME_LENGTH	CORACQ_VAL_FRAME_LENGTH_FIX (0x1) CORACQ_VAL_FRAME_LENGTH_VARIABLE (0x2)
CORACQ_PRM_FLIP	Not available
CORACQ_PRM_SHARPNESS	min = 0 max = 0 step = 1
CORACQ_PRM_EXT_TRIGGER_DURATION	min = 0 µs max = 65535 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_CAM_RESET_DELAY	min = 0 µs max = 0 µs step = 1 µs
CORACQ_PRM_CAM_TRIGGER_DELAY	min = 0 µs max = 65535000 µs step = 1 µs
CORACQ_PRM_SHAFT_ENCODER_LEVEL	CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_WEN_ENABLE	Not available
CORACQ_PRM_LUT_NENTRIES	256 entries
CORACQ_PRM_EXT_FRAME_TRIGGER_SOURCE (*)	min = 0 max = 5 step = 1
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE (*)	min = 0 max = 7 step = 1
CORACQ_PRM_EXT_TRIGGER_SOURCE (*)	min = 0 max = 5 step = 1
CORACQ_PRM_SHAFT_ENCODER_MULTIPLY	min = 1 max = 32 step = (2**N)
CORACQ_PRM_PLANAR_INPUT_SOURCES	Not available
CORACQ_PRM_CAM_TRIGGER_DELAY	min = 0 max = 255 step = 1
CORACQ_PRM_EXT_TRIGGER_DELAY_TIME_BASE	CORACQ_VAL_TIME_BASE_LINE (0x4)
CORACQ_PRM_BAYER_DECODER_ENABLE	Not available
CORACQ_PRM_BAYER_DECODER_METHOD	Not available
CORACQ_PRM_BAYER_DECODER_WB_GAIN_RED	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_GAIN_GREEN	min = 0 max = 0 step = 1

CORACQ_PRM_BAYER_DECODER_WB_GAIN_BLUE	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_OFFSET_RED	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_OFFSET_GREEN	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_WB_OFFSET_BLUE	min = 0 max = 0 step = 1
CORACQ_PRM_CAM_CONTROL_PULSE0_HD_ALIGN	Not available
CORACQ_PRM_CAM_CONTROL_PULSE1_HD_ALIGN	Not available
CORACQ_PRM_EXT_TRIGGER_IGNORE_DELAY	min = 0 max = 65535000 step = 1
CORACQ_PRM_CONTROL_SIGNAL_OUTPUT1	Not available
CORACQ_PRM_CONTROL_SIGNAL_OUTPUT2	Not available
CORACQ_PRM_EXT_TRIGGER_SOURCE_STR	[0] = Automatic [1] = From Triggin #1 [2] = From Triggin #2 [3] = From Board Sync [4] = To Board Sync [5] = From GIO
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE_STR	[0] = Automatic [1] = From Shaft Encoder Phase A [2] = From Shaft Encoder Phase B [3] = From Shaft Encoder Phase A & B [4] = From Board Sync [5] = To Board Sync [6] = Pulse to Board Sync [7] = To Board Sync When Grabbing
CORACQ_PRM_VERTICAL_TIMEOUT_DELAY	min = 0 max = 16383000 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_FACTOR	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_WEIGHT_RED	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_WEIGHT_GREEN	min = 0 max = 0 step = 1
CORACQ_PRM_BAYER_DECODER_SATURATION_WEIGHT_BLUE	min = 0 max = 0 step = 1

ACQ Related Parameters - CameraLink Medium Color RGB

Acquisition Device (1)

Parameter	Values
CORACQ_PRM_LABEL	CameraLink Medium Color RGB #1
CORACQ_PRM_EVENT_TYPE	CORACQ_VAL_EVENT_TYPE_START_OF_FRAME CORACQ_VAL_EVENT_TYPE_END_OF_FRAME CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER CORACQ_VAL_EVENT_TYPE_NO_PIXEL_CLK CORACQ_VAL_EVENT_TYPE_PIXEL_CLK CORACQ_VAL_EVENT_TYPE_FRAME_LOST CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED CORACQ_VAL_EVENT_TYPE_VERTICAL_TIMEOUT
CORACQ_PRM_SIGNAL_STATUS	CORACQ_VAL_SIGNAL_HSYNC_PRESENT CORACQ_VAL_SIGNAL_VSYNC_PRESENT CORACQ_VAL_SIGNAL_PIXEL_CLK_PRESENT
CORACQ_PRM_DETECT_PIXEL_CLK	Not available
CORACQ_PRM_DETECT_HACTIVE	0
CORACQ_PRM_DETECT_VACTIVE	0
CORACQ_PRM_FLAT_FIELD_SELECT	0
CORACQ_PRM_FLAT_FIELD_ENABLE	TRUE FALSE
CORACQ_CAP_SERIAL_PORT_INDEX	Supported

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 Xcelera-CL PX4 does not have enough onboard memory for two frame buffers.

The X64 Xcelera-CL PX4 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 Xcelera-CL PX4 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 Xcelera-CL PX4 dual configuration equally divides the onboard memory between the two acquisition modules, reducing the available memory for the two buffers by half.

Sapera Servers & Resources

Servers and Resources

The following table describes the X64 Xcelera-CL PX4 Full board

Servers	Resources			
Name	**Type**	**Name**	**Index**	**Description**
Xcelera-CL_PX4_1 (default firmware)	Acquisition	CameraLink Full Mono	0	Full configuration, monochrome output, Camera #1
		CameraLink Full Color RGB #1	1	Full configuration, RGB output, Camera #1
Xcelera-CL_PX4_1 (10 tap firmware)	Acquisition	CameraLink 10 Tap Parallel Mono #1	0	Full configuration, 10 Tap Format 2, Camera #1
Xcelera-CL_PX4_1 (Bayer firmware)	Acquisition	CameraLink Bayer #1	0	Base configuration, Bayer Decoder, Camera #1

The following table describes the X64 Xcelera-CL PX4 Dual board

Servers	Resources			
Name	**Type**	**Name**	**Index**	**Description**
Xcelera-CL_PX4_1 (default Base firmware)	Acquisition	CameraLink Base Mono 1	0	Base configuration, monochrome Camera #1
		CameraLink Base Mono 2	1	Base configuration, monochrome Camera #2
		CameraLink Base RGB 1	2	Base configuration, color RGB Camera #1
		CameraLink Base RGB 2	3	Base configuration, color RGB Camera #2
Xcelera-CL_PX4_1 (Medium firmware)	Acquisition	CameraLink Medium Monochrome 1	0	Medium configuration, monochrome Camera #1
		CameraLink Medium Color RGB 1	1	Medium configuration, RGB Camera #1
Xcelera-CL_PX4_1 (Bayer firmware)	Acquisition	CameraLink Base Bayer 1	0	Base configuration, Bayer Decoder, Camera #1
		CameraLink Base Bayer 2	1	Base configuration, Bayer Decoder, Camera #2

Transfer Resource Locations

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

Source	Transfer passing through	Destination
X64 Xcelera-CL PX4 Acquisition	1 to 2^{17} internal buffers & the X64 internal processor	1 to 2^{17} Host Buffers

Technical Specifications

X64 Xcelera-CL PX4 Board Specifications

X64 Xcelera-CL PX4 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; 1Full or 1 Medium or 1 Base
Common Pixel Formats	CameraLink tap configuration for 8, 10, 12, 14 and 16-bit mono, 24-bit RGB and Bayer.
Tap Format Details	1 Tap – 8/10/12/14/16-bit mono 2 Taps – 8/10/12/14/16-bit mono 4 Taps – 8/10/12-bit mono 4 Taps – 14/16-bit mono (not defined by CameraLink Standard 1.10) 8 taps – 8-bit mono 10 taps – 8-bit mono 3 taps – 8/10/12-bit RGB
Scanning	Area scan and Linescan: Progressive, Multi-Tap, Multi-Channel, Tap reversal, Segmented Tap Configuration, Alternate Tap Configuration
Scanning Directions	Left to Right, Right to Left, Up-Down, Down-Up From Top, From Middle, From Bottom
Resolution *note: these are X64 Xcelera-CL PX4 maximums, not Camera Link specifications*	Horizontal Minimum: 8 Pixels per tap (8-bits/pixel) Horizontal Maximum: 8-bits/pixel x 256K Pixels/line 16-bits/pixel x 128K Pixels/line 32-bits/pixel x 64K Pixels/line 64-bits/pixel x 32K Pixels/line Vertical Minimum: 1 line Vertical Maximum: up to 16,000,000 lines—for area scan sensors infinite line count—for linescan sensors
Pixel Clock Range	20 MHz to 85 MHz as follows: 8-bit: 8 taps @ 85 MHz, any tap configuration 12/14/16-bit: 4 taps @ 85 MHz, any tap configuration
Synchronization Minimums	Horizontal Sync minimum: 4 pixels Vertical Sync minimum: 1 line

Image Buffer	Available with 128 MB
Bandwidth to Host System	Approximately 680MB/s.
Serial Port	Supports communication speeds from 9600 to 115 kbps
Controls	Compliant with DALSA Trigger-to-Image Reliability framework Comprehensive event notifications (see "Supported Events" on page 50) Timing control logic for EXSYNC, PRIN and strobe signals Dual independent opto-coupled external trigger inputs programmable as active high or low (edge or level trigger, where pulse width minimum is 100ns) External trigger latency less than 1 µsec. Dual independent TTL Strobe outputs Quadrature (AB) shaft-encoder inputs for external web synchronization (opto-coupler maximum frequency for any shaft encoder input is 200 KHz) 4 opto-coupled general inputs (5V/24V) 4 opto-coupled general outputs
Processing *Dependant on user loaded firmware configuration*	**Output Lookup Table** one 8-bit in – 8-bit out one 10-bit in – 10-bit out one 12-bit in – 12-bit out three 8-bit in – 8-bit out (RGB) See section "LUT Availability" (page 52) for details. **Bayer Mosaic Filter**: Hardware Bayer Engine supports one 8, 10 or 12-bit Bayer camera input. Bayer output format supports 8 or 10-bit RGB/pixel. Zero host CPU utilization for Bayer conversion. **Flat Field Correction** (Shading Correction): Real-time Flat-line and Flat-field correction. Compensates for sensor defects such as FPN, PRNU, defective pixels and variations between pixels due to the light refraction through a lens (Shading effect). PRNU (*Photo Response Non Uniformity*): PRNU is the variation in response between sensor pixels. 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 Xcelera-CL PX4

- PCI Express x4 slot or x8 slot compatible
- On some computers the X64 Xcelera-CL PX4 installed in a x16 slot may function. The computer documentation or direct testing is required.

Operating System Support

Windows 2000 SP1, Windows XP, XP 64-bit, Windows Vista

Environment

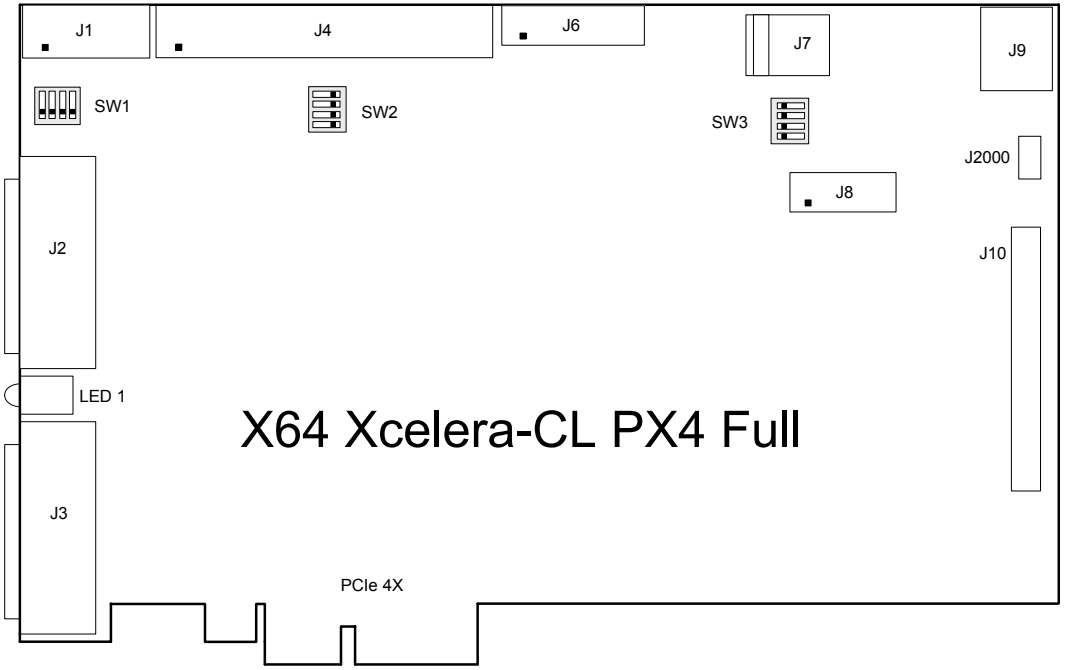
Ambient Temperature:	10° to 50° C (operation) 0° to 70° C (storage)
Relative Humidity:	5% to 90% non-condensing (operating) 0% to 95% (storage)

EMI Certifications

Class B, both FCC and CE

Connector and Switch Locations

X64 Xcelera-CL PX4 Board Layout Drawing



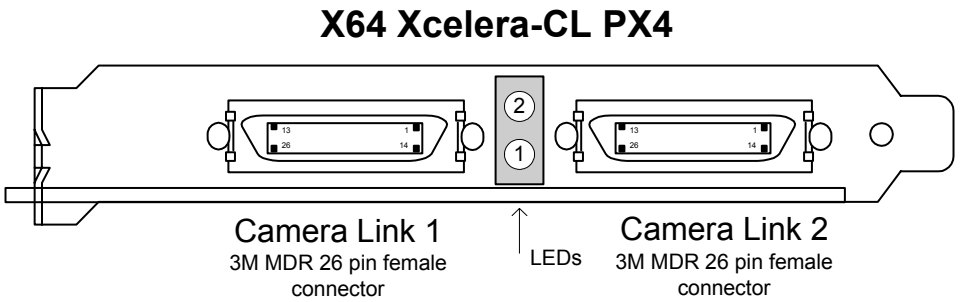
Connector Description List

The following table lists components on the X64 Xcelera-CL PX4 board. Detailed information follows for connectors or switches the end user may have need of.

Connector	Description	Connector	Description
J2	Camera Link Connector	J6	X-I/O Module Interface
J3	Camera Link Connector	J7	PC power to camera interface.
J4	External Signals connector	J1, J8, J9 J10, J2000	Reserved
		SW1, SW2, SW3	Configuration micro-switches

Connector and Switch Specifications

X64 Xcelera-CL PX4 End Bracket Detail



The hardware installation process is completed with the connection of a supported camera to the X64 Xcelera-CL PX4 board using Camera Link cables (see “Camera Link Cables” on page **101**).

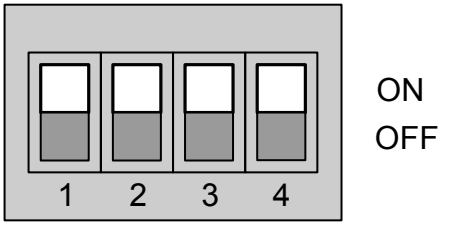
- The X64 Xcelera-CL PX4 board supports a camera with one or two Camera Link MDR-26 connectors (two Base or one Medium – see “Data Port Summary” on page 100 for information on Camera Link configurations).
- Connect the camera to the J1 connector with a Camera Link cable. When using a Medium camera, connect the second camera connector to J2.

Note: If the camera is powered by the X64 Xcelera-CL PX4, refer to "External Signals Connector Bracket Assembly" on page 94 for power connections.

Contact DALSA or browse our web site http://www.imaging.com/camsearch for the latest information on X64 Xcelera-CL PX4 supported cameras.

Configuration Micro-switches

Three sets of 4 switches are used for user configurations not controlled by software. The following figure is a typical view of each switch set, shown with the individual switch set in the OFF position. Following the figure, each of the three switch sets is described. Refer to the board component layout for their positions.



SW1, SW2, SW3 Component View

SW1: General Inputs Signal Switch Point

For each general input, select the threshold voltage detected as a logic high signal. See "Note 1: General Inputs Specifications" on page 90.

SW1 Switch Number	Assigned to	OFF Position	ON Position (default)
1	general input 1	Logic Transition at ~2 volts (preferred for differential signals)	Logic Transition at ~10 volts
2	general input 2		
3	general input 3		
4	general input 4		

SW2: Trigger Inputs Signal Switch Point

For each trigger input, select the threshold voltage detected as a logic high signal. See "Note 3: External Trigger Input Specifications" on page 91.

SW2 Switch Number	Assigned to	OFF Position (default)	ON Position
1	trigger input 1	Logic Transition at ~2 volts (preferred for differential signals)	Logic Transition at ~10 volts
2	trigger input 2		
3	NA		
4	NA		

SW3: Additional Controls

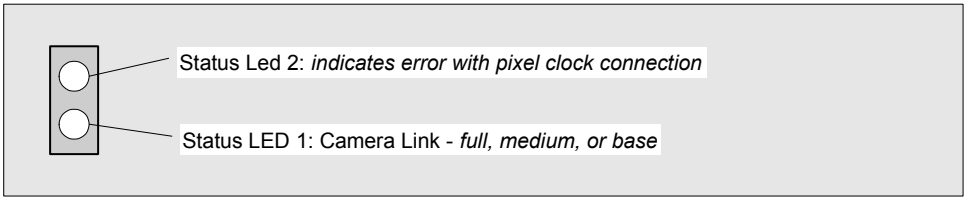
The X64 Xcelera-CL PX4 powers up either in its normal state or a 'Safe Boot' mode required to load firmware under certain conditions. See the notes for SW3-1 following the table for details.

SW3 Switch Number	Assigned to	OFF Position (default)	ON Position
1	Boot Mode	Normal	Safe
2	reserved		
3	reserved		
4	reserved		

SW3-1 Details

- Normal Mode: Board powers up in the normal operating mode.
- Safe Mode: With the computer off, move the switch to the ON position. This mode is required if any problems occurred while updating firmware. With the switch in the ON position, power on the computer and update the firmware again. When the update is complete, power off the computer and move the switch to the OFF position. Power on the computer once again for normal operation. (See "Recovering from a Firmware Update Error" on page 22).

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** (~2 Hz): Camera Line Valid signal detected.
- **Fast Flashing Green** (~16 Hz): Acquisition in progress.
- ***Status LED 2 flashing red***: Camera pixel clock incorrectly connected to J3 instead of J2. (Example - a Base camera is incorrectly connected to J3).

J2: 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

J3: Camera Link Connector 2

Medium and Full Camera Link sources require cables connected to both J2 and J3.

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
FULL_X0-	19	Input	Neg. Full Data 0
FULL _X0+	6	Input	Pos. Full Data 0
FULL _X1-	18	Input	Neg. Full Data 1
FULL _X1+	5	Input	Pos. Full Data 1
FULL _X2-	17	Input	Neg. Full Data 2
FULL _X2+	4	Input	Pos. Full Data 2
FULL _X3-	15	Input	Neg. Full Data 3
FULL _X3+	2	Input	Pos. Full Data 3
FULL _XCLK-	16	Input	Neg. Full Clock
FULL _XCLK+	3	Input	Pos. Full Clock
GND	1, 13, 14, 26		Ground

Camera Link Camera Control Signal Overview

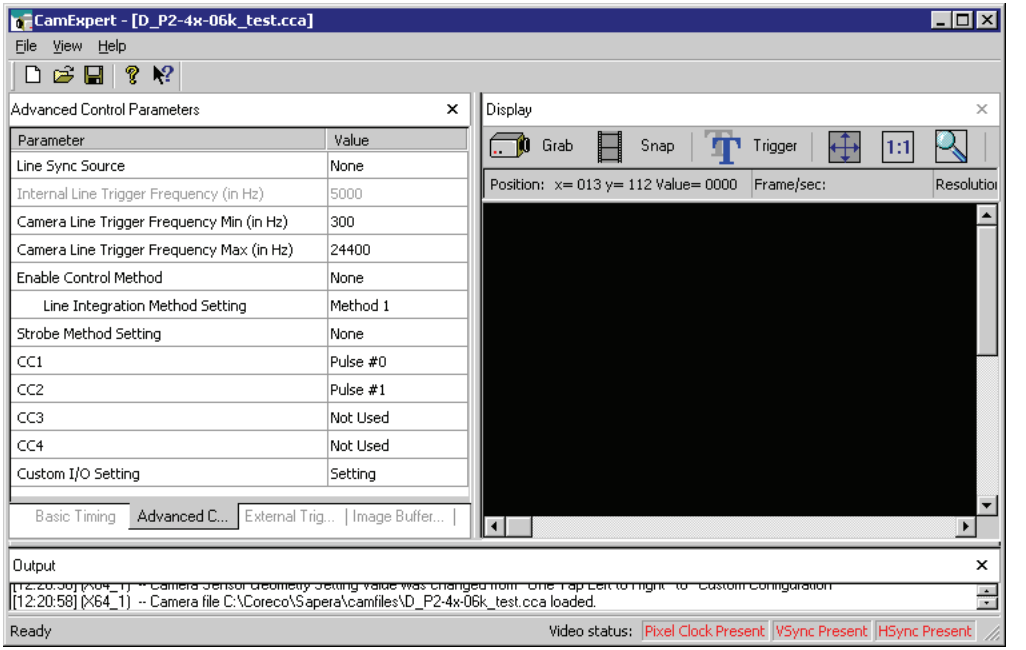
Four LVDS pairs are for general-purpose camera control, defined as camera inputs / frame grabber outputs by the Camera Link Base camera specification. These controls are on J1 and also on J2 for the second Base camera input of the X64 Xcelera-CL PX4 in two Base configuration.

- 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 Xcelera-CL PX4 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 Xcelera-CL PX4 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

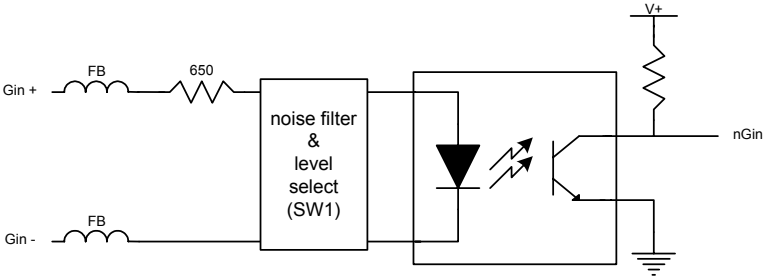
2	4	...	38	40
1	3	...	37	39

J4 Signal Descriptions

Description	Pin #	Pin #	Description
Ground	1	2	Ground
General Input 0 + (see note 1)	3	4	General Input 0 -
General Input 1 +	5	6	General Input 1 -
General Input 2 +	7	8	General Input 2 -
General Input 3 +	9	10	General Input 3 -
General Output 0 + (see note 2)	11	12	General Output 0 -
General Output 1 +	13	14	General Output 1 -
General Output 2 +	15	16	General Output 2 -
General Output 3 +	17	18	General Output 3 -
External Trigger Input 0 + (see note 3)	19	20	External Trigger Input 0 -
External Trigger Input 1 +	21	22	External Trigger Input 1 -
Shaft Encoder Phase A + (see note 4)	23	24	Shaft Encoder Phase A -
Shaft Encoder Phase B +	25	26	Shaft Encoder Phase B -
Ground	27	28	Strobe Output 0 (see note 5)
Ground	29	30	Strobe Output 1
Ground	31	32	Ground
Power Output 5 Volts, 1.5A max (see note 6)	33	34	Power Output 5 Volts, 1.5A max
Power Output 12 Volts, 1.5A max	35	36	Power Output 12 Volts, 1.5A max
Ground	37	38	Ground
Ground	39	40	Ground

Note 1: General Inputs Specifications

Each of the four General Inputs are opto-coupled and able to connect to differential signals (LVDS or RS422) or single ended TTL source signals. These inputs generate individual interrupts and are read by the Sapera application. The following figure is typical for each Genera Input.

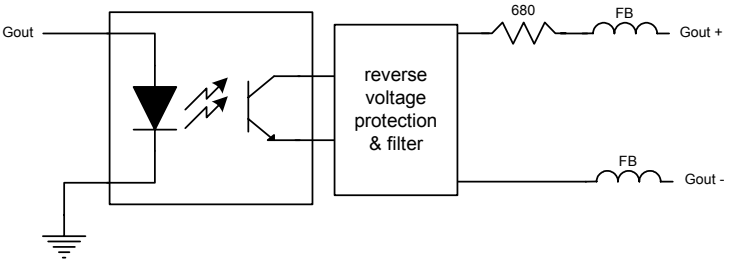


Input Details:

- For single ended TTL signals, the Gin- pin is connected to ground. The switch point is ~10V by default and can be change to ~2V with **SW1**.
- Each input has a ferrite bead plus a 650 ohm series resistor on the opto-coupler anode.
- Each input provides some high frequency noise filtering.
- Maximum input signal frequency is 25 KHz.

Note 2: General Outputs Specifications

Each of the four General Outputs are opto-coupled. Each output is an isolated open-collector NPN transistor switch. The following figure is typical for each General Output.

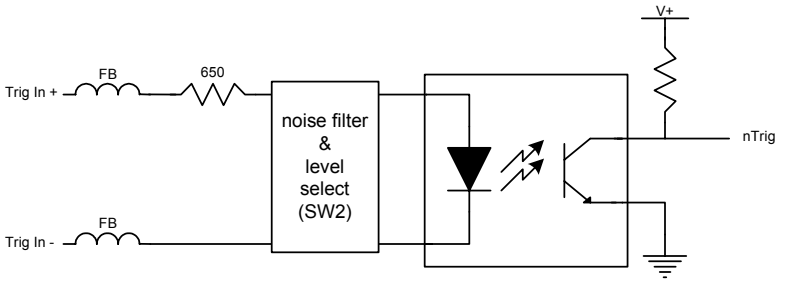


Output Details:

- Each output has ferrite beads plus a 680 ohm series resistor on the cathode (+) connection.
- Maximum output device differential voltage is 25V.
- Maximum output device sink current is 35mA with 25V output differential.
- Maximum reverse voltage is 25V.
- Maximum output switching frequency is limited by driver and register access on the PCIe bus.

Note 3: External Trigger Input Specifications

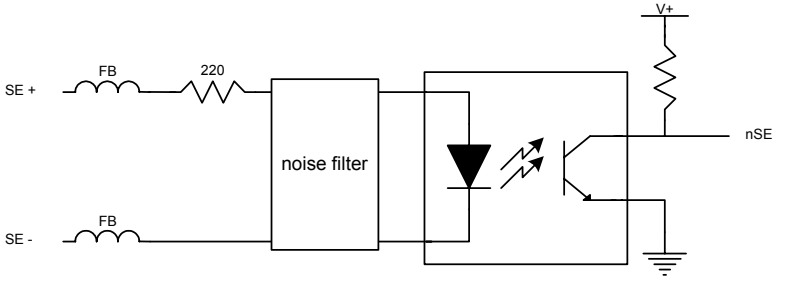
The two Trigger Inputs are opto-coupled and compatible to differential signals (LVDS or RS422) or single ended TTL source signals. The following figure is typical for each External Trigger Input.



- For single ended TTL signals, the TrigIn- pin is connected to ground. The switch point is ~10V by default and can be change to ~2V with **SW2**.
- The incoming trigger pulse is software "debounced" to ensure that no voltage glitch is detected as a valid trigger pulse. This debounce circuit time constant can be programmed from 0ms to 255ms. Any pulse smaller than the programmed value is blocked and therefore not seen by the acquisition circuitry.
- Each input has a ferrite bead plus a 650 ohm series resistor on the opto-coupler anode.
- Maximum input signal frequency is 200 KHz.
- Refer to Sapera parameters:
  CORACQ_PRM_EXT_TRIGGER_SOURCE
  CORACQ_PRM_EXT_TRIGGER_ENABLE
  CORACQ_PRM_EXT_TRIGGER_LEVEL
  CORACQ_PRM_EXT_FRAME_TRIGGER_LEVEL
  CORACQ_PRM_EXT_TRIGGER_DETECTION
  CORACQ_PRM_EXT_TRIGGER_DURATION
- See also *.cvi file entries:
  External Trigger Level, External Frame Trigger Level, External Trigger Enable, External Trigger Detection.
- External Trigger Input 2 used for two pulse external trigger with variable frame length linescan acquisition.

Note 4: Shaft Encoder Input Specifications

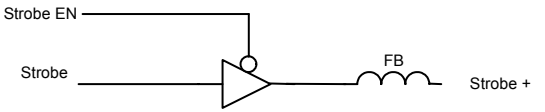
Dual Quadrature Shaft Encoder Inputs (phase A and phase B) are opto-coupled and able to connect to differential signals (LVDS or RS422) or single ended TTL source signals. The following figure is typical for each input.



- For single ended TTL signals, the SE- pin is connected to ground. The switch point is ~2V.
- Each input has a ferrite bead plus a 220 ohm series resistor on the opto-coupler anode.
- Maximum input signal frequency is 200 KHz.
- See "Line Trigger Source Selection for Linescan Applications" on page 45 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.

Note 5: Strobe Output Specifications

Dual TTL Strobe outputs are provided. The following figure is typical for each strobe out.



- Each strobe output is a tri-state driver, enabled by software.
- Each strobe output is 5V TTL level.
- Each output has a ferrite bead.
- Maximum source current is 32mA typical.
- Maximum sink current is 32mA typical.
- Output switching is < 4.2ns typical.
- Refer to Sapera Strobe Methods 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.

Note 6: DC Power Details

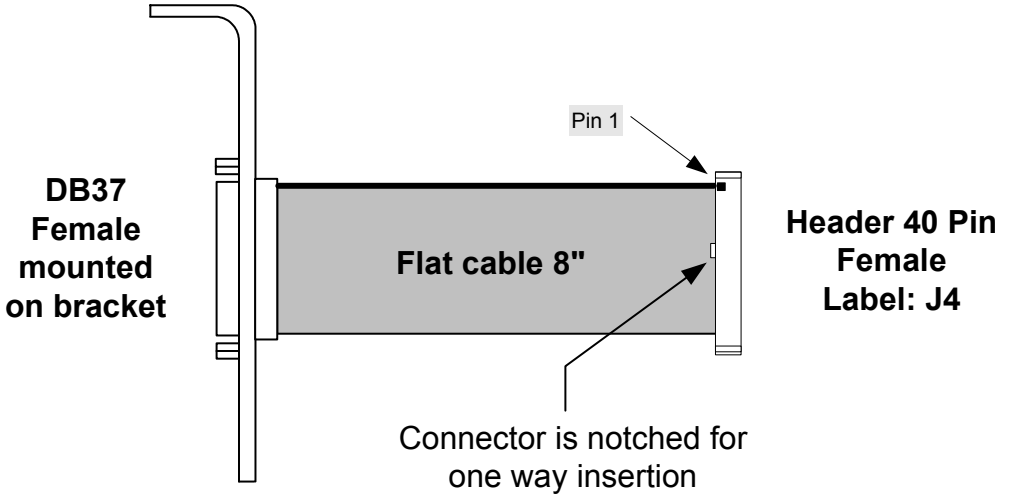
- Connect the PC floppy drive power connector to J7 so as to supply DC power to a camera. Both 5Vdc and 12Vdc are available on J4 or on the DB37 External Signals Bracket Assembly.
- Both the 5Volt and 12Volt power pins have a 1.5 amp re-settable fuse on the board. If the fuse is tripped, turn off the host computer power. When the computer is turned on again, the fuse is automatically reset.

External Signals Connector Bracket Assembly (Type 1)

The External Signals bracket (OC-X4CC-IOCAB) provides a simple way to bring out the signals from the External Signals Connector J4 to a bracket mounted DB37. Install the bracket assembly into an adjacent PC expansion slot and connect the free cable end to the board's J4 header. When connecting to J4, make sure that the cable pin 1 goes to J4 pin 1 (see the layout drawing "X64 Xcelera-CL PX4 Board Layout Drawing" on page 82).

Note: For additional independent I/O signals use the optional X-I/O module. See "Appendix: X-I/O Module Option" on page 103.

External Signals Connector Bracket Assembly (Type 1) Drawing



External Signals Connector Bracket Assembly (Type 1) Pinout

The following table defines the signal pinout on the DB37 connector. Refer to the table "J4: External Signals Connector " on page 89 for signal descriptions.

DB37 Pin Number	Signal	J4 Connector Pin Number
1	Ground	1
20	Ground	2
2	General Input 0 +	3
21	General Input 0 -	4
3	General Input 1 +	5
22	General Input 1 -	6
4	General Input 2 +	7
23	General Input 2 -	8
5	General Input 3 +	9
24	General Input 3 -	10
6	General Output 0 +	11
25	General Output 0 -	12
7	General Output 1 +	13
26	General Output 1 -	14
8	General Output 2 +	15
27	General Output 2 -	16
9	General Output 3 +	17
28	General Output 3 -	18
10	External Trigger Input 0 +	19
29	External Trigger Input 0 -	20
11	External Trigger Input 1 +	21
30	External Trigger Input 1 -	22
12	Shaft Encoder Phase A +	23
31	Shaft Encoder Phase A -	24
13	Shaft Encoder Phase B +	25
32	Shaft Encoder Phase B -	26
14	Ground	27
33	Strobe Output 0	28
15	Ground	29
34	Strobe Output 1	30

16	Ground	31
35	Ground	32
17	+5V	33
36	+5V	34
18	+12V	35
37	+12V	36
19	Ground	37
—	—	38
—	—	39
—	—	40

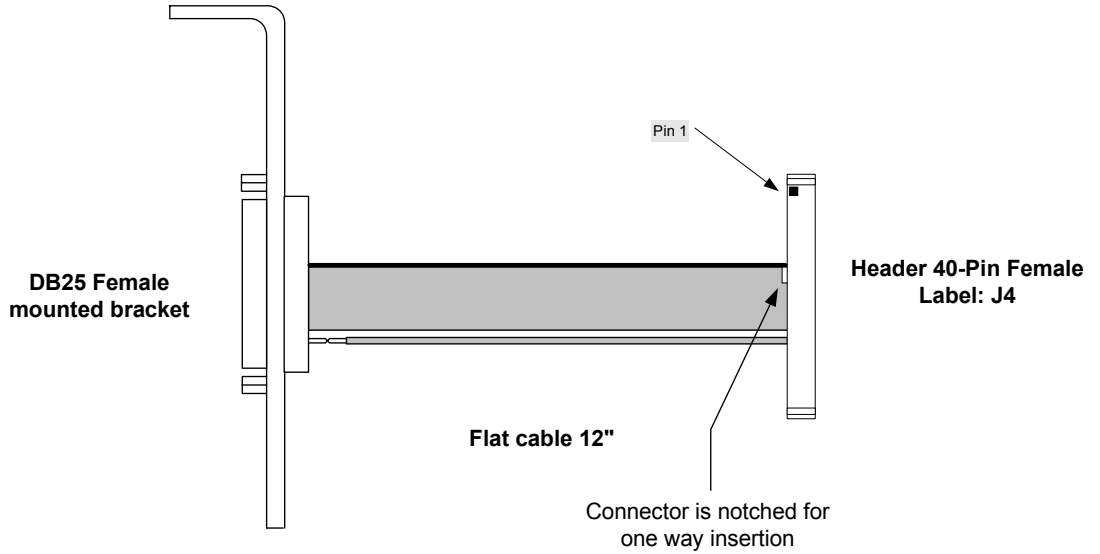
External Signals Connector Bracket Assembly (Type 2)

The External Signals bracket (OR-X4CC-0TIO2) provides a simple way to bring out the signals from the External Signals Connector J4 to a bracket mounted DB25. External cables designed for the DALSA X64-CL iPro can be connected directly.

Install the bracket assembly into an adjacent PC expansion slot and connect the free cable end to the board's J4 header. When connecting to J4, make sure that the cable pin 1 goes to J4 pin 1 (see the layout drawing "X64 Xcelera-CL PX4 Board Layout Drawing" on page 82).

Note: For additional independent I/O signals use the optional X-I/O module. See "Appendix: X-I/O Module Option" on page 103.

External Signals Connector Bracket Assembly (Type 2) Drawing



External Signals Connector Bracket Assembly (Type 2) Pinout

The following table defines the signal pinout on the DB25 connector.
Refer to the table "J4: External Signals Connector " on page 89 for signal descriptions.

DB25 Pin Number	Signal	J4 Connector Pin Number
6	TRIG_0+	19
19	TRIG_0-	20
7	TRIG_1+	21
20	TRIG_1-	22
8	PHY_0+	23
21	PHY_0-	24
9	PHY_1+	25
22	PHY_1-	26
11	Strobe_0	28
24	GND	29
10	Strobe_1	30
14	GND	31
15	GND	38
16	GND	39
25	GND	40

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:

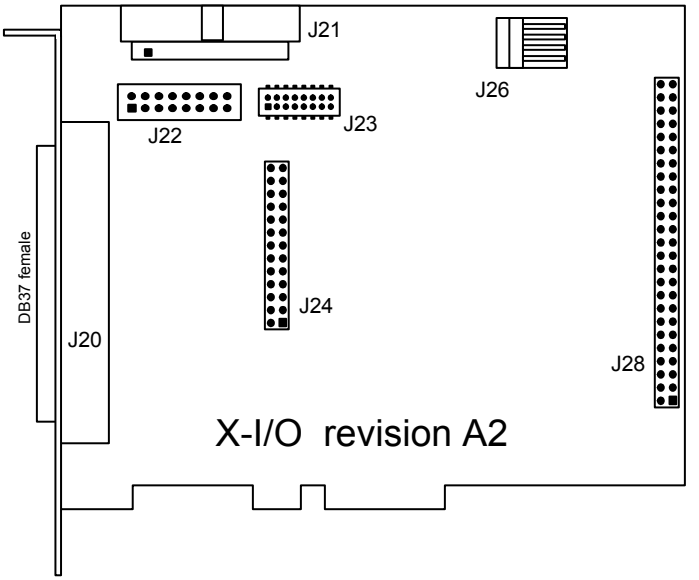
3 M	http://www.3m.com/us/electronics_mfg/interconnects/ *(enter Camera Link as the search keyword)*
Nortech Systems	http://www.nortechsys.com/intercon/CameraLinkMain.htm

Appendix: X-I/O Module Option

X-I/O Module Overview

- The X-I/O module requires X64 Xcelera-CL PX4 board driver version 1.00 (or later) and Sapera LT version 5.30 (or later).
- Occupies an adjacent slot to the X64 Xcelera-CL PX4 Full. Slot can be either PCI-32 or PCI-64—no PCI signals or power are used.
- Connects to the X64 Xcelera-CL PX4 via a 16 pin flat ribbon cable. J23 on X-I/O to J6 on X64 Xcelera-CL PX4.
- The X-I/O signals supplement the external signal I/O available on the main board. The two sets of I/O are independent of each other.
- 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 107 and "TTL Output in PNP Mode: Electrical Details" on page 108.
- X-I/O provides 2 opto-coupled inputs. See "Opto-coupled Input: Electrical Details" on page 109.
- X-I/O provides 6 TTL level inputs with software selectable transition point. See "TTL Input Electrical Details" on page 109.
- 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.
J23	16 pin header connector (interconnect to the X64 Xcelera-CL PX4 via supplied ribbon cable).
J21, J22, 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 Xcelera-CL PX4 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 Xcelera-CL PX4 board.
- 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 16 pin ribbon cable from J23 to the X64 Xcelera-CL PX4 board J6.
- Power on the computer again.
- For new X64 Xcelera-CL PX4 and X-I/O module installations, simply follow the procedure to install Sapera and the X64 Xcelera-CL PX4 driver (start with "Sapera LT Library Installation" on page 12).

X64 Xcelera-CL PX4 and X-I/O Driver Update

- If both Sapera and X64 Xcelera-CL PX4 driver need to be installed, follow the procedure "Sapera and Board Driver Upgrades" on page 12. 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 Xcelera-CL PX4 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 15.

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.

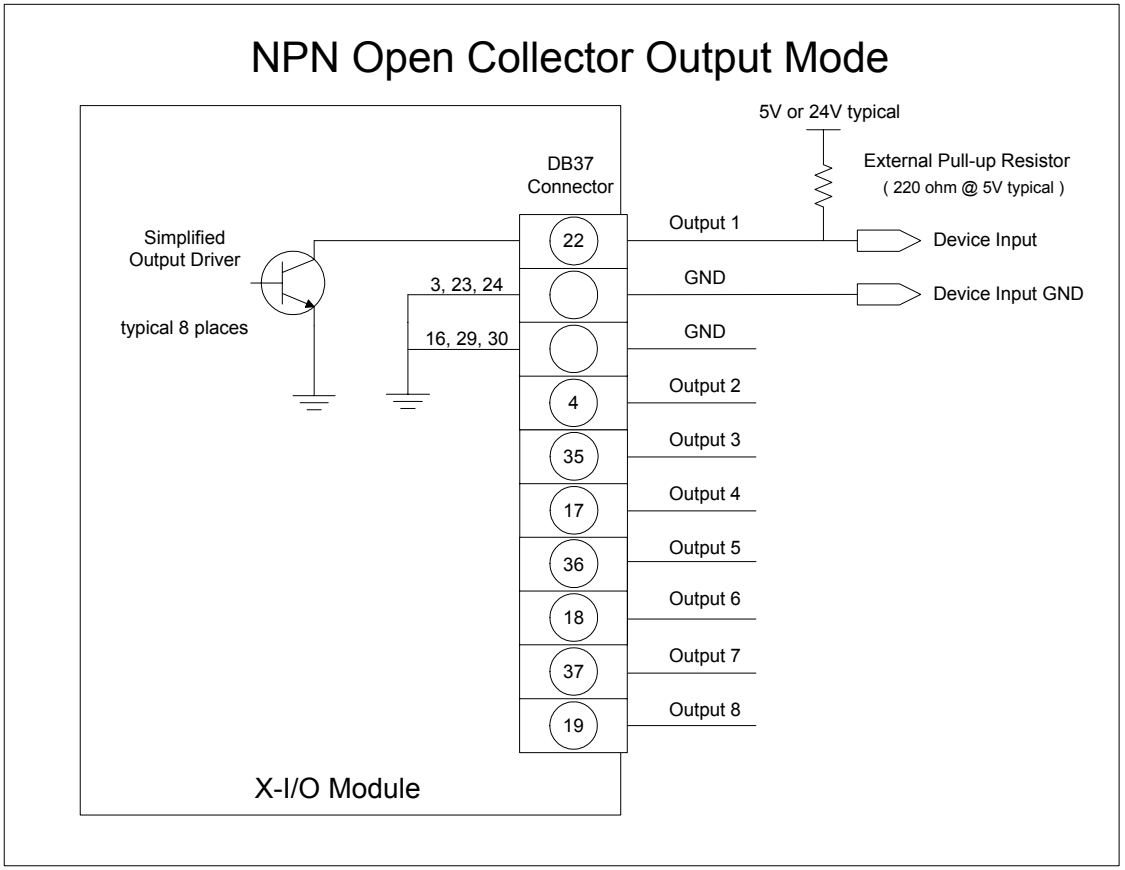
For the external signals Trigger Input, Shaft Encoder Input, and Strobe output, now available on the DB37, refer "J4: External Signals Connector" on page 89 to for signal details.

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+
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

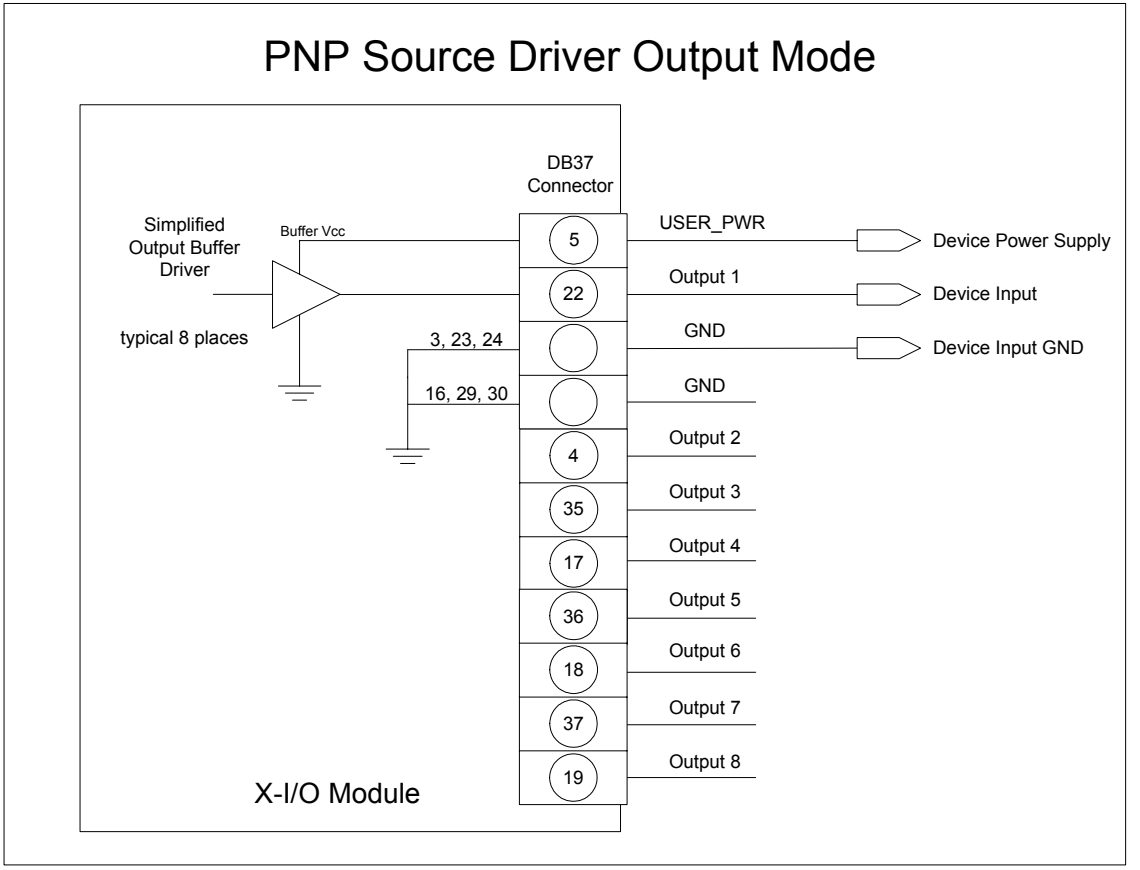
When the TTL outputs are configured for NPN mode (open collector - sink mode) the user is required to provide an external input pull-up resistor on the signal being controlled by the X-I/O output. A simplified schematic and important output specifications follow:



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

TTL Output in PNP Mode: Electrical Details

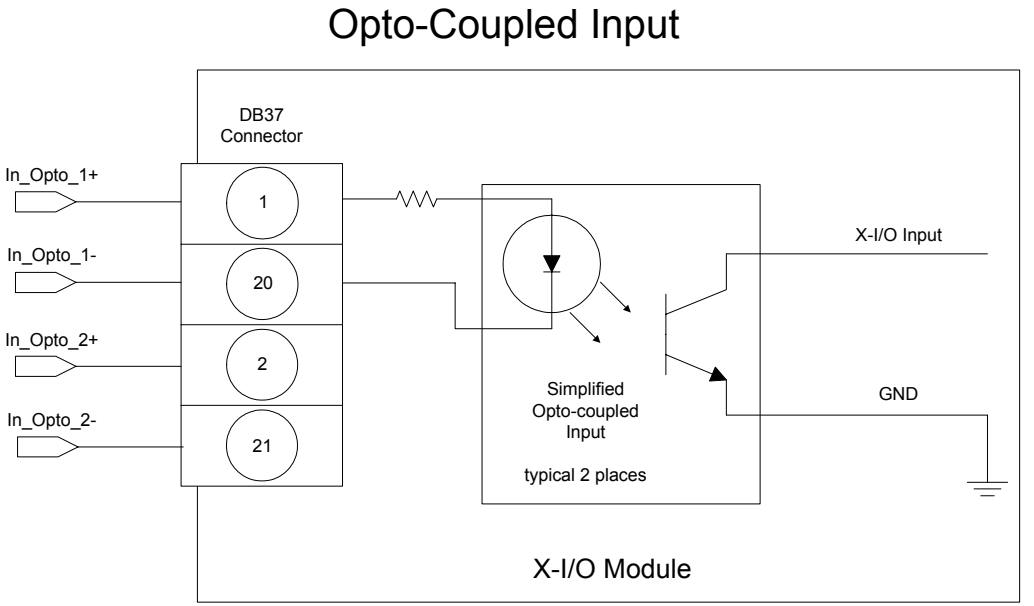
When the TTL outputs are configured for PNP mode (source driver) an external power supply is required to provide the buffer output supply voltage (USER_PWR). A simplified schematic and important output specifications follow:



- User provides the output power supply voltage (7 volts to 35 volts).
- Maximum source driver output current is 350 mA.
- 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 110) 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

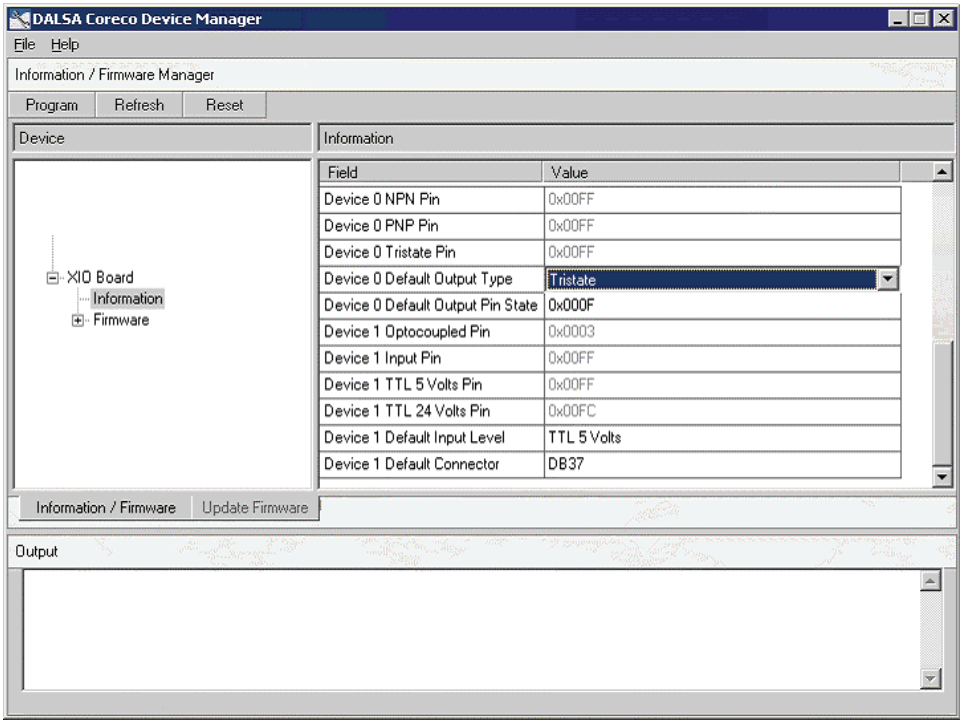
Sapera version 5.30 (or later) provides support for the X-I/O module via an I/O class and demonstration program. Users can use the demonstration program as is, or use the demo program source code to implement X-I/O controls within the custom imaging application.

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 Device Manager program. Run the program via the windows start menu: (**Start • Programs • DALSA • X64 Xcelera-CL PX4 Device Driver • Device Manager**).

The Device Manager provides information on the installed X64 Xcelera-CL PX4 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 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 • 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 Xcelera-CL PX4 board is selected and click OK to continue.

General I/O Module Control Panel

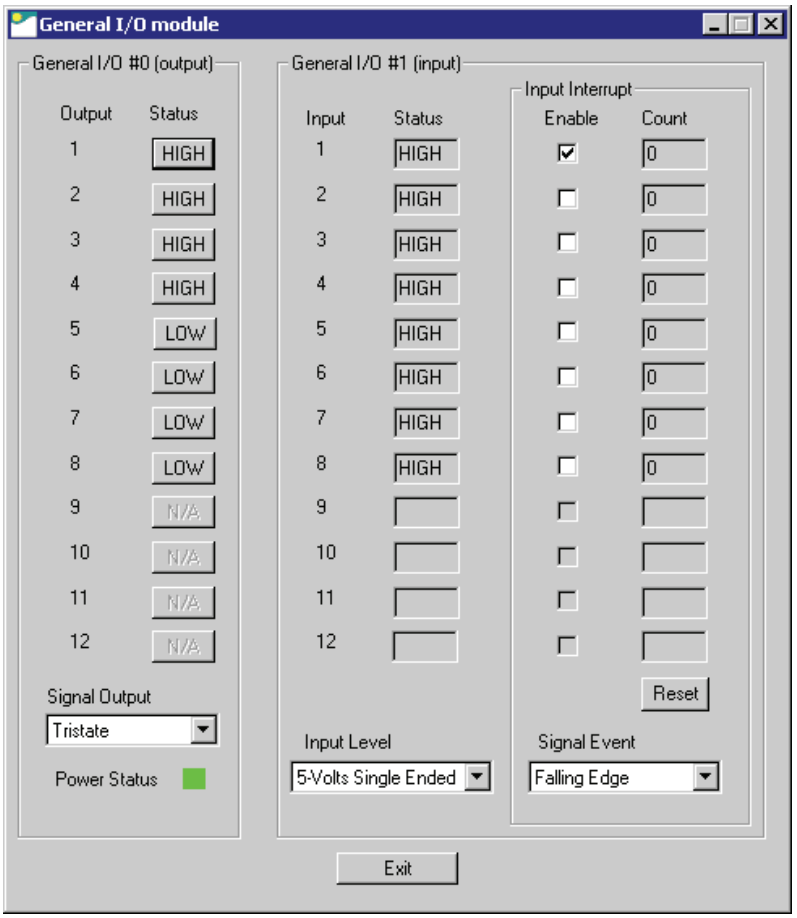
The I/O module control demo presents the I/O capabilities of the installed hardware. The following figure shows the X-I/O module connected to the X64 Xcelera-CL PX4 board.

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 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 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 Contact Information

Sales Information

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

Email: mailto:info@coreco.com

Canada/International

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

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

US Sales Office

DALSA
700 Technology Park Drive
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: Support requests for imaging product installations, 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 (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
Client **M**odification **I**nstruction. A client requested engineering change applied to a DALSA 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, 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

V

W

X