TELEDYNE DALSA

Teledyne DALSA • 7075 Place Robert-Joncas, Suite 142 • St-Laurent, Quebec, H4M 2Z2 • Canada
www.teledynedalsa.com

X64 Xcelera-LVDS PX4™

User's Manual

Edition 1.05

Part number OC-X4LM-PUSR0

NOTICE

Edition 1.05 November 11, 2011

Document Number: OC-X4LM-PUSR0

Printed in Canada

Contents

X64 Xcelera-LVDS PX4 Overview

Product Part Numbers

X64 Xcelera-LVDS PX4 Board

Item	Product Number
X64 Xcelera-LVDS PX4 with 128 MB of memory	OR-X4L0-XPX00
X-I/O Module (optional): provides an additional 8 input & 8 output general I/Os (see "Appendix: X-I/O Module Option" on page 113)	OC-IO01-STD00
For OEM clients, this manual in printed form, is available on request	OC-X4LM-USER0

X64 Xcelera-LVDS PX4 Software

Item	Product Number
Sapera LT version 6.10 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-LVDS PX4 Cables & Accessories

Item	Product Number
(optional) X64 Xcelera-LVDS PX4 shipped with an External Signals Connector Bracket Assembly, either with a **DB37 or DB25** connector (see the two product numbers below). Specify either cable if required, 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 107. This cable assembly connects to J3.	OR-X4CC-IOCAB
DB25 assembly see "External Signals Connector Bracket Assembly (Type 2)" on page 109. Provides direct compatibility with external cables made for products such as the X64-CL iPro. This cable assembly connects to J3.	OR-X4CC-0TIO2
DB25 male to color coded blunt end cable – 6 foot (1.82 meter) length	OC-COMC-XEND1
(optional) Power interface cable required when supplying power to cameras	OR-COMC-POW03

About the X64 Xcelera-LVDS PX4 Frame Grabber

Key Features

- Legacy support for LVDS area and Line Scan, monochrome and RGB digital cameras (EIA-644)
- Single slot solution for cameras with up to 8 taps
- Pixel clock: 1 Hz to 85 MHz
- Half-length PCIe x4card, compliant with PCIe Rev. 1.1
- Flat Field/Flat Line Correction
- Output lookup tables available for each mode
- Vertical Flip supported on board
- Compliant with DALSA Trigger-to-Image Reliability framework
- RoHS compliant

See "Technical Specifications" on page 81 for detailed information.

ACU-Plus: X64-LVDS Acquisition Control Unit

- Provides a flexible front-end for interfacing LVDS/RS422 cameras.
- ACU incorporates a fault-tolerant image synchronization design, allowing automatic detection, reporting and recovery from lost camera signals ensuring image sequence reliability.
- Embedded timing logic within the ACU-Plus identifies each acquired image with a time code.

DTE: Intelligent Data Transfer Engine

The X64 Xcelera-LVDS 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-LVDS PX4 is a universal PCI Express x4 board, compliant with the PCI Express 1.0a specification. The X64 Xcelera-LVDS 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-LVDS 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-LVDS PX4 board can also be used in a PCI Express x8 slot typically without issue.
- Refer to the computer documentation if there is only a PCI Express x16 slot. Such slots may not support PCI Express x4 products. Many computer motherboards only support x16 products in x16 slots (x16 slots are used with graphic video boards).

User Programmable Configurations

Use the X64 Xcelera-LVDS 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 13).

For the X64 Xcelera-LVDS PX4 board the firmware choices are:

- **One LVDS Camera Input, Data Sampling on the Rising Edge of the Pixel Clock** (*installation default selection*):
  Support for 1 LVDS camera with 1-8 taps. Data sampling with this firmware is compatible with the X64-LVDS board.
- **One LVDS Camera Input, Data Sampling on the Falling Edge of the Pixel Clock:**
  Support for 1 LVDS camera with 1-8 taps. Data sampling with this firmware is compatible with driver version 1.00 of the board.

Advanced Controls Overview

Visual Indicators

X64 Xcelera-LVDS PX4 features a LED indicator to facilitate system installation and setup. This provides visual feedback indicating when the camera is properly connected and sending data. See "LVDS Camera Connections & " on page 20 for led color definitions.

External Event Synchronization

Trigger inputs and strobe signals provide precisely synchronize image captures with external events.

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-LVDS PX4. The X-I/O module provides two 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.

Purchase the X-I/O module preinstalled on the X64 Xcelera-LVDS PX4 board or later for installation into the computer system. The module occupies one adjacent PCI slot and connects to the X64 Xcelera-LVDS PX4 via a ribbon cable. X-I/O Module external connections use the DB37 connector on the module bracket.

X-I/O requires X64 Xcelera-LVDS 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 113 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-LVDS 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.

Important: Never remove or install any hardware component with the computer power on. Disconnect the power cord from the computer to disable the power standby mode. This prevents the case where some computers unexpectedly power up when a board is installed.

Sapera LT Library Installation

Note: to install Sapera LT and the X64 Xcelera-LVDS 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 board device driver.

- Insert the DALSA Sapera CD-ROM. With **AUTORUN** enabled on your computer, the installation menu runs automatically.
- If **AUTORUN** is disabled, 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-LVDS PX4 Hardware and Driver

In a Windows XP/Vista/7 System

- Turn the computer off, disconnect the power cord (disables power standby mode), and open the computer chassis to allow access to the expansion slot area.
- Install the X64 Xcelera-LVDS PX4 into a free PCI Express x4 expansion slot. The X64 Xcelera-LVDS PX4 could also be installed in a PCI Express x8.
- 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-LVDS PX4 and start its **Found New Hardware Wizard**. Click on the **Cancel** button to close the Wizard.
- Insert the DALSA Sapera CD-ROM. With **AUTORUN** enabled on your computer, the installation menu opens. Install the X64 Xcelera-LVDS PX4 driver.
- Without **AUTORUN** 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-LVDS PX4 driver. During the late stages of the installation, the X64 Xcelera-LVDS PX4 firmware loader application starts. See the following section for a detailed description.
- If Windows displays any unexpected message concerning the installed board, power off the system and verify that the X64 Xcelera-LVDS LX4 is properly installed in the slot.
- When using Windows XP, if a message stating that the X64 Xcelera-LVDS PX4 software has not passed Windows Logo Testing displays, click on Continue Anyway to finish the driver installation. Reboot the computer if prompted to do so.
- When using Windows Vista/7, a message asking to install the DALSA device software displays. Click Install.

X64 Xcelera-LVDS PX4 Firmware Loader

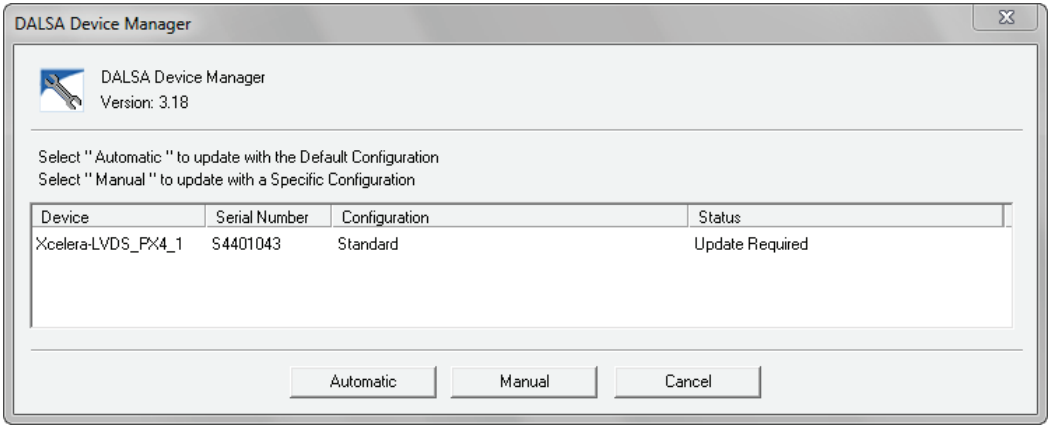
After Windows boots, the Device Manager-Firmware Loader program automatically executes at the end of the driver installation and on every subsequent reboot of the computer. It will determine if the X64 Xcelera-LVDS PX4 requires a firmware update. If firmware is required, a dialog displays, which also allows the user to load firmware for alternate operational modes of the X64 Xcelera-LVDS PX4.

***Important*:** In the rare case of firmware loader errors please see "Recovering from a Firmware Update Error" on page 39.

Firmware Update: Automatic Mode

Click **Update All** to update the X64 Xcelera-LVDS PX4 firmware. The **X64 Xcelera-LVDS PX4** currently supports one standard firmware configuration for one LVDS camera.

If multiple X64 Xcelera-LVDS PX4 boards are in the system, new firmware will update each board. If any installed X64 Xcelera-LVDS PX4 board installed in a system already has the correct firmware version, an update is not required. In the following screen shot, a single installed X64 Xcelera-LVDS PX4 board requires new firmware.



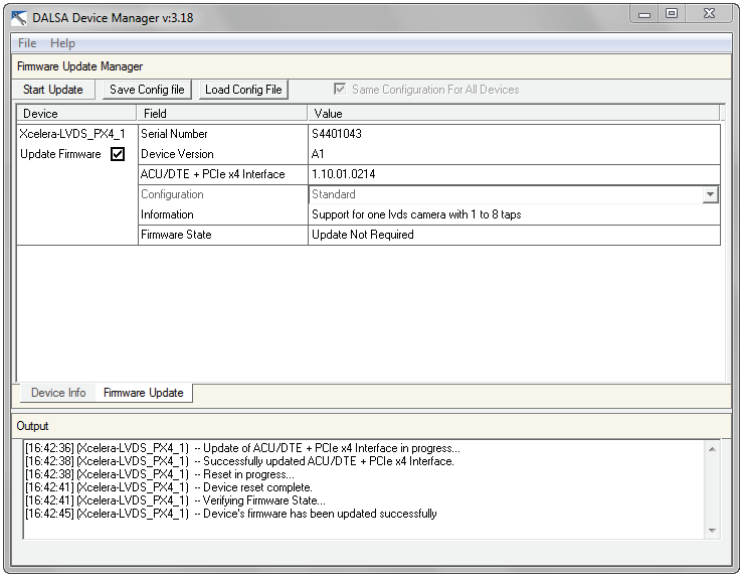
Firmware Update: Manual Mode

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

The figure below shows the Device Manager manual firmware screen, which displays information on all installed X64 Xcelera-LVDS PX4 boards, their serial numbers, and their firmware components.

Perform a manual firmware update as follows:

- Select the X64 Xcelera-LVDS 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 is complete.



Executing the Firmware Loader from the Start Menu

If required, run the Firmware Loader program via the Windows Start Menu shortcut **Start • Programs • DALSA • X64 Xcelera-LVDS PX4 Driver • Firmware Update**. A firmware change after installation would be required to select a different configuration mode if available. See "User Programmable Configurations" on page 8.

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 the board driver and if required upgrade Sapera, as described below.

Board Driver Upgrade Only

Download upgrades to acquisition board drivers from the DALSA web site www.dalsa.com/mv/support.

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 "Contact Information" on page 127).

To upgrade the board driver only:

- Logon the computer as an administrator or with an account that has administrator privileges.
- In **Windows XP**, from the start menu select **Start • Settings • Control Panel • Add or Remove Programs**. Select the DALSA Xcelera board driver and click **Remove**. When the driver un-install is complete, reboot the computer and logon the computer as an administrator again.
- In **Windows Vista/7**, from the start menu select **Start • Settings • Control Panel • Programs and Features**. Double-click the DALSA Xcelera board driver and click **Remove**.
- 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-LVDS PX4 Hardware and Driver" on page 12.
- Note that you cannot install a DALSA board driver without Sapera LT installed on the computer.

Sapera and Board Driver Upgrades

To upgrade both Sapera and the acquisition board driver, follow the procedure described below.

- Logon the computer as an administrator or with an account that has administrator privileges.
- In **Windows XP**, from the start menu select **Start • Settings • Control Panel • Add or Remove Programs**. Select the DALSA Xcelera board driver and click **Remove**. Follow by also removing the older version of Sapera LT.
- In **Windows Vista/7**, from the start menu select **Start • Settings • Control Panel • Programs and Features**. Double-click the DALSA Xcelera board driver and click **Remove**. Follow by also removing the older version of Sapera LT.
- 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 11 and "Installing X64 Xcelera-LVDS PX4 Hardware and Driver" on page 12 for installation procedures.

Enabling the Serial Control Port

The X64 Xcelera-LVDS PX4 includes a serial communication port for direct camera control by the frame grabber. The port is available on the VHDCI J12-Connector 1, where pin-32 is serial out and pin-34 is serial in (see "J12: Dual 68 Pin VHDCI Connectors" on page 88). The X64-LVDS 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 at speeds up to 115 kbps.

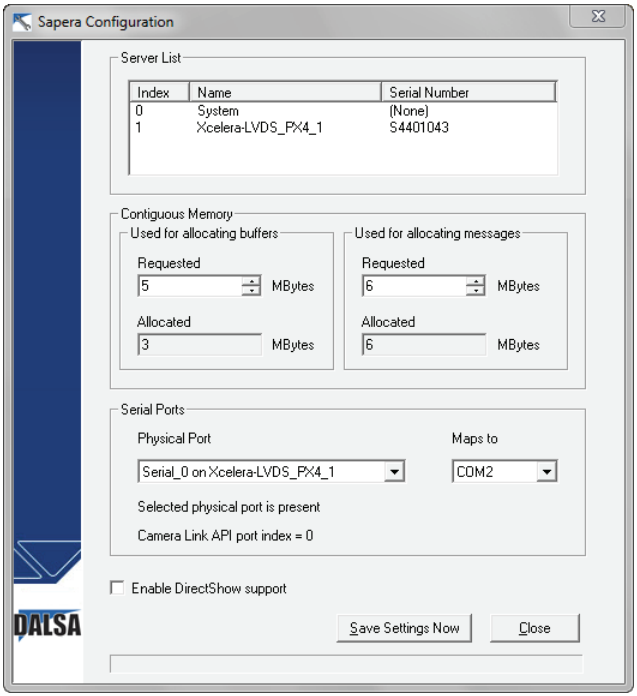
Note: if the serial communication program can directly select the X64 Xcelera-LVDS PX4 serial port then mapping to a system COM port is not necessary.

Map the X64 Xcelera-LVDS PX4 serial port 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:

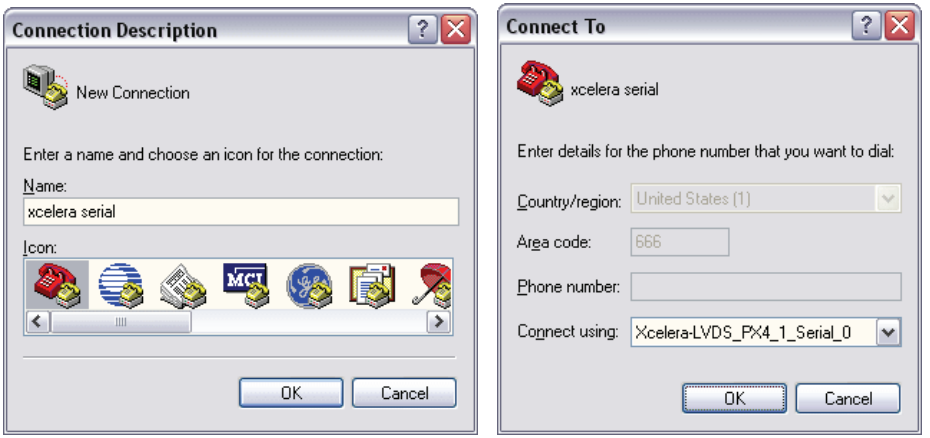
- Use the **Physical Port** drop menu to select the Sapera board device from all available Sapera boards with serial ports (when more then one board is in the system).
- Use the **Maps to** drop menu to assign an available COM number to that Sapera board serial port.
- Click on the **Save Settings Now** button then the **Close** button. Reboot the computer to enable the serial port mapping.
- The X64 Xcelera-LVDS 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 does not show in the **Windows Control Panel • System Properties • Device Manager**.
- An example setup using Windows HyperTerminal follows.



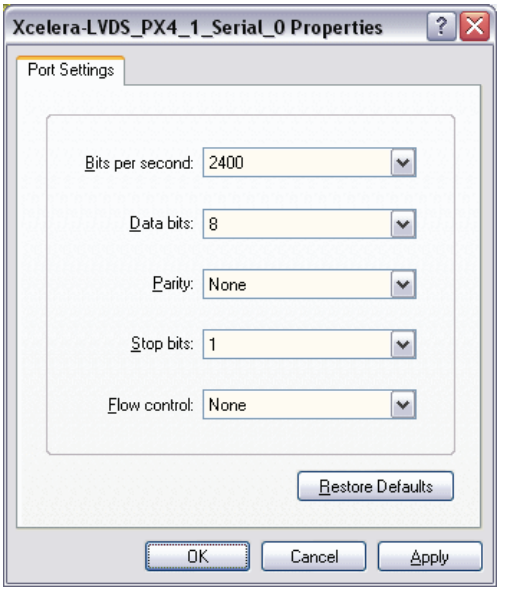
Setup Example with Windows HyperTerminal

The following instructions apply to **Windows XP** only. Windows Vista and Windows 7 no longer include HyperTerminal, but a number of alternatives are available for download by searching the internet.

- Run HyperTerminal and type a name for the new connection when prompted. Then click OK.
- On the following dialog, select the COM port to connect. The port could be the COM port mapped to the X64 Xcelera-LVDS 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 too. Note that the X64 Xcelera-LVDS PX4 serial port does not support hardware flow control.



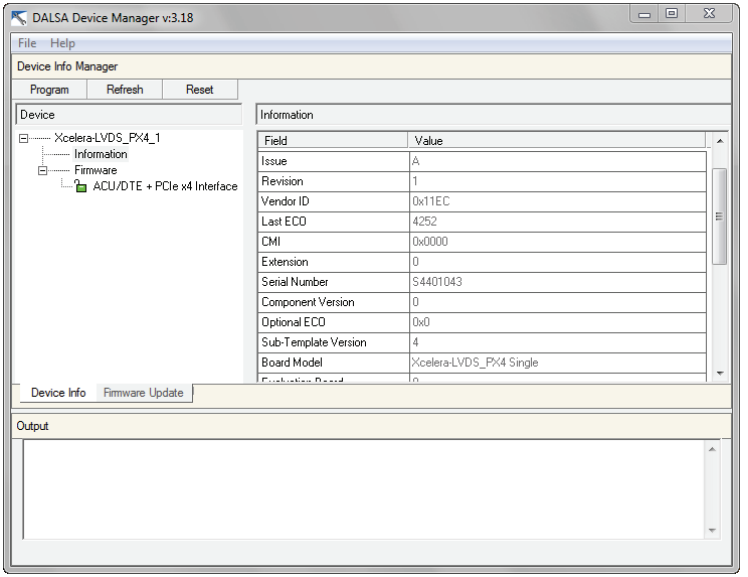
Displaying X64 Xcelera-LVDS PX4 Board Information

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

Device Manager – Board Viewer

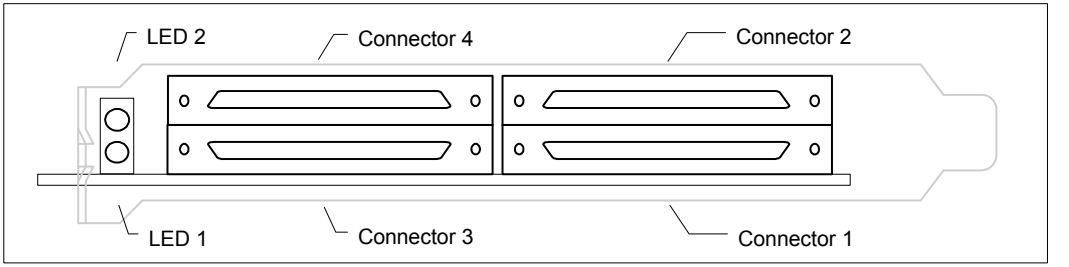
The following screen image shows the Device Manager program with the Information/Firmware tab active. The left window displays all X64 **Xcelera-LVDS** PX4 boards in the system and their individual device components. The right window displays the information stored in the selected board device. The following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

Generate the X64 **Xcelera-LVDS** PX4 device manager report file (BoardInfo.txt) by clicking **File • Save Device Info**. DALSA Technical Support may request this to aid in troubleshooting installation or operational problems.



LVDS Camera Connections & Status LEDs

X64 Xcelera-LVDS PX4 End Bracket View



- Connector #1 is for camera taps 1 and 2, while connector #2 is for camera taps 3 and 4.
- Connector #3 is for camera taps 5 and 6, while connector #4 is for camera taps 7 and 8.

Complete the hardware installation process with the connection of a supported camera to the X64 Xcelera-LVDS PX4 board using appropriate VHDCI cables. The camera and board cable connectors are labeled when a cable is purchased from DALSA. Connect the cable as indicated. If however, you fabricate the cable yourself, contact the DALSA Montreal Camera Application group for information and cable diagrams applicable to your camera.

Note: Refer to "External Signals Connector Bracket Assembly" on page 107 for power connections when the X64 Xcelera-LVDS PX4 supplies camera power.

Status LEDs Description

Status LED 1:	
Red	No camera connected or camera has no power
Green	Camera connected and 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:	
Off	Board initialization good
Red	Board initialization error on computer boot-up

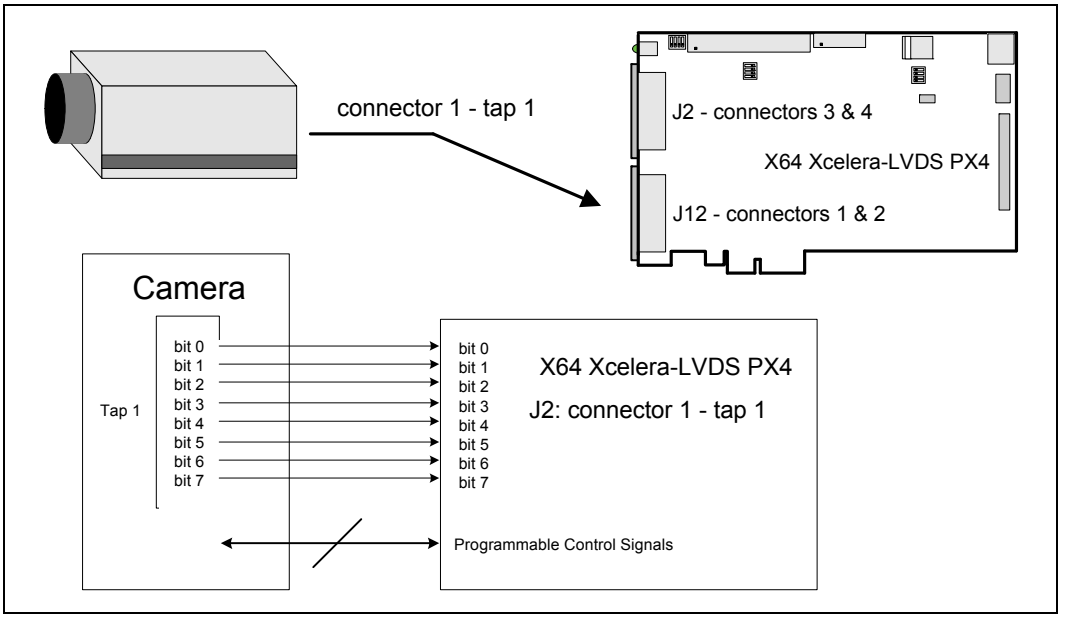
Contact DALSA or browse our web site www.dalsa.com/mv/support for the latest information on X64 Xcelera-LVDS PX4 supported cameras.

Camera Connection Examples

The following diagrams are examples of camera connections for the X64 Xcelera-LVDS PX4. The information presented is generic and does not detail specifics as to camera brand and its signal specifications, or camera cabling requirements.

The various data input configurations are automatically programmed by the parameters defined in the Sapera camera file loaded for the camera in use. Use the Sapera CamExpert tool to configure camera files.

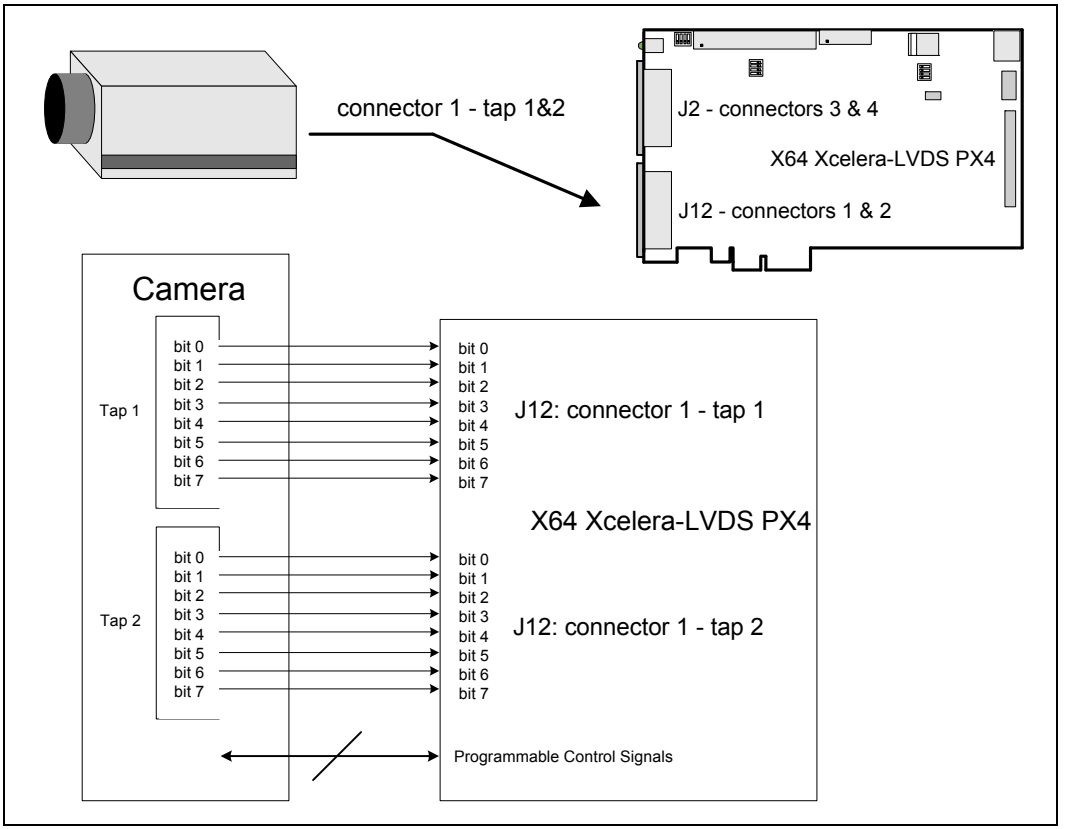
One Camera – One 8-bit Channel or Tap



If the camera has one channel or tap that outputs 8-bits per pixel:

Connect the camera to the 8-bit data port Tap 1, on input connector 1.

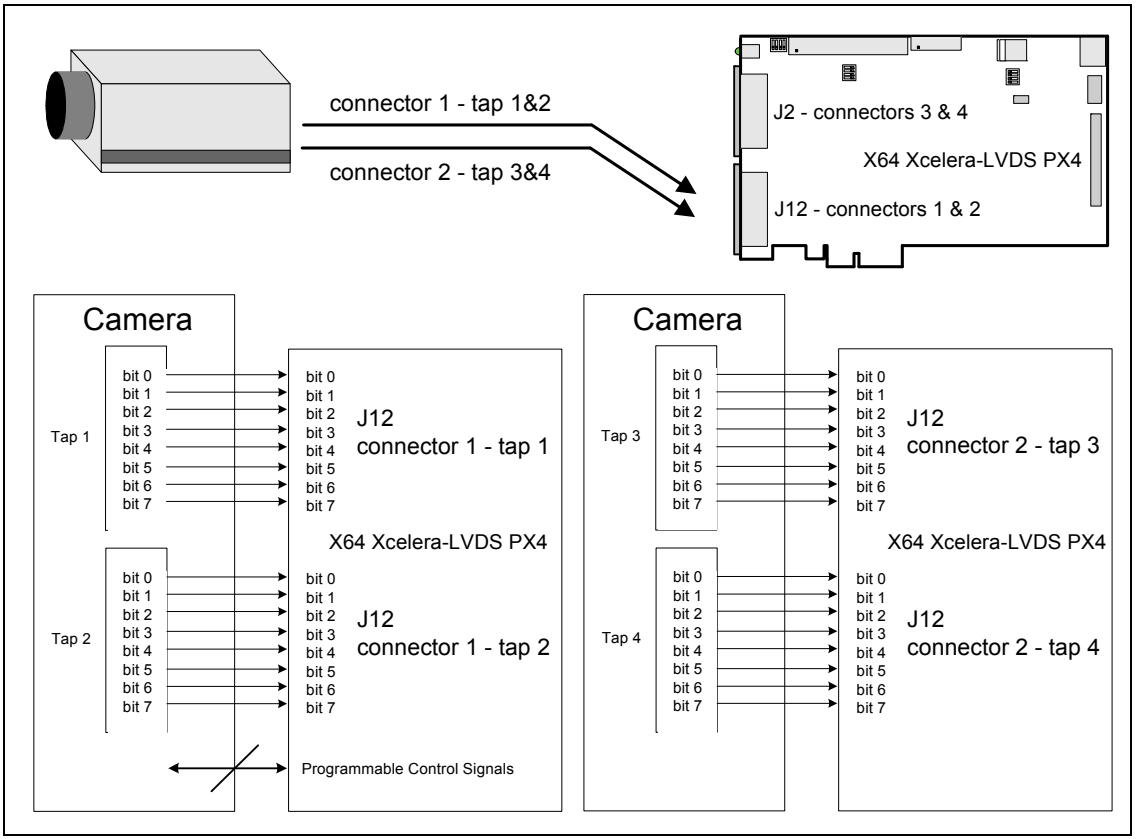
One Camera – Two 8-bit Channels or Taps



If the camera has two channels or taps that output 8-bits per pixel:

Connect the camera channel/tap 1 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel/tap 2 to the 8-bit data port Tap 2, on input connector 1.

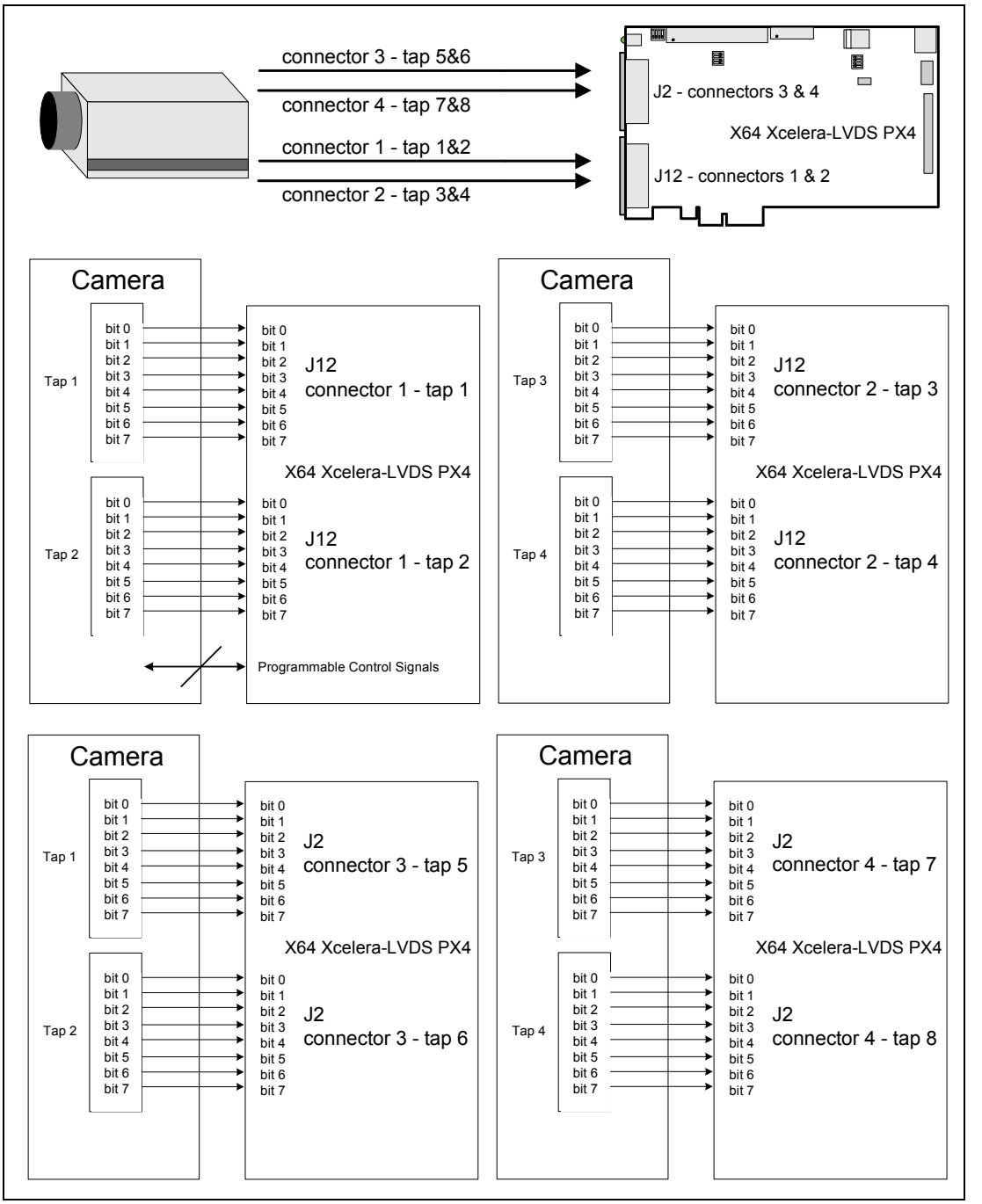
One Camera – Four 8-bit Channels or Taps



If the camera has four channels or taps that output 8-bits per pixel:

Connect the camera channel/tap 1 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel/tap 2 to the 8-bit data port Tap 2, on input connector 1.
Connect the camera channel/tap 3 to the 8-bit data port Tap 3, on input connector 2.
Connect the camera channel/tap 4 to the 8-bit data port Tap 4, on input connector 2.

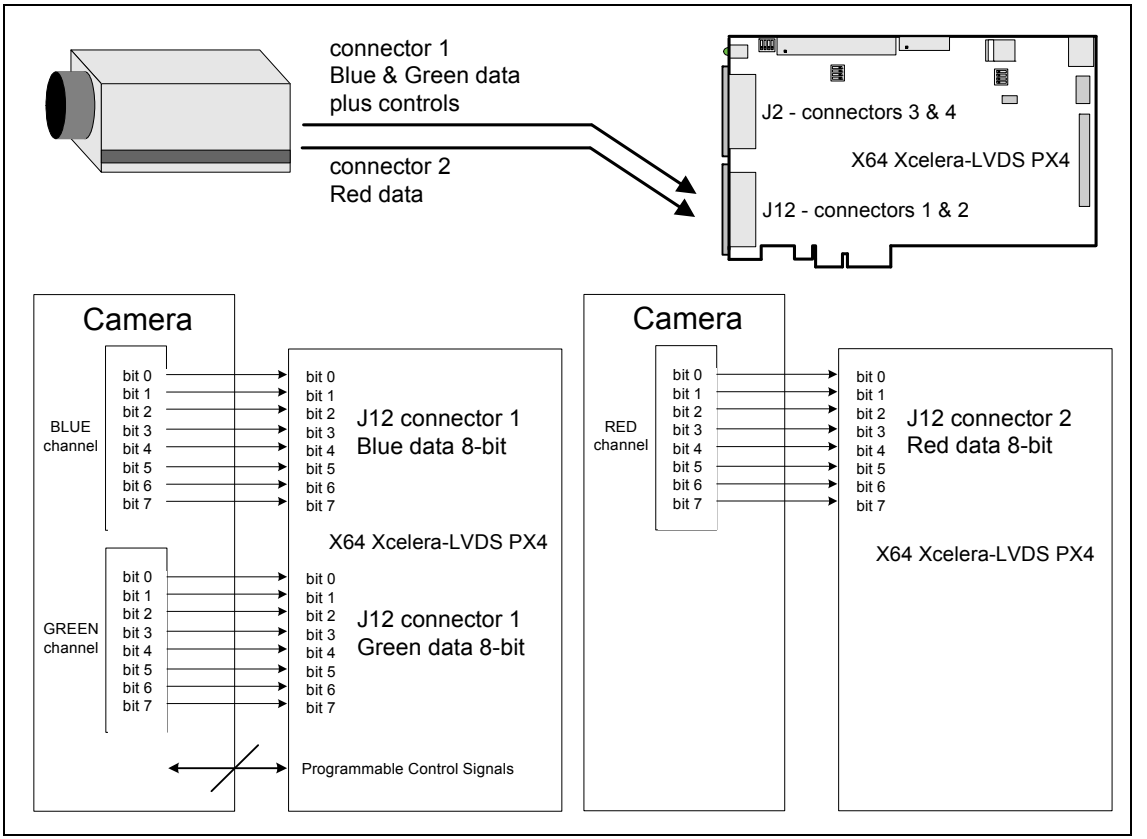
One Camera – Eight 8-bit Channels or Taps



If the camera has eight channels or taps that output 8-bits per pixel:

Connect the camera channel/tap 1 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel/tap 2 to the 8-bit data port Tap 2, on input connector 1.
Connect the camera channel/tap 3 to the 8-bit data port Tap 3, on input connector 2.
Connect the camera channel/tap 4 to the 8-bit data port Tap 4, on input connector 2.
Connect the camera channel/tap 5 to the 8-bit data port Tap 5, on input connector 3.
Connect the camera channel/tap 6 to the 8-bit data port Tap 6, on input connector 3.
Connect the camera channel/tap 7 to the 8-bit data port Tap 7, on input connector 4.
Connect the camera channel/tap 8 to the 8-bit data port Tap 8, on input connector 4.

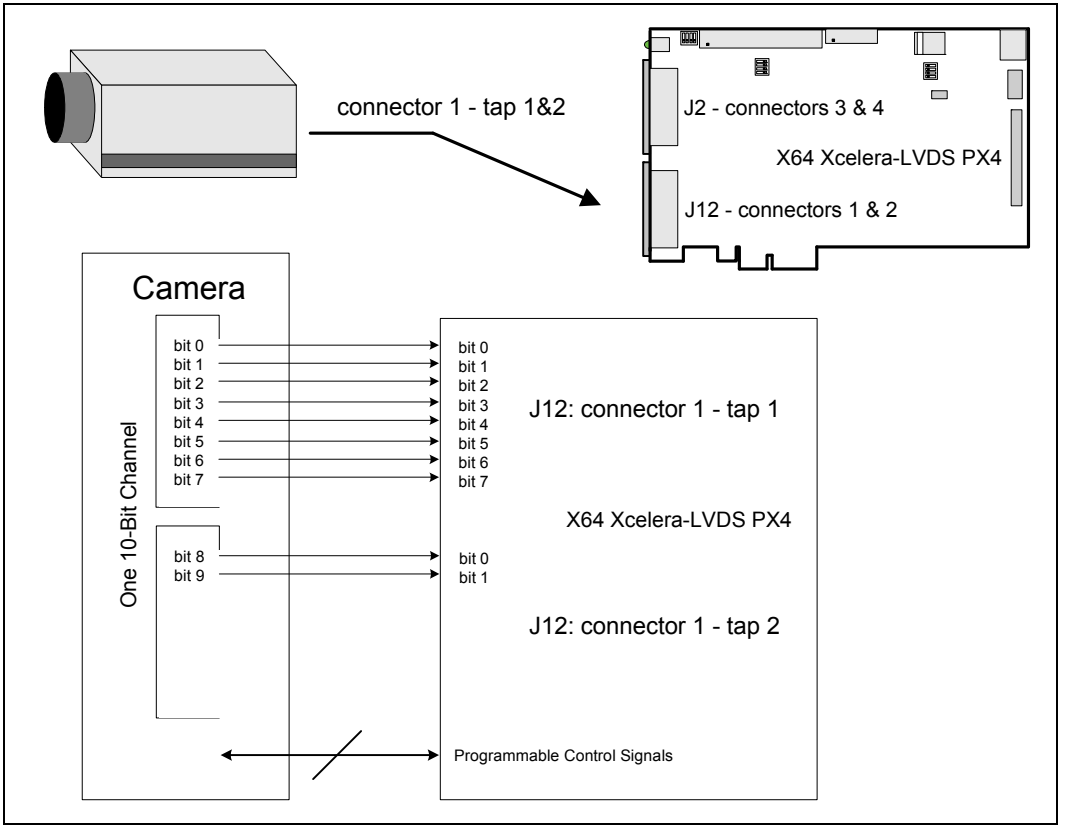
One Camera – 24-bit RGB



If the camera outputs RGB 24-bit data:

Connect the camera Blue data to input connector 1.
Connect the camera Green data to input connector 1.
Connect the camera Red data to input connector 2.
See "J12 – Connector 1: RGB-24 & RGB-30 Pinout" on page **93** and "J12 – Connector 2: RGB-24 & RGB-30 Pinout" on page **95** for details.

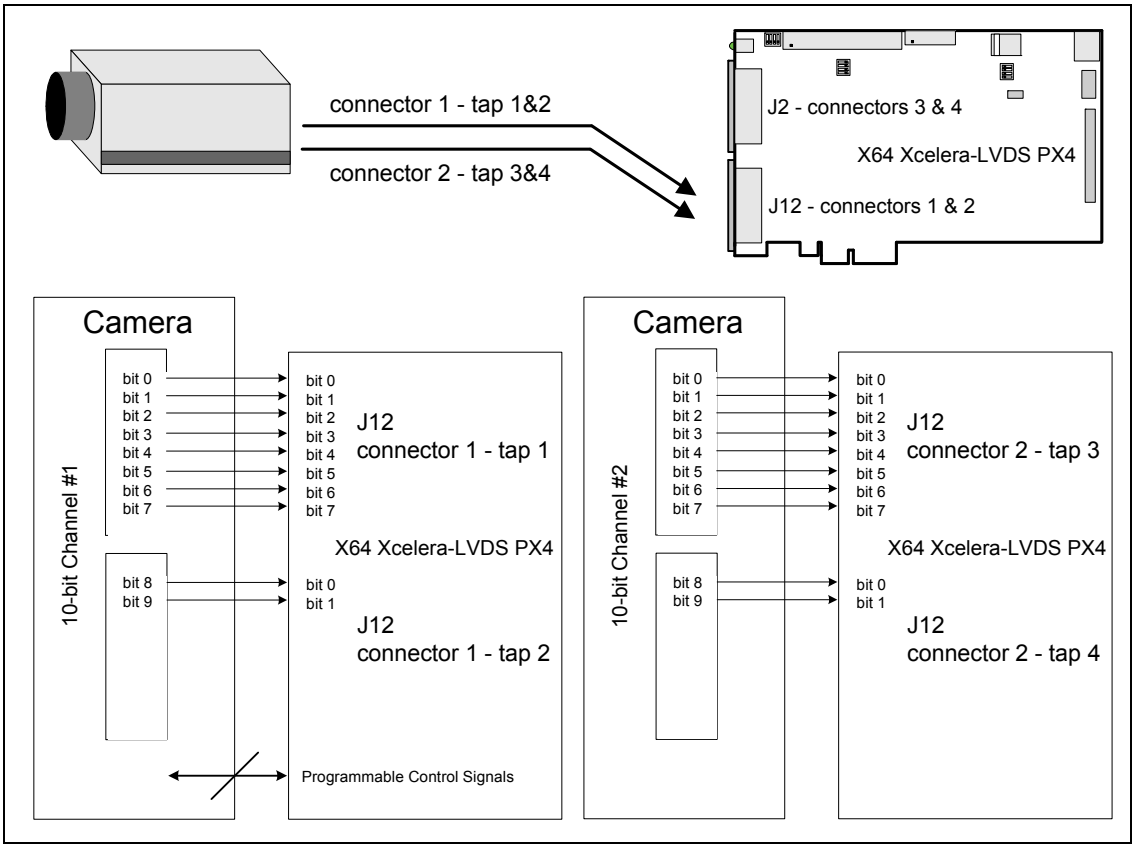
One Camera – One 10-bit Channel



If the camera has one channel that outputs 10-bits per pixel:

Connect the camera data bits 0-7 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera data bits 8, 9 to the first two bits on data port Tap 2, on input connector 1.

One Camera – Two 10-bit Channels

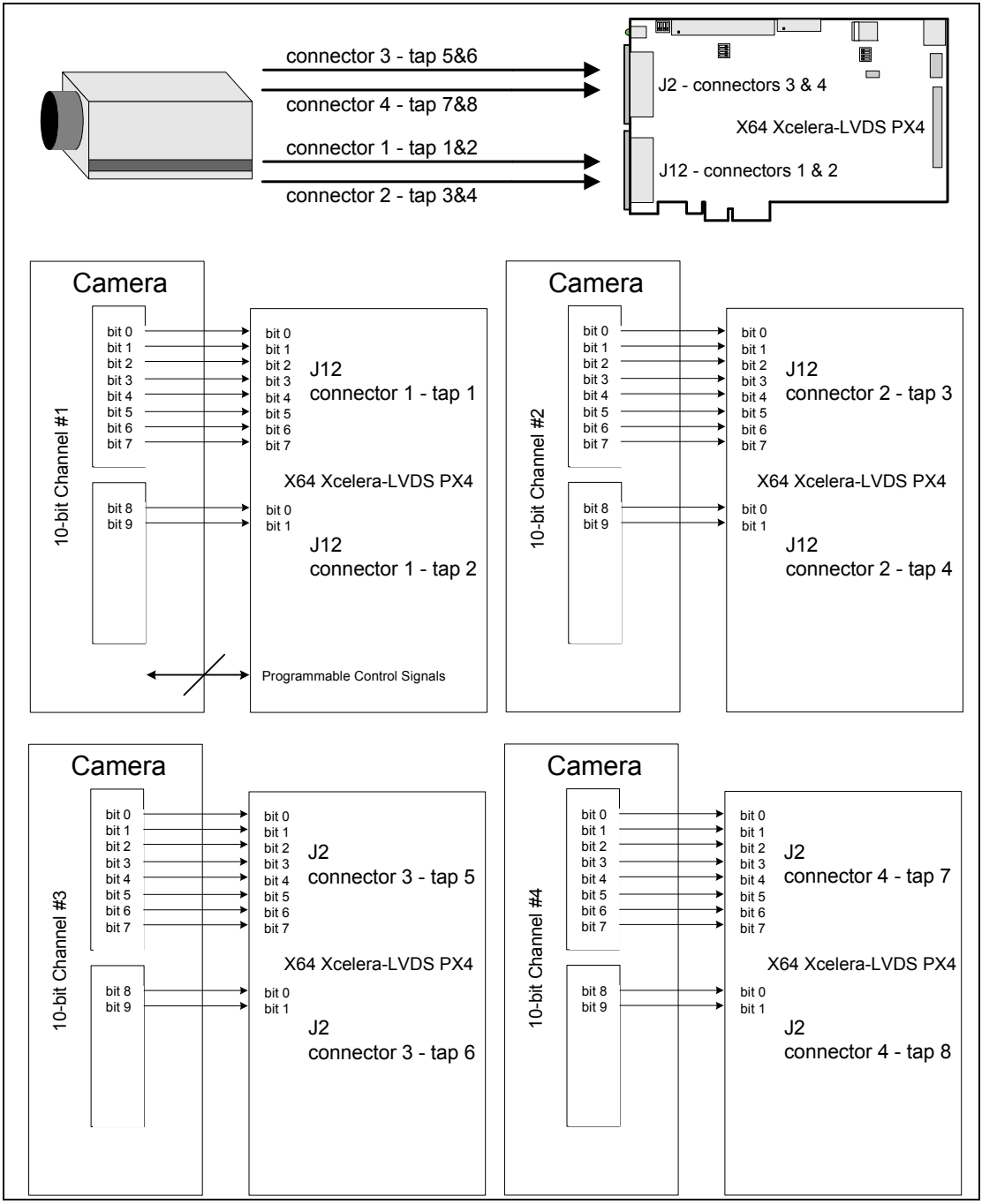


If the camera has two channels that output 10-bits per pixel:

Connect the camera channel #1 data bits 0-7 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel #1 data bits 8, 9 to the first two bits on data port Tap 2, on input connector 1.

Connect the camera channel #2 data bits 0-7 to the 8-bit data port Tap 3, on input connector 2.
Connect the camera channel #2 data bits 8, 9 to the first two bits on data port Tap 4, on input connector 2.

One Camera – Four 10-bit Channels



If the camera has four channels that output 10-bits per pixel:

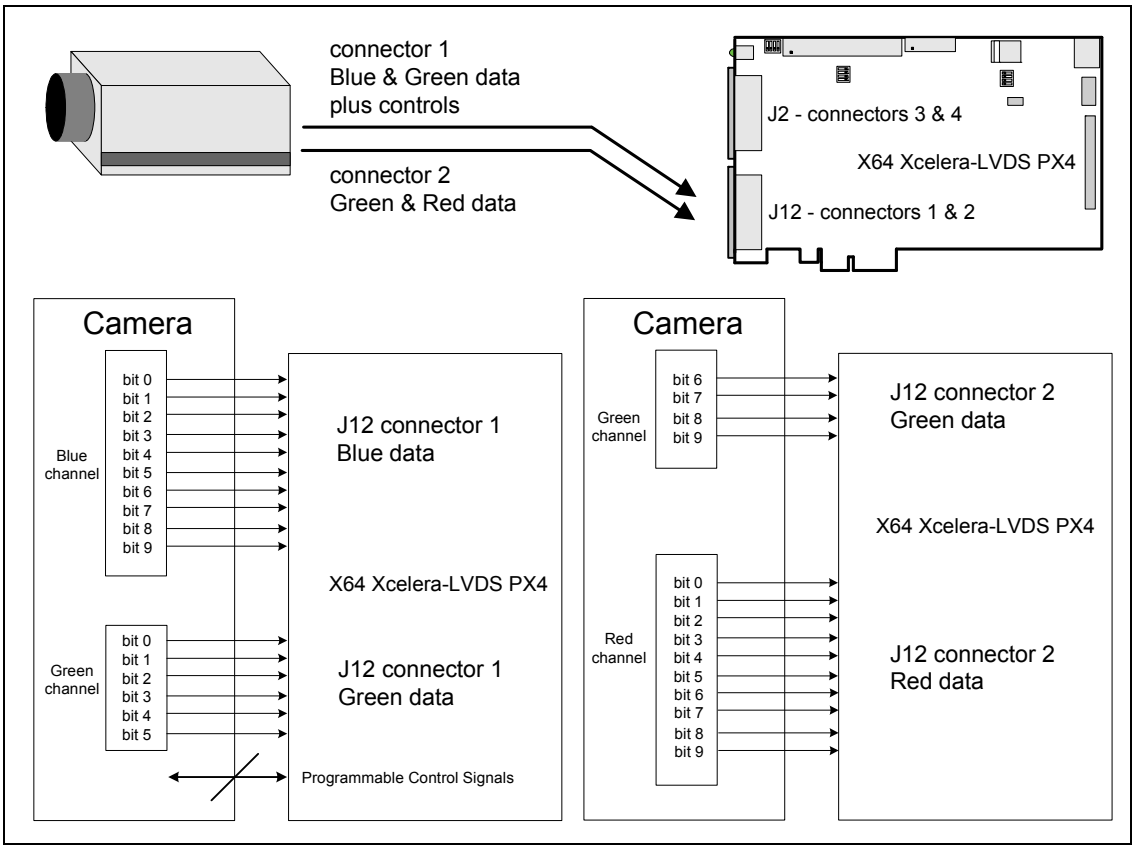
Connect the camera channel #1 data bits 0-7 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel #1 data bits 8, 9 to the first two bits on data port Tap 2, on input connector 1.

Connect the camera channel #2 data bits 0-7 to the 8-bit data port Tap 3, on input connector 2.
Connect the camera channel #2 data bits 8, 9 to the first two bits on data port Tap 4, on input connector 2.

Connect the camera channel #3 data bits 0-7 to the 8-bit data port Tap 5, on input connector 3.
Connect the camera channel #3 data bits 8, 9 to the first two bits on data port Tap 6, on input connector 3.

Connect the camera channel #4 data bits 0-7 to the 8-bit data port Tap 7, on input connector 4.
Connect the camera channel #4 data bits 8, 9 to the first two bits on data port Tap 8, on input connector 4.

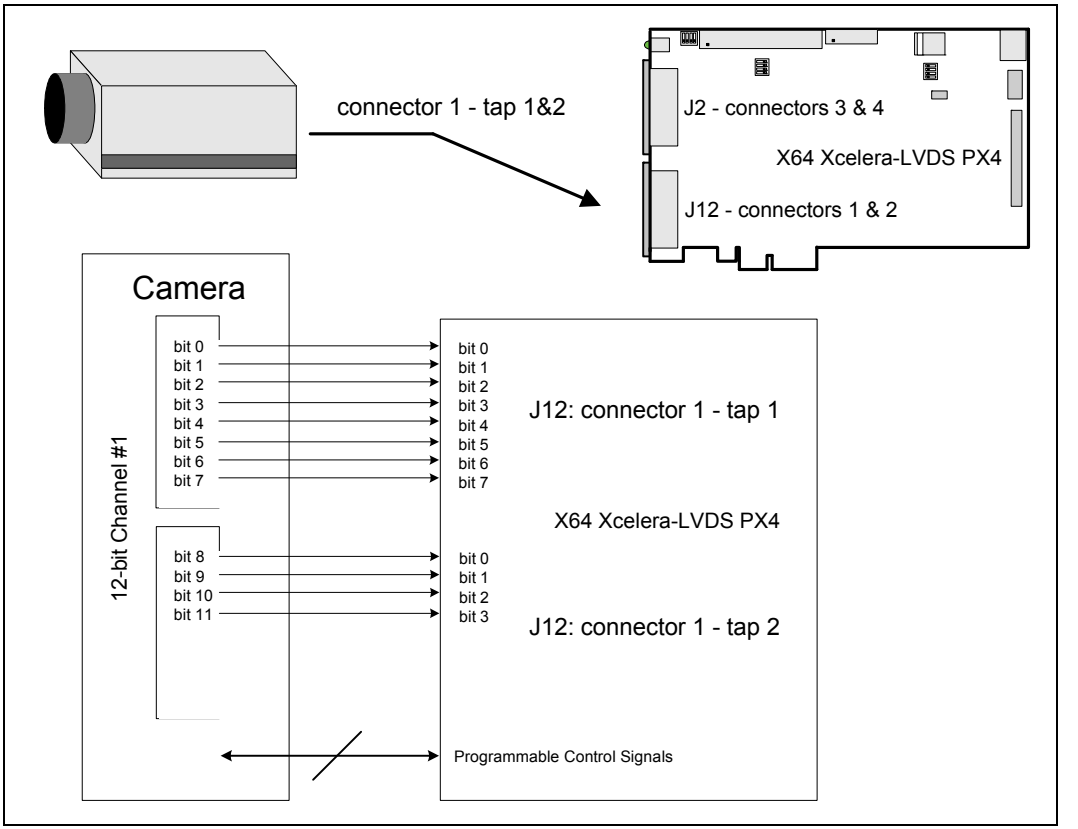
One Camera – 30-bit RGB



If the camera outputs RGB 30-bit data:

Connect the camera Blue data to input connector 1.
Connect the camera Green data (bits 0-5) to input connector 1.
Connect the camera Green data (bits 6-9) to input connector 2.
Connect the camera Red data to input connector 2.
See "J12 – Connector 1: RGB-24 & RGB-30 Pinout" on page **93** and "J12 – Connector 2: RGB-24 & RGB-30 Pinout" on page **95** for details.

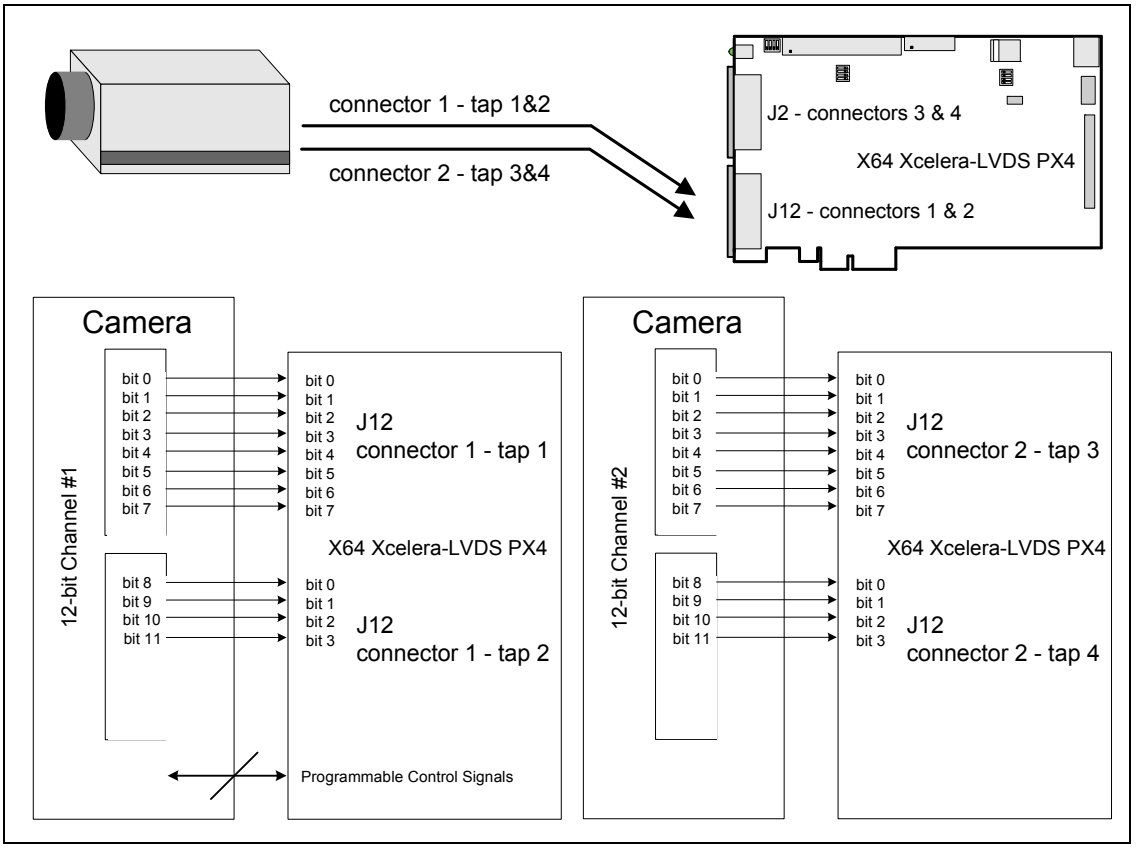
One Camera – One 12-bit Channel



If the camera has one channel that outputs 12-bits per pixel:

Connect the camera data bits 0-7 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera data bits 8-11 to the first four bits on data port Tap 2, on input connector 1.

One Camera – Two 12-bit Channels

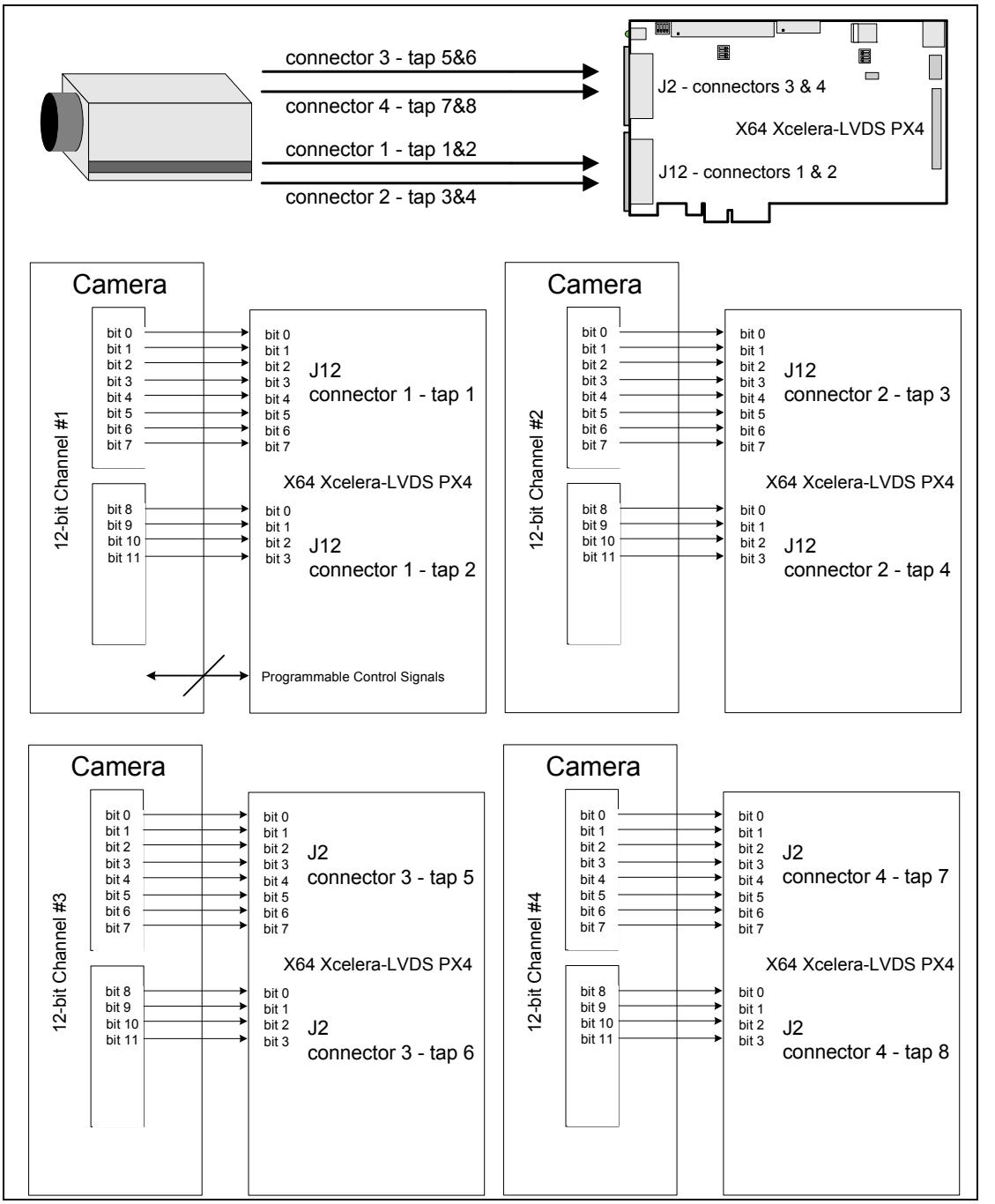


If the camera has two channels that output 12-bits per pixel:

Connect the camera channel #1 data bits 0-7 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel #1 data bits 8-11 to the first four bits on data port Tap 2, on input connector 1.

Connect the camera channel #2 data bits 0-7 to the 8-bit data port Tap 3, on input connector 2.
Connect the camera channel #2 data bits 8-11 to the first four bits on data port Tap 4, on input connector 2.

One Camera – Four 12-bit Channels



If the camera has four channels that output 12-bits per pixel:

Connect the camera channel #1 data bits 0-7 to the 8-bit data port Tap 1, on input connector 1.
Connect the camera channel #1 data bits 8-11 to the first four bits on data port Tap 2, on input connector 1.

Connect the camera channel #2 data bits 0-7 to the 8-bit data port Tap 3, on input connector 2.
Connect the camera channel #2 data bits 8-11 to the first four bits on data port Tap 4, on input connector 2.

Connect the camera channel #3 data bits 0-7 to the 8-bit data port Tap 5, on input connector 3.
Connect the camera channel #3 data bits 8-11 to the first four bits on data port Tap 6, on input connector 3.

Connect the camera channel #4 data bits 0-7 to the 8-bit data port Tap 7, on input connector 4.
Connect the camera channel #4 data bits 8-11 to the first four bits on data port Tap 8, on input connector 4.

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 successfully allocated. 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 55) 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 if the requested 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. As an example, Window XP 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, which is used to store arguments when a Sapera functions are called. Increase this value if you are using functions with large arguments, such as arrays and experience any memory errors.

Troubleshooting Installation Problems

Overview

DALSA has tested the X64 Xcelera-LVDS PX4 (and the X64 family of products) 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.

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 "Contact Information" on page 127.

Problem Type Summary

X64 Xcelera-LVDS PX4 problems either are installation types because the system PCIe bus does not recognize the board hardware (i.e. trained) or function errors due to camera connections or bandwidth issues. The following links jump to various topics in this troubleshooting section.

First Step: Check the Status LED

A RED Status LED 1 indicates a camera problem, while various Green states indicate the acquisition mode.

Status LED 2, if flashing RED, indicates a PCIe bus problem. If you run the PCI Diagnostics tool, the board is not in the PCI device list. If the board is installed in a computer which supports PCIe GEN2 expansion slots, see section " SW3-2 GEN2 Slot Workaround Details" on page 86.

The complete status LED description is available in the technical reference section (see "Status LEDs Functional Description" on page 87).

Possible Installation Problems

- **Hardware PCI bus conflict:** When a new installation produces PCI bus error messages or the board driver does not install, it is important to verify that there are no conflicts with other PCI or system devices already installed. Use the DALSA PCI Diagnostic tool as described in "Checking for PCI Bus Conflicts" on page 37. Also, verify the installation via the "Driver Information via the Device Manager Program" on page 40.
- **Gen2 slot errors:** There is a PCI bus error message from the computer bios. Follow the instructions " SW3-2 GEN2 Slot Workaround Details" on page 86.
- **Verify Sapera and Board drivers:** If there are errors when running applications, confirm that all Sapera and board drivers are running. See "Sapera and Hardware Windows Drivers" on page 39 for details. In addition, DALSA technical support will ask for the log file of messages by DALSA drivers. Follow the instructions describe in "DALSA Log Viewer" on page 41.
- **Firmware update error:** There was an error during the board update procedure. The user usually easily corrects this by following the instructions "Recovering from a Firmware Update Error" on page 39.
- Installation went well but the board does not work or stopped working. Review these steps described in "Symptoms: CamExpert Detects no Boards" on page 42.

Possible Functional Problems

- **Driver Information:** Use the DALSA device manager program to view information about the installed X64 Xcelera-LVDS PX4 board and driver. See "Driver Information via the Device Manager Program" on page 40.

Sometimes the problem symptoms are not the result of an installation issue but due to other system issues. Review the sections described below for solutions to various X64 Xcelera-LVDS PX4 functional problems.

- "Symptoms: X64 Xcelera-LVDS PX4 Does Not Grab" on page 42
- "Symptoms: Card grabs black" on page 43
- "Symptoms: Card acquisition bandwidth is less than expected" on page 43

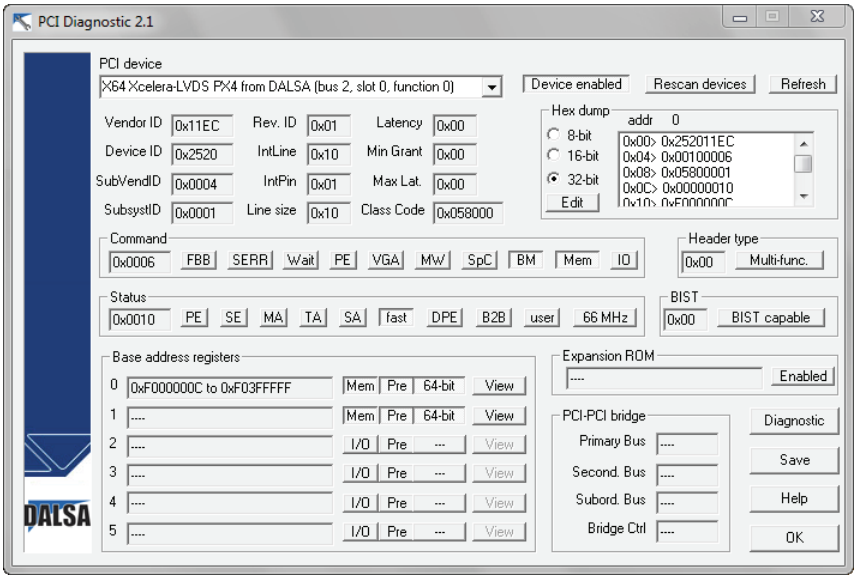
Troubleshooting Procedures

The following sections provide information and solutions to possible X64 Xcelera-LVDS PX4 installation and functional problems. A summary of these topics is in the previous section of this manual.

Checking for PCI Bus Conflicts

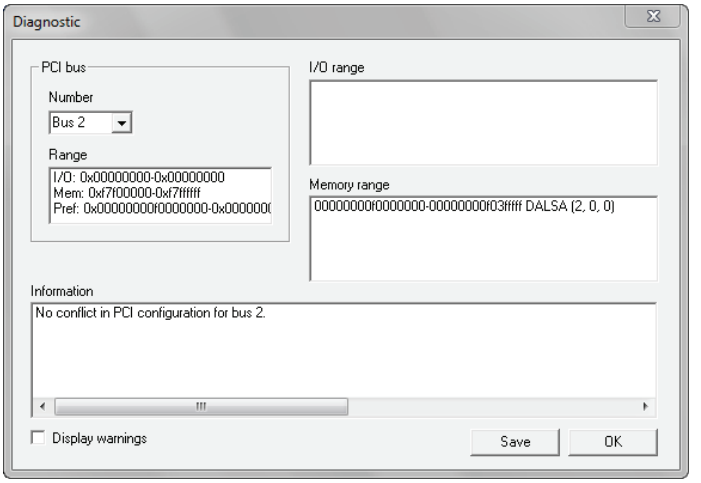
One of the first items to check when there is a problem with any PCI board is to examine the system PCI configuration and ensure that there are no conflicts with other PCI or system devices. The *PCI Diagnostic* program (**cpcidiag.exe**) allows examination of the PCI configuration registers and can 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.



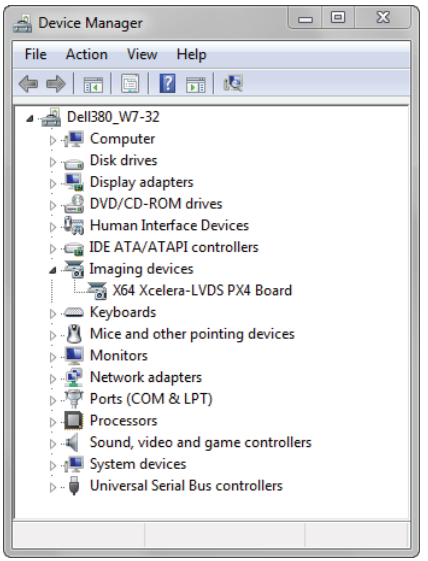
Clicking on the **Diagnostic** button opens a new window with the diagnostic report. From the PCI Bus Number drop menu, select the bus number associated with the X64 Xcelera-LVDS PX4—in this example, the slot is bus 2.

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.



Windows Device Manager

From the Windows Start Menu, select **Start • Control Panel • System • Hardware • Device Manager**. As shown in the following screen images, look for *X64 Xcelera-LVDS 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 has no conflicts.



GEN2 PCI Slot Computer Issue

At boot time, the PX4 status LED 2 keeps on flashing red. If you run the PCI Diagnostics tool, the PX4 is not in the PCI device list. If the board is installed in a computer which supports PCIe GEN2 expansion slots, see section " SW3-2 GEN2 Slot Workaround Details" on page 86.

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 **System Drivers**. Make certain the following drivers have started for the **X64 Xcelera-LVDS PX4**.

Device	Description	Type	Started
Corx64xcelerapx4lvds	X64 Xcelera-LVDS 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.

Recovering from a Firmware Update Error

This procedure is required if any failure occurred while updating the X64 Xcelera-LVDS PX4 firmware on installation or during a manual firmware upgrade. On the chance 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 brownout.
- System PCI bus or checksum errors.
- PCI bus timeout conditions due to other devices.
- Users forcing an upload using an invalid firmware source file.

When the X64 Xcelera-LVDS PX4 firmware is corrupted, executing a manual firmware upload will not work because the firmware loader cannot 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, reboot the board to initialize in its normal operational mode.

- Note that this procedure may require removing the X64 Xcelera-LVDS 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-LVDS PX4 is SW3-1 (see "SW3: " on page 86).
- 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 14).
- 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-LVDS PX4 board.

Driver Information via the Device Manager Program

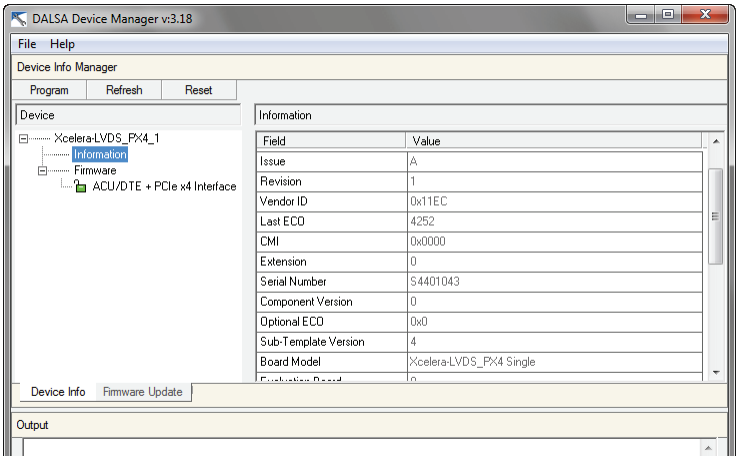
The Device Manager program provides a convenient method of collecting information about the installed X64 Xcelera-LVDS PX4 Full. System information such as operating system, computer CPU, system memory, PCI configuration space, plus X64 Xcelera-LVDS PX4 firmware information is displayed or written to a text file (default file name – BoardInfo.txt). Note that this tool is also used to upload firmware to the X64 Xcelera-LVDS PX4 Full.

Execute the program via the Windows Start Menu shortcut **Start • Programs • DALSA • X64 Xcelera-LVDS PX4 Device Driver • Device Manager**. If the Device Manager program does not run, it will exit with a message that the board was not found. Since the X64 **Xcelera-LVDS** 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 right hand windowpane displays information for that item, as described below.



- Select **Information** to display identification and information stored in the X64 **Xcelera-LVDS** 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.

DALSA Log Viewer

An additional 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 at the prompt enter 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.

Memory Requirements with Area Scan Acquisitions

The X64 Xcelera-LVDS 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-LVDS 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-LVDS 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 6.10 or later:**
  When starting CamExpert, if there are no DALSA board 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 the card is properly inserted 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-LVDS 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 the camera is powered on.
- 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 requires one. Use the software trigger feature of CamExpert if you do not have a trigger source.
- Make certain that the camera cable is properly connected.
- Make certain that the camera is configured properly for the required 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, set the exposure to a longer period, and 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.
- A PCIe transfer issue sometimes causes this problem. 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 the **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-LVDS 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-LVDS PX4 installed in a PCI Express x16 slot?
  Note that some computer's x16 slot may only support non x16 boards at x1 speed or not at all. Check the computer documentation or test an X64 Xcelera-LVDS PX4 installation. Note that the X64 Xcelera-LVDS PX4 board does not function at x1 speeds.

CamExpert Quick Start

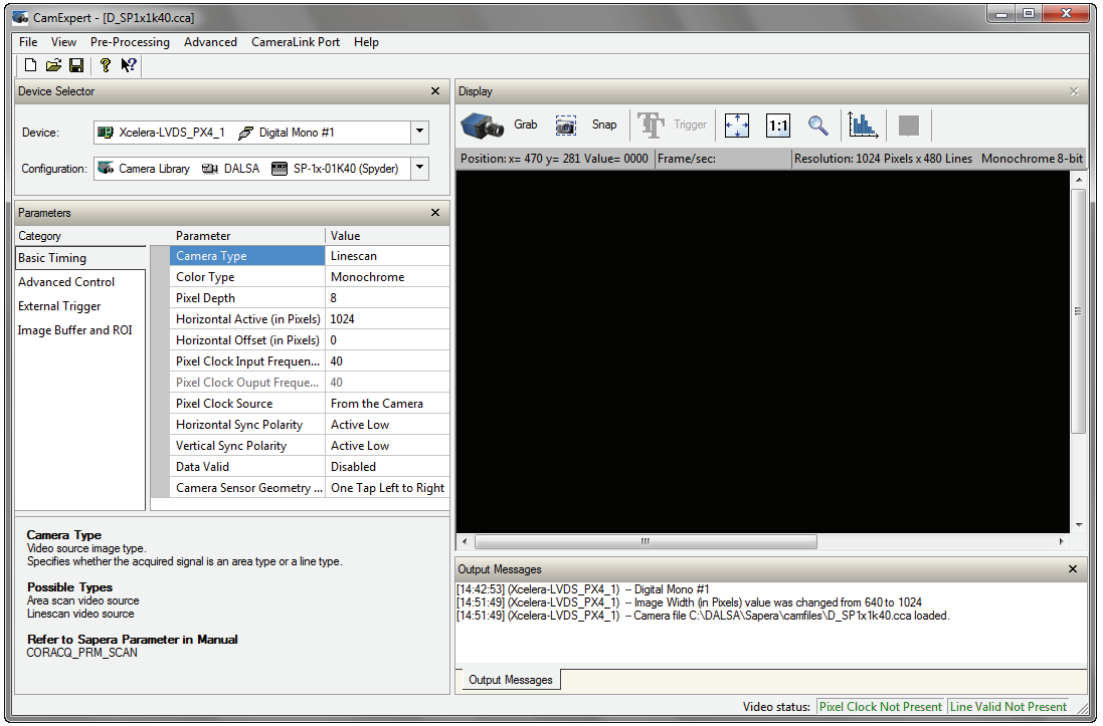
Interfacing Cameras with CamExpert

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-LVDS PX4 with common video signals, a camera file is required. Therefore, CamExpert is typically the first Sapera application run after an installation. Previously created .ccf files are easily usable with a new Sapera installation when copied into that system.

CamExpert Example with a Monochrome Camera

The image below shows CamExpert with the X64 Xcelera-LVDS PX4. The selected camera outputs monochrome 8-bit video. After selecting the camera model, the default timing parameters are displayed and the user can test by clicking on *Grab*. CamExpert descriptions follow the image.


CamExpert - [D_SP1x1k40.cca]
File View Pre-Processing Advanced CameraLink Port Help
Device Selector
Device: Xcelera-LVDS_PX4_1 Digital Mono #1
Configuration: Camera Library DALSA SP-1x-01K40 (Spyder)
Parameters
Category
Basic Timing
Advanced Control
External Trigger
Image Buffer and ROI
Parameter
Value
Camera Type
Linescan
Color Type
Monochrome
Pixel Depth
8
Horizontal Active (in Pixels)
1024
Horizontal Offset (in Pixels)
0
Pixel Clock Input Frequen...
40
Pixel Clock Ouput Freque...
40
Pixel Clock Source
From the Camera
Horizontal Sync Polarity
Active Low
Vertical Sync Polarity
Active Low
Data Valid
Disabled
Camera Sensor Geometry ...
One Tap Left to Right
Camera Type
Video source image type.
Specifies whether the acquired signal is an area type or a line type.
Possible Types
Area scan video source
Linescan video source
Refer to Sapera Parameter in Manual
CORACQ_PRM_SCAN
Display
Grab
Snap
Trigger
1:1
Position: x= 470 y= 281 Value= 0000
Frame/sec:
Resolution: 1024 Pixels x 480 Lines
Monochrome 8-bit
Output Messages
[14:42:53] (Xcelera-LVDS_PX4_1) -- Digital Mono #1
[14:51:49] (Xcelera-LVDS_PX4_1) -- Image Width (in Pixels) value was changed from 640 to 1024
[14:51:49] (Xcelera-LVDS_PX4_1) -- Camera file C:\DALSA\Sapera\camfiles\D_SP1x1k40.cca loaded.
Output Messages
Video status:
Pixel Clock Not Present
Line Valid Not Present

The CamExpert sections are:

- **Device:** Select which acquisition device to control and configure a camera file. Required in cases where there are multiple boards in a system and when one board supports multiple acquisition types. Note in this example, the X64 Xcelera-LVDS PX4 has firmware for monochrome or RGB cameras.
- **Camera:** Select the timing for a specific camera model included with the Sapera installation unless a new camera type is used. 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-LVDS PX4. 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, 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:** Window displays event and error messages. Displays camera connection status where green indicates signal present.

For context sensitive help click on the  button then click on a camera configuration parameter. A popup shows a short description of the configuration parameter. 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.

Camera Files Distributed with Sapera

The Sapera distribution CDROM includes camera files for a selection of X64 Xcelera-LVDS 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.

Browse our web site [www.dalsa.com/mv/support] for the latest information and application notes on X64 Xcelera-LVDS PX4 supported cameras.

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.dalsa.com/Public/Sapera/CamFile_Updates/]. Camera files are ASCII text and are read with Windows Notepad or any other text file reader, 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-LVDS PX4 does not have enough onboard memory for two frame buffers.

The X64 Xcelera-LVDS 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-LVDS 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. Note that the X64 Xcelera-LVDS 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 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), combine the camera (CCA) and frame grabber (CVI) parameters 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, and MONO16).
- Configuration of line/frame trigger parameters such as source (internal via the frame grabber /external via some outside event), electrical format (TTL, RS-422, 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 [www.dalsa.com/mv/support].
- Confirm that the correct version or board revision of X64 Xcelera-LVDS PX4 is used.
- 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, then modified to match timing and operating parameters for your camera, and lastly save them as Camera Configuration file (.ccf).
- Finally, if Sapera has no applicable camera file, 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 or from lens elements. That sensor when exposed to a uniformly lit field will have no gray level differences between pixels with calibrated flat field correction applied to the image. The CamExpert Flat Field tool functions with hardware supporting flat field processing.

X64 Xcelera-LVDS PX4 Flat Field Support

The X64 Xcelera-LVDS PX4 supports hardware based real-time Flat Field Correction with monochrome cameras.

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.

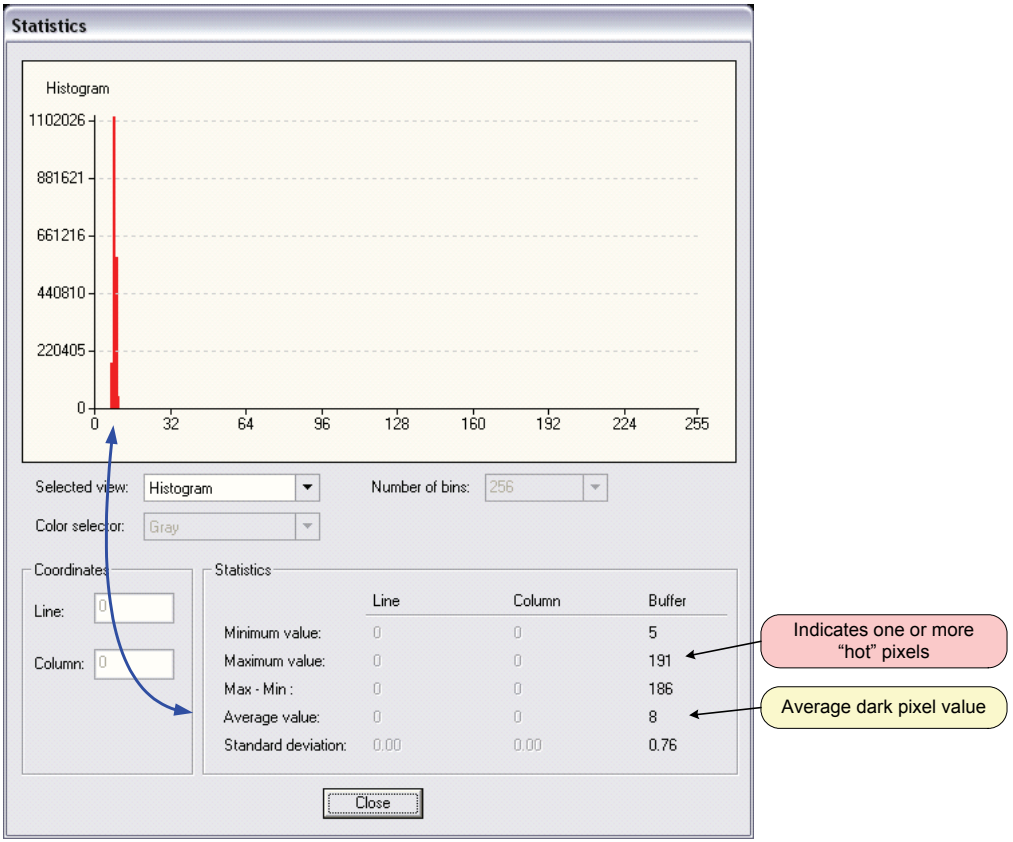
Also at this time, make preparations to grab a flat gray level image such as a clean evenly lit white wall or non-glossy paper. Note the lens iris position for a white but not saturated image. This white image is required for the calibration process.

Set up Dark and Bright Acquisitions with the Histogram Tool

Before performing calibration, verify the acquisition with a live grab. Also at this time, make preparations to grab a flat light gray level image, required for the calibration, such as a clean evenly lighted white wall or non-glossy paper with the lens slightly out of focus. Ideally, aim a controlled diffused light source directly at the lens. Note the lens iris position for a bright but not saturated image. Additionally check that the lens iris closes well or have a lens cover to grab the dark calibration image.

Verify a Dark Acquisition

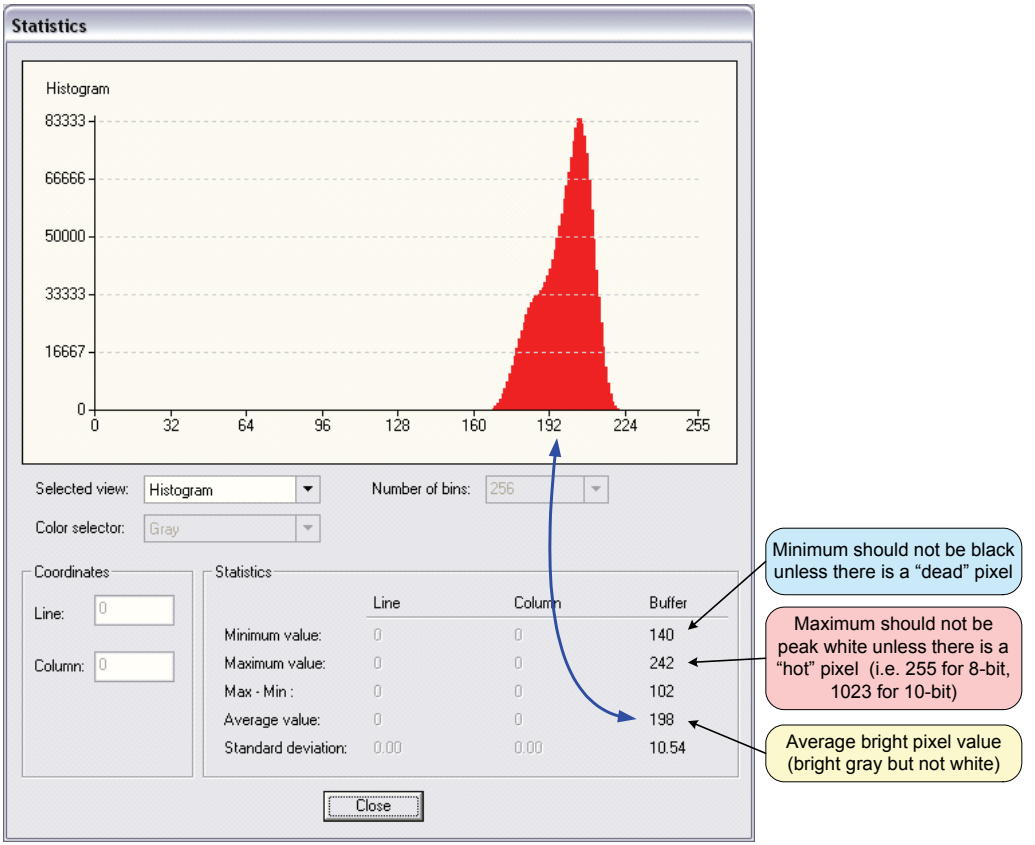
Close the camera lens iris and cover the lens with a lens cap. Using CamExpert, click on the grab button and then the histogram button. The following figure shows a typical histogram for a very dark image.



Important: In this example, the **average** pixel value for the frame is close to black. Also, note that most sensors will show a much higher maximum pixel value due to one or more "hot pixels". The sensor specification accounts for a small number of hot or stuck pixels (pixels that do not react to light over the full dynamic range specified for that sensor).

Verify a Bright Acquisition

Aim the camera at a diffused light source or evenly lit white wall with no shadows falling on it. Using CamExpert, click on the grab button and then the histogram button. Use the lens iris to adjust for a bright gray approximately around a pixel value of 200 (for 8-bit pixels). The following figure shows a typical histogram for a bright gray image.



Important: In this example, the **average** pixel value for the frame is bright gray. Also, note that sensors may show a much higher maximum or a much lower minimum pixel value due to one or more "hot or dead pixels". The sensor specification accounts for a small number of hot, stuck, or dead pixels (pixels that do not react to light over the full dynamic range specified for that sensor).

Once the bright gray acquisition setup is done, note the camera position and lens iris position to be able to repeat it during the calibration procedure.

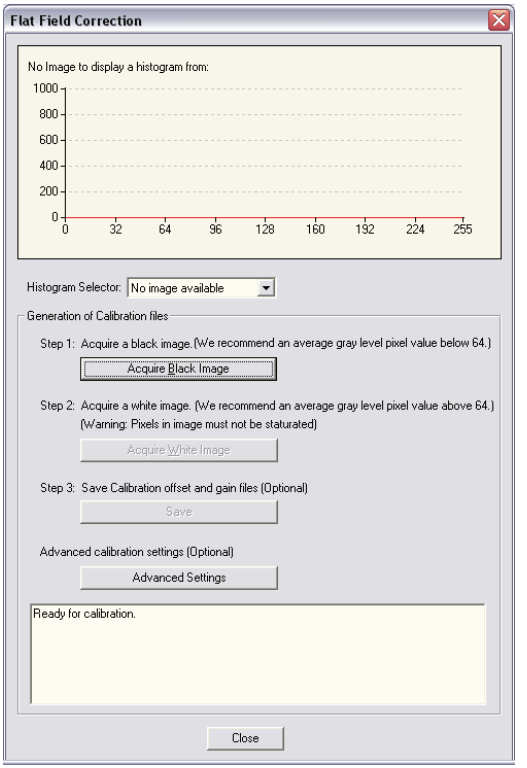
Flat Field Correction Calibration Procedure

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 sensor. Each captured 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:
Pre-Processing • 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, use the histogram tool to 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, select the image as the black reference.
- Setup the camera to acquire a uniform white image (but not saturated white). Use even illumination on white paper or a diffused light aimed at the camera, to acquire an image with a gray level of 128 minimum. It is preferable to prepare for the white level calibration before the calibration procedure as described in this manual.
- 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, select the image as the white reference.
- Click on **Save**. The flat field correction data is saved as a TIF image file with the 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.

Sapera Demo Applications

Grab Demo Overview

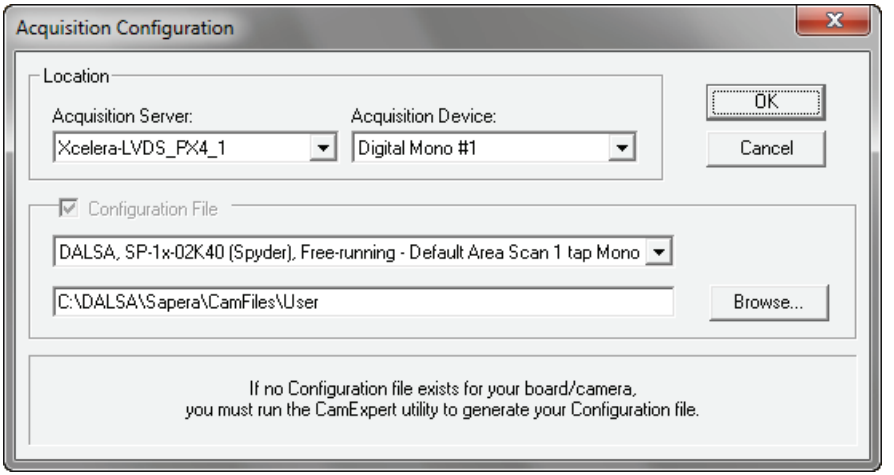
Program	**Start•Programs•DALSA•Sapera LT•Demos•Frame Grabbers•Grab Demo**
Program file	**\DALSA\Sapera\Demos\Classes\vc\GrabDemo\Release\GrabDemo.exe**
Workspace	**\DALSA\Sapera\Demos\Classes\vc\SapDemos.dsw**
.NET Solution	**\DALSA\Sapera\Demos\Classes\vc\SapDemos_2003.sln** **\DALSA\Sapera\Demos\Classes\vc\SapDemos_2005.sln** **\DALSA\Sapera\Demos\Classes\vc\SapDemos_2008.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, and 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.



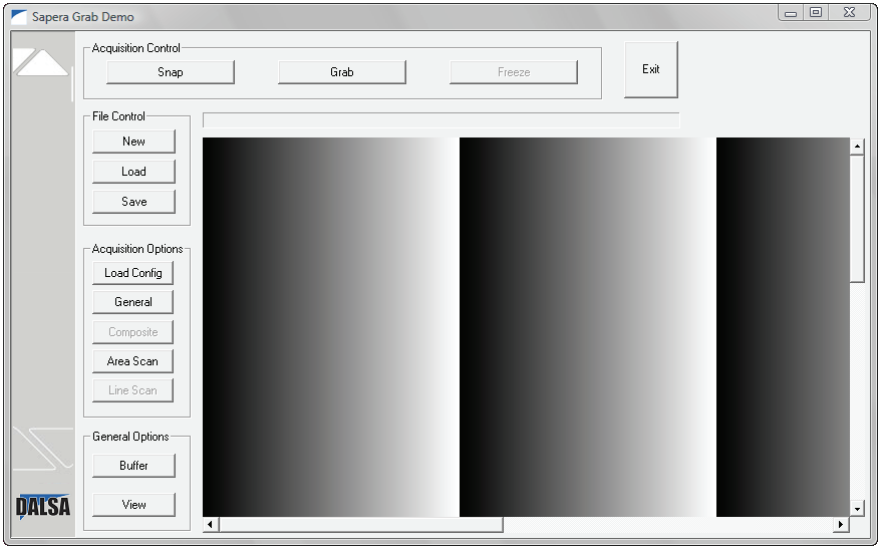
CCF File Selection

Use the acquisition configuration menu 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 the same 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

The Grab Demo program provides basic acquisition control for the selected frame grabber. The loaded camera file (.ccf) defines the frame buffer parameters.



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

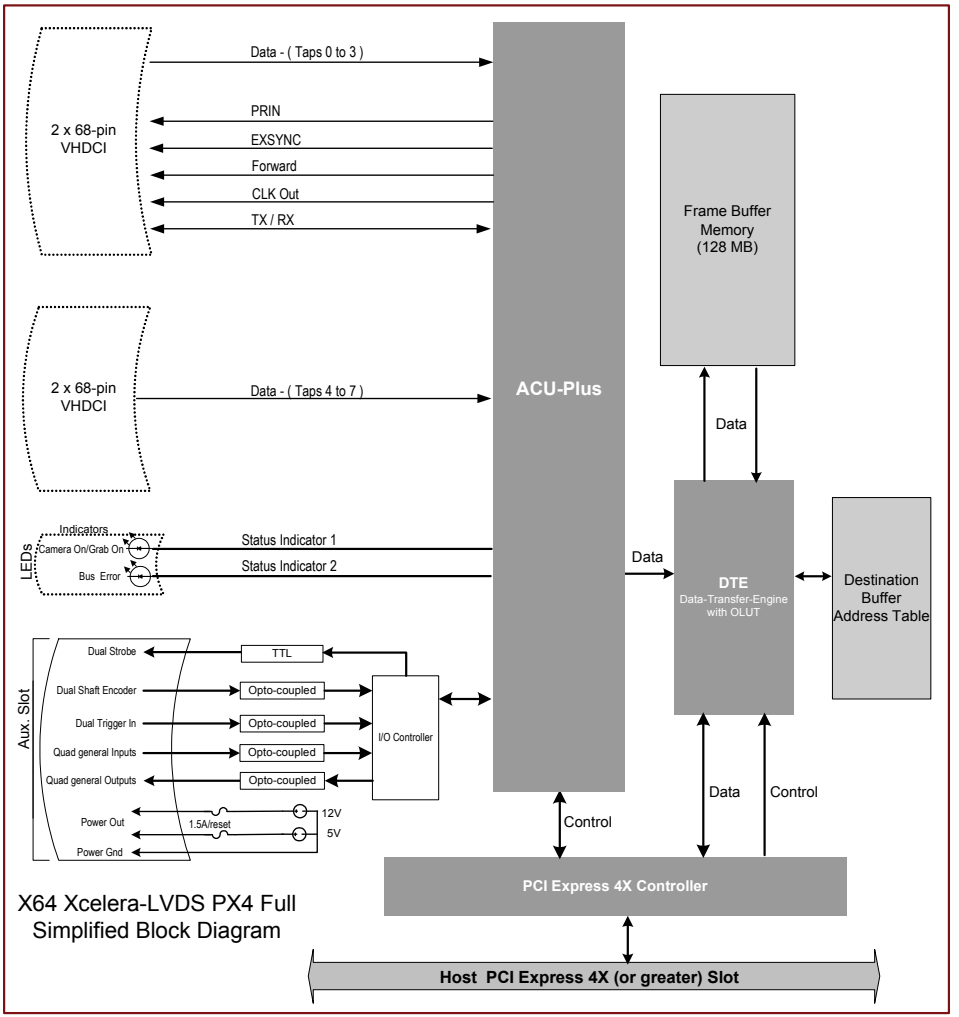
Program	**Start•Programs•DALSA•Sapera LT•Demos•Frame Grabbers•Flat Field Demo**
Program file	**\DALSA\Sapera\Demos\Classes\vc\FlatFieldDemo\Release\FlatfieldDemo.exe**
Workspace	**\DALSA\Sapera\Demos\Classes\vc\SapDemos.dsw**
Description	This program demonstrates Flat Field or Flat Line processing, either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

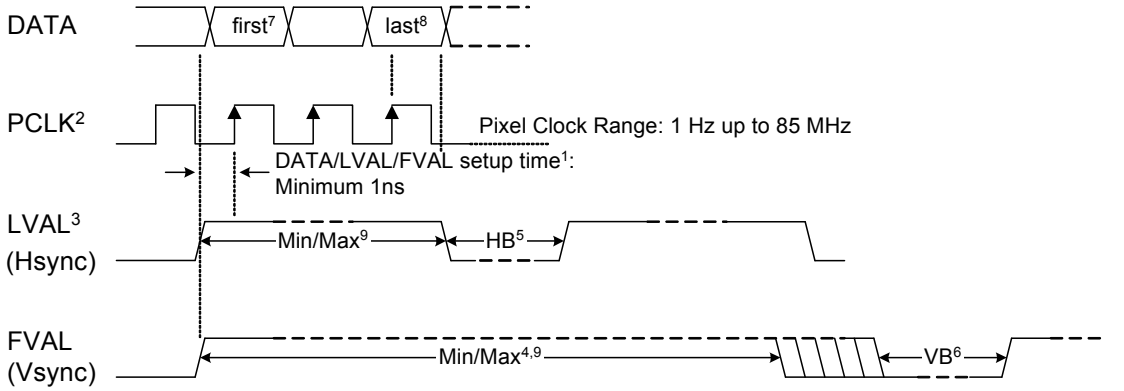
X64 Xcelera-LVDS PX4 Reference

Full Block Diagram



X64 Xcelera-LVDS PX4 Full Simplified Block Diagram

Acquisition Timing



- [1] The setup times for DATA, LVAL and FVAL are the same. All signals must be high and stable before the rising edge of the Pixel Clock. Sampling can also be done on the falling edge of the Pixel Clock when selecting the appropriate firmware.
- [2] Pixel Clock must always be present.
  Data is sampled on rising edge (default firmware) or on falling edge (optional firmware).
- [3] LVAL must be active high to acquire camera data.
- [4] Minimum of 1.
- [5] HB - Horizontal Blanking:

Minimum:	4 clock cycles
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 Line Scan Applications

Line Scan imaging applications require some form of external event trigger to synchronize Line Scan 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-LVDS 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-LVDS 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-LVDS PX4 configuration used:

PRM Value	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	Shaft Encoder Phase A	Shaft Encoder Phase A & B
1	Shaft Encoder Phase A	Shaft Encoder Phase A
2	Shaft Encoder Phase B	Shaft Encoder Phase B
3	n/a	n/a – use parameter value = 0

See "J3: External Signals Connector " on page 101 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

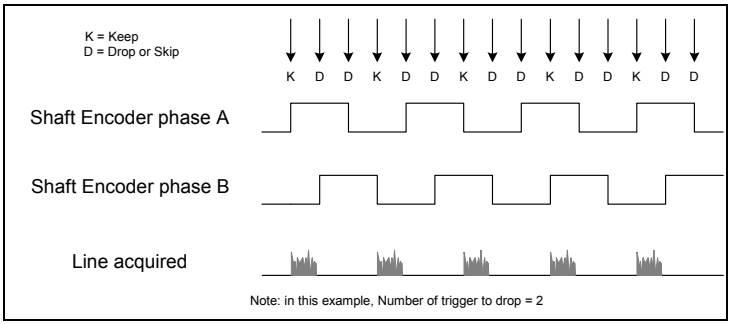
Shaft Encoder Interface Timing

Connector J3, Dual Balanced Shaft Encoder Inputs:

- Input 1: Pin 23 (Phase A +) & Pin 24 (Phase A -)
  (see "J3: External Signals Connector " on page 101 for complete connector signal details)
- Input 2: Pin 25 (Phase B +) & Pin 26 (Phase B -)
- See "External Signals Connector Bracket Assembly" on page 107 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 Line Scan camera. The X64 Xcelera-LVDS 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 Line Scan Cameras

When using Line Scan 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.

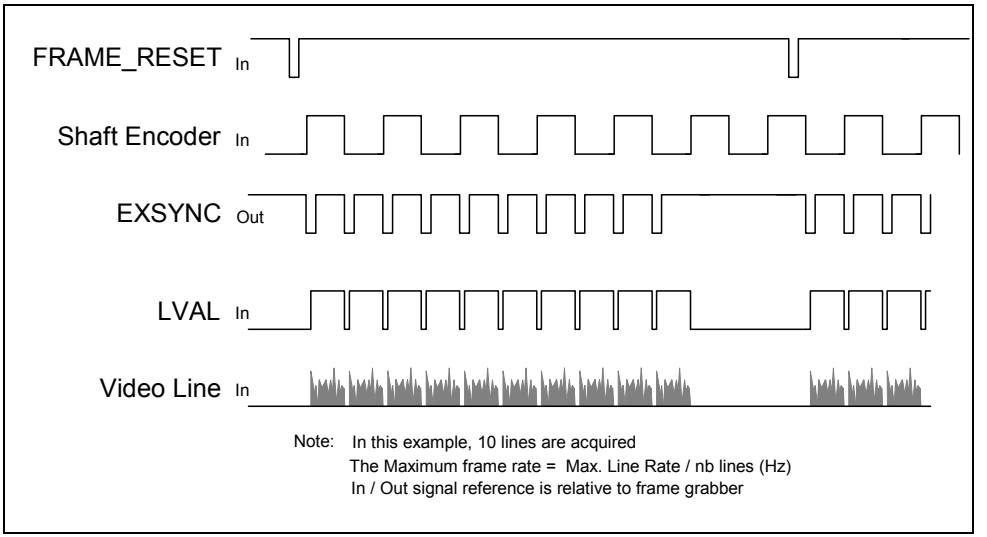
Virtual Frame_Reset Timing Diagram

The following timing diagram shows an example of grabbing 10 video lines from a Line Scan 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-LVDS PX4 trigger input.

- **FRAME_RESET** can be TTL or RS-422 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-LVDS PX4 via the Trigger In 1 balanced inputs on connector J3 pin 11 (+) and 12 (-).
- After the X64 Xcelera-LVDS 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-LVDS PX4 shaft encoder inputs (see "J3: External Signals Connector " on page 101 pinout) or an internal X64 Xcelera-LVDS 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 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 transfer to on-board memory failed. 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 is not sustainable 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-LVDS PX4 Output LUT availability.

Input Pixel Format (Bits)	Bits per Pixel (frame buffer format)	LUT Supported	Maximum TAPS Available with LUT
8	8 (mono-8)	Y	8
10	10 (mono-16)	Y	4
10	8 (mono-8)	Y	4
12	12 (mono-16)	Y	4
12	8 (mono-8)	Y	4
14	14 (mono-16)	N	-
16	16 (mono-16)	N	-
RGB 8	8 (RGB-8888)	Y	1
RGB 10	10 (RGB-101010)	Y	1
RGB 10	8 (RGB-8888)	Y	1
RGB 12	12 (RGB-16161616)	Y	1

X64 Xcelera-LVDS PX4 Supported Parameters

The tables below describe the Sapera capabilities supported by the X64 Xcelera-LVDS PX4 (i.e. default firmware is loaded), where acquisition device (0) is the monochrome, and acquisition device (1) is the color RGB.

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-LVDS PX4 family is described in Sapera as:

- Board Server: Xcelera-LVDS_PX4_1
- Acquisition modules available are (select one or the other):
  Acquisition Device (0) Digital Monochrome
  Acquisition Device (1) Digital Color RGB

Camera Related Capabilities

Capability	Values
CORACQ_CAP_CONNECTOR_TYPE	CORACQ_VAL_CONNECTOR_TYPE_CAM_CONTROL (0x4)
CORACQ_CAP_CONNECTOR_CAM_CONTROL (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_CAM_CONTROL (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)

Camera Related Parameters

Parameter		Values
CORACQ_PRM_CHANNEL	*monochrome and RGB*	CORACQ_VAL_CHANNEL_SINGLE (0x1)
	monochrome	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_VAL_VIDEO_RGB (0x8)
CORACQ_PRM_PIXEL_DEPTH	*digital monochrome mode*	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
	digital color RGB mode	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_COLORNI10
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_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_VAL_PIXEL_CLK_SRC_EXT_INT (0x4)
CORACQ_PRM_PIXEL_CLK_INT		min = 1 Hz, max = 85000000 Hz, step = 1 Hz
CORACQ_PRM_PIXEL_CLK_11		20000000 Hz
CORACQ_PRM_PIXEL_CLK_EXT		min = 1 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_VACTIVE		0 lines per field
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_VAL_LINE_INTEGRATE_METHOD_7 (0x40)
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	*monochrome*	min = 1 tap, max = 16 tap, step = 1 tap
	color RGB	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_TAP_5_DIRECTION	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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_VAL_FALLING_EDGE (0x8) * Value set has no effect on the board. Selection of the pixel clock detection is done by using the appropriate firmware configuration.
CORACQ_PRM_CHANNELS_ORDER		CORACQ_VAL_CHANNELS_ORDER_NORMAL (0x1) CORACQ_VAL_CHANNELS_ORDER_REVERSE (0x2)
CORACQ_PRM_LINESCAN_DIRECTION		Default = 1
CORACQ_PRM_LINESCAN_DIRECTION_POLARITY		CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
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_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_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_TAP_9_DIRECTION	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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	*monochrome only*	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_CAM_CONTROL_DURING_READOUT		TRUE FALSE
CORACQ_PRM_CAMERA_POWER		CORACQ_VAL_CAMERA_POWER_EXTERNAL (0x1)

VIC Related Parameters

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_DC_REST_MODE	CORACQ_VAL_DC_REST_MODE_AUTO (0x1)
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_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_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	*monochrome mode*	CORACQ_VAL_OUTPUT_FORMAT_MONO8 CORACQ_VAL_OUTPUT_FORMAT_MONO16
	color RGB mode	CORACQ_VAL_OUTPUT_FORMAT_RGB8888 CORACQ_VAL_OUTPUT_FORMAT_RGBP8 CORACQ_VAL_OUTPUT_FORMAT_RGBP16
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	*monochrome mode*	Default = CORACQ_VAL_OUTPUT_FORMAT_MONO8
	color RGB mode	Default = CORACQ_VAL_OUTPUT_FORMAT_RGBP16
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 (*)	CORACQ_VAL_LINESCAN_DIRECTION_FORWARD (0x1)
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_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_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_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_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_CAM_TRIGGER_DELAY	min = 0, max = 65535, step = 1
CORACQ_PRM_EXT_TRIGGER_DELAY_TIME_BASE	CORACQ_VAL_TIME_BASE_LINE (0x4)
CORACQ_PRM_EXT_TRIGGER_IGNORE_DELAY	min = 0 max = 65535000 step = 1

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] = Pulse to Board Sync
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

ACQ Related Parameters

Parameter	Value
CORACQ_PRM_LABEL	Digital Mono #1 Digital Color RGB #1
CORACQ_PRM_EVENT_TYPE	CORACQ_VAL_EVENT_TYPE_START_OF_FRAME (0x80000) CORACQ_VAL_EVENT_TYPE_END_OF_FRAME (0x800000) CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER (0x1000000) CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER2 (0x80) CORACQ_VAL_EVENT_TYPE_VERTICAL_SYNC (0x2000000) CORACQ_VAL_EVENT_TYPE_NO_PIXEL_CLK (0x40000000) CORACQ_VAL_EVENT_TYPE_PIXEL_CLK (0x80000000) CORACQ_VAL_EVENT_TYPE_FRAME_LOST (0x8000) CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW (0x4000) CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED (0x2000) CORACQ_VAL_EVENT_TYPE_VERTICAL_TIMEOUT (0x40)
CORACQ_PRM_SIGNAL_STATUS	CORACQ_VAL_SIGNAL_HSYNC_PRESENT (0x1) CORACQ_VAL_SIGNAL_VSYNC_PRESENT (0x2) CORACQ_VAL_SIGNAL_PIXEL_CLK_PRESENT (0x4)
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-LVDS PX4 does not have enough onboard memory for two frame buffers.

The X64 Xcelera-LVDS 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-LVDS 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-LVDS 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-LVDS PX4 Full board

Servers		Resources	
Name	**Type**	**Name**	**Index**
Xcelera-LVDS_PX4_1 (default firmware)	Acquisition	Digital Monochrome #1	0
	Acquisition	Digital Color RGB #1	1

Transfer Resource Locations

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

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

Technical Specifications

X64 Xcelera-LVDS PX4 Board Specifications

Card	Half-length PCIe x4 card, compliant with PCIe Rev. 1.1
Acquisition	Interfaces to LVDS (EIA-644) type cameras Up to 8-tap support Area scan and Line scan Pixel clock rate from 1 Hz up to 85 MHz
Feature	Compliant with DALSA Trigger-to-Image Reliability framework
Resolution	Horizontal size: 8 bytes to 256 Kbytes Vertical size: 1 line to infinite for Line Scan, 1 line to 16 million lines in area scan Support variable frame length (up to 16 million lines)
Synchronization Minimums	Horizontal Sync minimum: 4 pixels Vertical Sync minimum: 1 line
Base Board Memory	128 MB
Scanning	Integrated advanced tap reversal engine allows independent tap formatting: Progressive, multi-tap, multi-channel, tap reversal.
Max Data Rate	640 MB/sec 12/14/16-bit, 4 taps @ 85MHz any tap configuration 8-bit, 8 taps at 85 MHz at any tap configuration
Pixel Format	Support LVDS tap configuration for: 8, 10, 12, 14 and 16-bit mono 24, 30, 36, 42, 48-bit RGB
Basic Post-processing	LUT available (see LUT Availability for details): 1 x 8/10/12-bit in — 8/10/12-bit out monochrome 3 x 8/10/12-bit in — 8/10/12-bit out for RGB Real-time Flat-field/ flat-line (shading) correction supported in hardware. Compensates for sensor defects such as FPN, PRNU, defective pixels and variations between pixels due to the light refraction through a lens (Shading effect).
Controls	Comprehensive event notification Timing control logic for EXSYNC, PRIN and strobe signals External trigger latency less than 1 µsec. Serial port (9600 to 115 kbps)
Connectors	Honda-68 for camera, board trigger, strobe and serial port DB25 or DB37 option for opto-coupled GIOs, board trigger, strobe

I/Os	Opto-coupled dual-phase quadrature shaft encoder (TTL/RS-422 with a maximum input frequency of 200 KHz) 1 opto-coupled external trigger inputs (5V/24V) 1 strobe outputs (5V) 4 opto-coupled general inputs (5V/24V) 4 general outputs X/IO Interface (optional)
Software	Supported by Sapera LT Microsoft Windows XP 32/64-bit, Vista 32/64-bit
System Requirements	Intel Pentium IV class CPU, 1 GB system memory, 20 MB free hard-drive space, one free PCIe x4 slot.
Board Dimensions	Approximately 6.5 in. (16.6 cm) wide by 4 in. (10 cm) high

Host System Requirements

General System Requirements for the X64 Xcelera-LVDS PX4

- PCI Express x4 slot or x8 slot compatible
- Note: There is no assumption made that the X64 Xcelera-LVDS PX4 will function when installed in a x16 slot. Direct testing by the user is required.

Operating System Support

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)

Power Requirements

+3.3V:	1.92A (standby) 2.64A (during acquisition)
+12V:	0.72A (standby) 0.81A (during acquisition)

EMI Certifications

Class B, both FCC and CE

EC & FCC DECLARATION OF CONFORMITY

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

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

FRAME GRABBER BOARD: **Xcelera-LVDS PX4**

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

EN55022:1998- Residential, Commercial and Light Industry
ENV50204: 1995
EN61000-3 & -4: 1994, 1995, 1996, 1998, 2000, 2001
EN60255-22-4: 2002

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

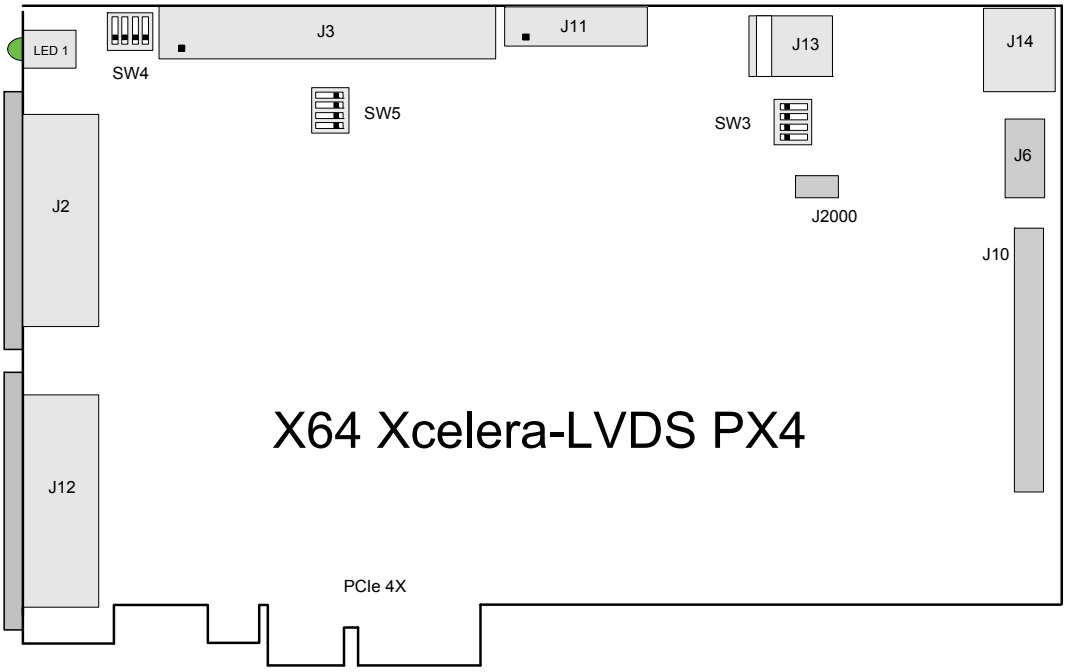
St. Laurent, Canada — Location
2008/10/03 — Date

Eric Carey
Director,
Research & Development

Connector and Switch Locations

Refer to the board layout to locate connectors or configuration switches.

X64 Xcelera-LVDS PX4 Board Layout Drawing



Connector, Switch, Jumper Description List

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

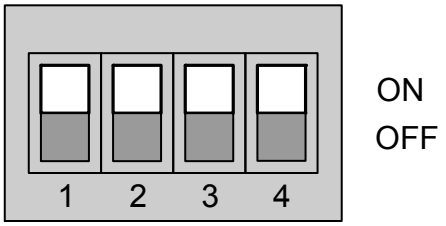
Location	Description	Location	Description
J2	LVDS connector 3 and 4	J11	X-I/O Module Interface
J12	LVDS connector 1 and 2	J13	PC power to camera interface.
J3	External Signals connector	J6, J10, J2000	Reserved
J14	Multi Board Sync	SW4, SW5, SW3	Configuration micro-switches

Connector and Switch Specifications

The following section provides the technical reference for user set configuration switches and camera connection details.

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

SW4: General Inputs Signal Logic Threshold Level

For each general input, select the threshold voltage detected as a logic high signal. See "Note 1: General Inputs Specifications" on page 102.

SW4 Switch Number	Assigned to	OFF Position (TTL & low volatage differential)	ON Position (default for 24V systems)
1	general input 1	Logic Threshold at ~2 volts (preferred for differential signals)	Logic Threshold at ~10 volts
2	general input 2		
3	general input 3		
4	general input 4		

SW5: 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 103.

SW5 Switch Number	Assigned to	OFF Position (TTL & low volatage differential)	ON Position (for 24V systems)
1	trigger input 1	Logic Threshold at ~2 volts (preferred for differential signals)	Logic Threshold at ~10 volts
2	trigger input 2		
3	NA		
4	NA		

SW3: Normal/Safe Boot Mode & GEN2 Slot Workaround

The X64 Xcelera-LVDS 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	GEN2 Slot Workaround	Disabled	Enabled
3	reserved		
4	reserved		

SW3-1 Boot Mode Details

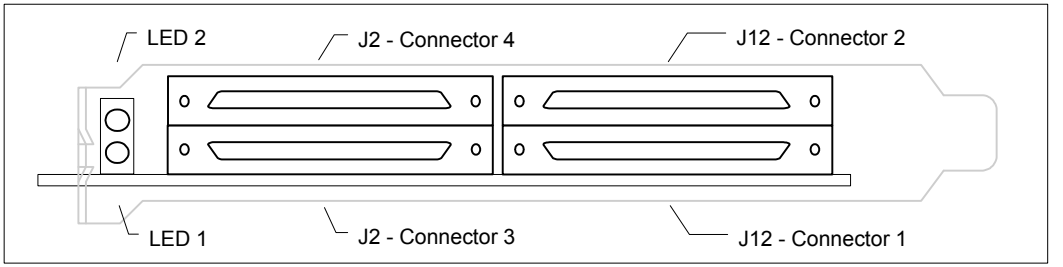
- Normal Mode: Board powers up in the normal operating mode.
- Safe Mode: With the computer off, move the switch to the ON position. This mode is required if any problems occurred while updating firmware. With the switch in the ON position, power on the computer and update the firmware again. After the update completes power off the computer and move the switch to the OFF position. Power on the computer once again to use the Xcelera-LVDS PX4 normally. (See "Recovering from a Firmware Update Error" on page 39).

SW3-2 GEN2 Slot Workaround Details

- **Normal Mode:** Normal operation of the Xcelera-LVDS PX4.
- **GEN2 Slot Workaround:** In computers with GEN2 slots and the Intel 5400 chipset. there have been circumstances where the board is not detected properly. This issue is identified by the status LED 2 that keeps on flashing red at boot time. In one example, with a Dell T5400 or T7400 computer, the following message was displayed by the computer BIOS:"Alert! Error initializing PCI Express slot".

- Therefore when using such a computer, with the Xcelera SW3-2 in the ON position, the computer should boot normally and the Xcelera should function. If this is not the case, please contact Technical Support ("Contact Information" on page 127) with details about your computer.
- Note: ECO 4252 is needed for SW3-2 to be functional.

X64 Xcelera-LVDS PX4 End Bracket View



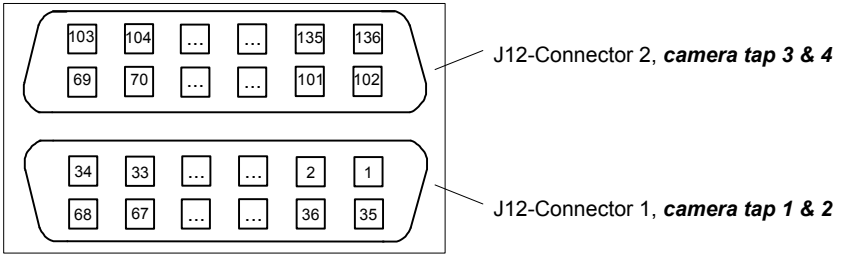
- Connector #1 is for camera Taps 1 and 2
- Connector #2 is for camera Taps 3 and 4
- Connector #3 is for camera Taps 5 and 6
- Connector #4 is for camera Taps 7 and 8

Status LEDs Functional Description

Status LED 1:	
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:	
Off	Board initialization good
Red	Board initialization error on computer boot-up

J12: Dual 68 Pin VHDCI Connectors

Camera interface J12 consists of two VHDCI connectors (refer to the connector bracket view "X64 Xcelera-LVDS PX4 End Bracket View" on page 87). Connector #1 is for camera taps 1 and 2, while connector #2 is for camera taps 3 and 4.



All signals support RS-644 (LVDS)

- High voltage/Low voltage minimum differential threshold = 100mV
- Maximum common mode voltage = 5 V
- Maximum Input Current = ± 10mA
- Typical connector part number: Honda HDRA-E68W1LFDT1EC-SL+

J12 – Connector 1: Monochrome Tap 1 & 2 Pinout

Camera taps #1 and #2.

J12 – Connector 1 VHDCI Pin Number	Name	Type	Description
1	DINa_0+	Input	Bit 0 + (Tap 1)
35	DINa_0-	Input	Bit 0 - (Tap 1)
2	DINa_1+	Input	Bit 1 + (Tap 1)
36	DINa_1-	Input	Bit 1 - (Tap 1)
3	DINa_2+	Input	Bit 2 + (Tap 1)
37	DINa_2-	Input	Bit 2 - (Tap 1)
4	DINa_3+	Input	Bit 3 + (Tap 1)
38	DINa_3-	Input	Bit 3 - (Tap 1)
5	DINa_4+	Input	Bit 4 + (Tap 1)
39	DINa_4-	Input	Bit 4 - (Tap 1)
6	DINa_5+	Input	Bit 5 + (Tap 1)
40	DINa_5-	Input	Bit 5 - (Tap 1)
7	DINa_6+	Input	Bit 6 + (Tap 1)

41	DINa_6-	Input	Bit 6 - (Tap 1)
8	DINa_7+	Input	Bit 7 + (Tap 1)
42	DINa_7-	Input	Bit 7 - (Tap 1)
9, 43	GND		Ground
10	DINb_0+	Input	Bit 0 + (Tap 2)
44	DINb_0-	Input	Bit 0 - (Tap 2)
11	DINb_1+	Input	Bit 1 + (Tap 2)
45	DINb_1-	Input	Bit 1 - (Tap 2)
12	DINb_2+	Input	Bit 2 + (Tap 2)
46	DINb_2-	Input	Bit 2 - (Tap 2)
13	DINb_3+	Input	Bit 3 + (Tap 2)
47	DINb_3-	Input	Bit 3 - (Tap 2)
14	DINb_4+	Input	Bit 4 + (Tap 2)
48	DINb_4-	Input	Bit 4 - (Tap 2)
15	DINb_5+	Input	Bit 5 + (Tap 2)
49	DINb_5-	Input	Bit 5 - (Tap 2)
16	DINb_6+	Input	Bit 6 + (Tap 2)
50	DINb_6-	Input	Bit 6 - (Tap 2)
17	DINb_7+	Input	Bit 7 + (Tap 2)
51	DINb_7-	Input	Bit 7 - (Tap 2)
18, 52	GND		Ground
19	PHA+	Input	Shaft Encoder Phase A +
53	PHA-	Input	Shaft Encoder Phase A -
20	PHB+	Input	Shaft Encoder Phase B +
54	PHB-	Input	Shaft Encoder Phase B -
21	Trig1+	Input	External Trigger 1 +
55	Trig1-	Input	External Trigger 1 -
22	Out12V	Output	12 Volt Source (fused – power off reset)
56	Out12V	Output	12 Volt Source (fused – power off reset)
23	cam CC2+	Output (default: PRIN)	Programmable Camera Control 2 +
57	cam CC2-	Output	Programmable Camera Control 2 -
24	cam CC1+	Output (default: EXSYNC)	Programmable Camera Control 1 +
58	cam CC1-	Output	Programmable Camera Control 1 -
25	cam CC3+	Output (default: Forward)	Programmable Camera Control 3 +

59	cam CC3-	Output	Programmable Camera Control 3 -
26	STROBE1	Output	Strobe control 1 (TTL)
60		Output	Reserved
27, 61	GND		Ground
28	cam LVAL+	Input	Camera Line Valid +
62	cam LVAL-	Input	Camera Line Valid -
29	cam FVAL+	Input	Camera Frame Valid +
63	cam FVAL-	Input	Camera Frame Valid -
30	cam DVAL+	Input	Camera Data Valid +
64	cam DVAL-	Input	Camera Data Valid -
31	cam PCLK+	Input	Camera Pixel Clock In +
65	cam PCLK-	Input	Camera Pixel Clock In -
32	RS-232-Tx	Serial Out	
66			Reserved
33	cam CC4+	Output (default: Pixel clk)	Programmable Camera Control 4 +
67	cam CC4-	Output	Programmable Camera Control 4 -
34	RS-232-Rx	Serial In	
68			Reserved

J12 – Connector 2: Monochrome Tap 3 & 4 Pinout

Camera taps #3 and #4.

J12 – Connector 2 VHDCI Pin Number	Name	Type	Description
69	DINc_0+	Input	Bit 0 + (Tap 3)
103	DINc_0-	Input	Bit 0 - (Tap 3)
70	DINc_1+	Input	Bit 1 + (Tap 3)
104	DINc_1-	Input	Bit 1 - (Tap 3)
71	DINc_2+	Input	Bit 2 + (Tap 3)
105	DINc_2-	Input	Bit 2 - (Tap 3)
72	DINc_3+	Input	Bit 3 + (Tap 3)
106	DINc_3-	Input	Bit 3 - (Tap 3)
73	DINc_4+	Input	Bit 4 + (Tap 3)
107	DINc_4-	Input	Bit 4 - (Tap 3)
74	DINc_5+	Input	Bit 5 + (Tap 3)

108	DINc_5-	Input	Bit 5 - (Tap 3)
75	DINc_6+	Input	Bit 6 + (Tap 3)
109	DINc_6-	Input	Bit 6 - (Tap 3)
76	DINc_7+	Input	Bit 7 + (Tap 3)
110	DINc_7-	Input	Bit 7 - (Tap 3)
77, 111	GND		Ground
78	DINd_0+	Input	Bit 0 + (Tap 4)
112	DINd_0-	Input	Bit 0 - (Tap 4)
79	DINd_1+	Input	Bit 1 + (Tap 4)
113	DINd_1-	Input	Bit 1 - (Tap 4)
80	DINd_2+	Input	Bit 2 + (Tap 4)
114	DINd_2-	Input	Bit 2 - (Tap 4)
81	DINd_3+	Input	Bit 3 + (Tap 4)
115	DINd_3-	Input	Bit 3 - (Tap 4)
82	DINd_4+	Input	Bit 4 + (Tap 4)
116	DINd_4-	Input	Bit 4 - (Tap 4)
83	DINd_5+	Input	Bit 5 + (Tap 4)
117	DINd_5-	Input	Bit 5 - (Tap 4)
84	DINd_6+	Input	Bit 6 + (Tap 4)
118	DINd_6-	Input	Bit 6 - (Tap 4)
85	DINd_7+	Input	Bit 7 + (Tap 4)
119	DINd_7-	Input	Bit 7 - (Tap 4)
86, 120	GND		Ground
87			Reserved
121			Reserved
88			Reserved
122			Reserved
89			Reserved
123			Reserved
90			Reserved
124			Reserved
91			Reserved
125			Reserved
92			Reserved
126			Reserved
93			Reserved
127			Reserved

94			Reserved
128			Reserved
95, 129	GND		Ground
96			Reserved
130			Reserved
97			Reserved
131			Reserved
98			Reserved
132			Reserved
99			Reserved
133			Reserved
100			Reserved
134			Reserved
101			Reserved
135			Reserved
102			Reserved
136			Reserved

J12 – Connector 1: RGB-24 & RGB-30 Pinout

Pin #	RGB-24	Type		RGB-30	Type
1	BLUE_0+ (LSB)	In		BLUE_0+ (LSB)	In
35	BLUE_0- (LSB)	In		BLUE_0- (LSB)	In
2	BLUE_1+	In		BLUE_1+	In
36	BLUE_1-	In		BLUE_1-	In
3	BLUE_2+	In		BLUE_2+	In
37	BLUE_2-	In		BLUE_2-	In
4	BLUE_3+	In		BLUE_3+	In
38	BLUE_3-	In		BLUE_3-	In
5	BLUE_4+	In		BLUE_4+	In
39	BLUE_4-	In		BLUE_4-	In
6	BLUE_5+	In		BLUE_5+	In
40	BLUE_5-	In		BLUE_5-	In
7	BLUE_6+	In		BLUE_6+	In
41	BLUE_6-	In		BLUE_6-	In
8	BLUE_7+ (MSB)	In		BLUE_7+	In
42	BLUE_7- (MSB)	In		BLUE_7-	In
9, 43	GND			GND	
10	GREEN_0+	In		BLUE_8+	In
44	GREEN_0-	In		BLUE_8-	In
11	GREEN_1+	In		BLUE_9+ (MSB)	In
45	GREEN_1-	In		BLUE_9- (MSB)	In
12	GREEN_2+	In		GREEN_0+	In
46	GREEN_2-	In		GREEN_0-	In
13	GREEN_3+	In		GREEN_1+	In
47	GREEN_3-	In		GREEN_1-	In
14	GREEN_4+	In		GREEN_2+	In
48	GREEN_4-	In		GREEN_2-	In
15	GREEN_5+	In		GREEN_3+	In
49	GREEN_5-	In		GREEN_3-	In
16	GREEN_6+	In		GREEN_4+	In
50	GREEN_6-	In		GREEN_4-	In
17	GREEN_7+	In		GREEN_5+	In
51	GREEN_7-	In		GREEN_5-	In

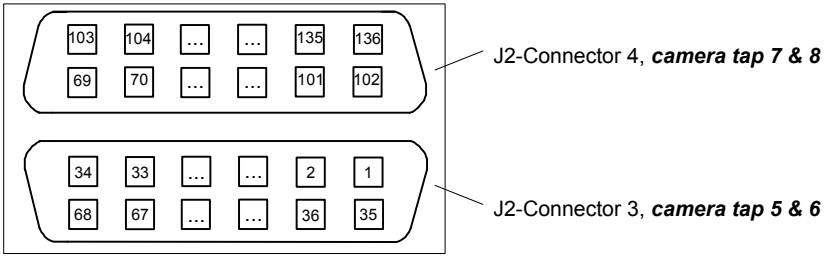
18, 52	GND			GND	
19	Shaft encoder phase A +	In		Shaft encoder phase A +	In
53	Shaft encoder phase A -	In		Shaft encoder phase A -	In
20	Shaft encoder phase B +	In		Shaft encoder phase B +	In
54	Shaft encoder phase B -	In		Shaft encoder phase B -	In
21	External Trigger 1 +	In		External Trigger 1 +	In
55	External Trigger 1 -	In		External Trigger 1 -	In
22	12Vout (fused – power off reset)	Out		12Vout (fused – power off reset)	Out
56	12Vout (fused – power off reset)	Out		12Vout (fused – power off reset)	Out
23	CC2+	Out		CC2+	Out
57	CC2-	Out		CC2-	Out
24	CC1+	Out		CC1+	Out
58	CC1-	Out		CC1-	Out
25	CC3+	Out		CC3+	Out
59	CC3-	Out		CC3-	Out
26	Strobe Control 1	Out		Strobe Control 1	Out
60	Strobe Control 2	Out		Strobe Control 2	Out
27, 61	GND			GND	
28	LVAL+	In		LVAL+	In
62	LVAL -	In		LVAL -	In
29	FVAL +	In		FVAL +	In
63	FVAL -	In		FVAL -	In
30	DVAL +	In		DVAL +	In
64	DVAL -	In		DVAL -	In
31	PCLK+	In		PCLK+	In
65	PCLK-	In		PCLK-	In
32	RS-232 Rx (revision B board)	In		RS-232 Rx (revision B board)	In
66	reserved			reserved	
33	CC4+	Out		CC4+	Out
67	CC4-	Out		CC4-	Out
34	RS-232 Tx (revision B board)	Out		RS-232 Tx (revision B board)	Out
68	reserved			reserved	

J12 – Connector 2: RGB-24 & RGB-30 Pinout

Pin #	RGB-24	Type		RGB-30	Type
69	RED_0+ (LSB)	In		GREEN_6+	In
103	RED_0- (LSB)	In		GREEN_6-	In
70	RED_1+	In		GREEN_7+	In
104	RED_1-	In		GREEN_7-	In
71	RED_2+	In		GREEN_8+	In
105	RED_2-	In		GREEN_8-	In
72	RED_3+	In		GREEN_9+	In
106	RED_3-	In		GREEN_9-	In
73	RED_4+	In		RED_0+ (LSB)	In
107	RED_4-	In		RED_0- (LSB)	In
74	RED_5+	In		RED_1+	In
108	RED_5-	In		RED_1-	In
75	RED_6+	In		RED_2+	In
109	RED_6-	In		RED_2-	In
76	RED_7+ (MSB)	In		RED_3+	In
110	RED_7- (MSB)	In		RED_3-	In
77, 111	GND			GND	
78	reserved			RED_4+	In
112	reserved			RED_4-	In
79	reserved			RED_5+	In
113	reserved			RED_5-	In
80	reserved			RED_6+	In
114	reserved			RED_6-	In
81	reserved			RED_7+	In
115	reserved			RED_7-	In
82	reserved			RED_8+	In
116	reserved			RED_8+	In
83	reserved			RED_9+ (MSB)	In
117	reserved			RED_9+ (MSB)	In
86, 120	GND			GND	
95, 129	GND			GND	
	All remaining pins are Reserved			All remaining pins are Reserved	

J2: Dual 68 Pin VHDCI Connectors

Camera interface J2 consists of two VHDCI connectors (refer to the X64-LVDS connector bracket view "X64 Xcelera-LVDS PX4 End Bracket View" on page 87). Connector #3 is for camera taps 5 and 6, while connector #4 is for camera taps 7 and 8.



All signals support RS-644 (LVDS)

- High voltage/Low voltage minimum differential threshold = 100mV
- Maximum common mode voltage = 5 V
- Maximum Input Current = ± 10mA
- Typical connector part number: Honda HDRA-E68W1LFDT1EC-SL+

J2 – Connector 3: Monochrome Tap 5 & 6 Pinout

Camera taps #5 and #6.

J2 – Connector 3 VHDCI Pin Number	Name	Type	Description
1	DINe_0+	Input	Bit 0 + (Tap 5)
35	DINe_0-	Input	Bit 0 - (Tap 5)
2	DINe_1+	Input	Bit 1 + (Tap 5)
36	DINe_1-	Input	Bit 1 - (Tap 5)
3	DINe_2+	Input	Bit 2 + (Tap 5)
37	DINe_2-	Input	Bit 2 - (Tap 5)
4	DINe_3+	Input	Bit 3 + (Tap 5)
38	DINe_3-	Input	Bit 3 - (Tap 5)
5	DINe_4+	Input	Bit 4 + (Tap 5)
39	DINe_4-	Input	Bit 4 - (Tap 5)
6	DINe_5+	Input	Bit 5 + (Tap 5)
40	DINe_5-	Input	Bit 5 - (Tap 5)

7	DINe_6+	Input	Bit 6 + (Tap 5)
41	DINe_6-	Input	Bit 6 - (Tap 5)
8	DINe_7+	Input	Bit 7 + (Tap 5)
42	DINe_7-	Input	Bit 7 - (Tap 5)
9, 43	GND		Ground
10	DINf_0+	Input	Bit 0 + (Tap 6)
44	DINf_0-	Input	Bit 0 - (Tap 6)
11	DINf_1+	Input	Bit 1 + (Tap 6)
45	DINf_1-	Input	Bit 1 - (Tap 6)
12	DINf_2+	Input	Bit 2 + (Tap 6)
46	DINf_2-	Input	Bit 2 - (Tap 6)
13	DINf_3+	Input	Bit 3 + (Tap 6)
47	DINf_3-	Input	Bit 3 - (Tap 6)
14	DINf_4+	Input	Bit 4 + (Tap 6)
48	DINf_4-	Input	Bit 4 - (Tap 6)
15	DINf_5+	Input	Bit 5 + (Tap 6)
49	DINf_5-	Input	Bit 5 - (Tap 6)
16	DINf_6+	Input	Bit 6 + (Tap 6)
50	DINf_6-	Input	Bit 6 - (Tap 6)
17	DINf_7+	Input	Bit 7 + (Tap 6)
51	DINf_7-	Input	Bit 7 - (Tap 6)
18, 52	GND		Ground
19	PHA+	Input	Shaft Encoder Phase A +
53	PHA-	Input	Shaft Encoder Phase A -
20	PHB+	Input	Shaft Encoder Phase B +
54	PHB-	Input	Shaft Encoder Phase B -
21	Reserved	Input	Reserved
55	Reserved	Input	Reserved
22	Out12V	Output	12 Volt Source (fused – power off reset)
56	Out12V	Output	12 Volt Source (fused – power off reset)
23		Output	Reserved
57		Output	Reserved
24		Output	Reserved
58		Output	Reserved
25		Output	Reserved
59		Output	Reserved
26	STROBE1	Output	Strobe control 1 (TTL)

60		Output	Reserved
27, 61	GND		Ground
28		Input	Reserved
62		Input	Reserved
29		Input	Reserved
63		Input	Reserved
30		Input	Reserved
64		Input	Reserved
31		Input	Reserved
65		Input	Reserved
32			Reserved
66			Reserved
33		Output	Reserved
67		Output	Reserved
34			Reserved
68			Reserved

J2 – Connector 4: Monochrome Tap 7 & 8 Pinout

Camera taps #7 and #8.

J2 – Connector 4 VHDCI Pin Number	Name	Type	Description
69	DINg_0+	Input	Bit 0 + (Tap 7)
103	DINg_0-	Input	Bit 0 - (Tap 7)
70	DINg_1+	Input	Bit 1 + (Tap 7)
104	DINg_1-	Input	Bit 1 - (Tap 7)
71	DINg_2+	Input	Bit 2 + (Tap 7)
105	DINg_2-	Input	Bit 2 - (Tap 7)
72	DINg_3+	Input	Bit 3 + (Tap 7)
106	DINg_3-	Input	Bit 3 - (Tap 7)
73	DINg_4+	Input	Bit 4 + (Tap 7)
107	DINg_4-	Input	Bit 4 - (Tap 7)
74	DINg_5+	Input	Bit 5 + (Tap 7)
108	DINg_5-	Input	Bit 5 - (Tap 7)
75	DINg_6+	Input	Bit 6 + (Tap 7)
109	DINg_6-	Input	Bit 6 - (Tap 7)

76	DINg_7+	Input	Bit 7 + (Tap 7)
110	DINg_7-	Input	Bit 7 - (Tap 7)
77, 111	GND		Ground
78	DINh_0+	Input	Bit 0 + (Tap 8)
112	DINh_0-	Input	Bit 0 - (Tap 8)
79	DINh_1+	Input	Bit 1 + (Tap 8)
113	DINh_1-	Input	Bit 1 - (Tap 8)
80	DINh_2+	Input	Bit 2 + (Tap 8)
114	DINh_2-	Input	Bit 2 - (Tap 8)
81	DINh_3+	Input	Bit 3 + (Tap 8)
115	DINh_3-	Input	Bit 3 - (Tap 8)
82	DINh_4+	Input	Bit 4 + (Tap 8)
116	DINh_4-	Input	Bit 4 - (Tap 8)
83	DINh_5+	Input	Bit 5 + (Tap 8)
117	DINh_5-	Input	Bit 5 - (Tap 8)
84	DINh_6+	Input	Bit 6 + (Tap 8)
118	DINh_6-	Input	Bit 6 - (Tap 8)
85	DINh_7+	Input	Bit 7 + (Tap 8)
119	DINh_7-	Input	Bit 7 - (Tap 8)
86, 120	GND		Ground
87			Reserved
121			Reserved
88			Reserved
122			Reserved
89			Reserved
123			Reserved
90			Reserved
124			Reserved
91			Reserved
125			Reserved
92			Reserved
126			Reserved
93			Reserved
127			Reserved
94			Reserved
128			Reserved
95, 129	GND		Ground

96			Reserved
130			Reserved
97			Reserved
131			Reserved
98			Reserved
132			Reserved
99			Reserved
133			Reserved
100			Reserved
134			Reserved
101			Reserved
135			Reserved
102			Reserved
136			Reserved

J3: External Signals Connector

J3 Pin Header Numbering Detail

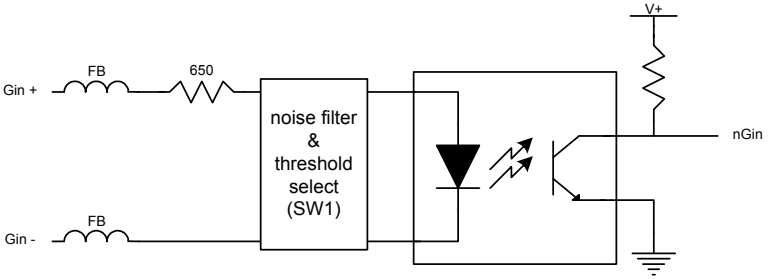
2	4	...	38	40
1	3	...	37	39

J3 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 (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.

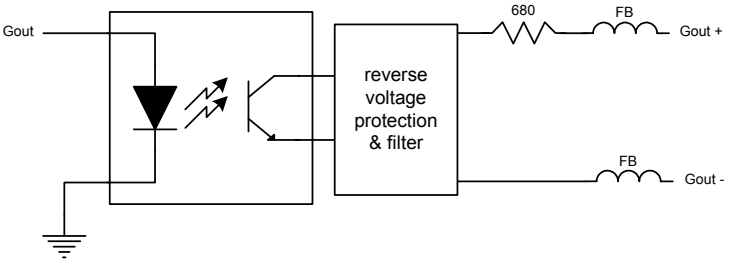


Input Details:

- For single ended TTL signals, the Gin- pin is connected to ground. The threshold point is ~10V by default (for 24V systems) and can be change to ~2V (for TTL or low voltage differential) 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.

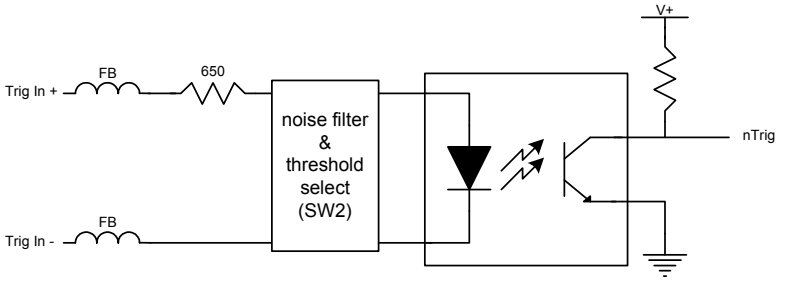


Output Details:

- 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 (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 threshold point is ~10V by default (for 24V systems) and can be changed to ~2V (for TTL or low voltage differential) 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 0µs to 255µs. Any pulse smaller than the programmed value is blocked and therefore not seen by the acquisition circuitry. If no debouncing value is specified (value of 0µs), the minimum value of 1µs will be used.
- Each input has a ferrite bead plus a 650 ohm series resistor on the opto-coupler anode.
- Maximum input signal frequency is 100 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 Line Scan acquisition.
-

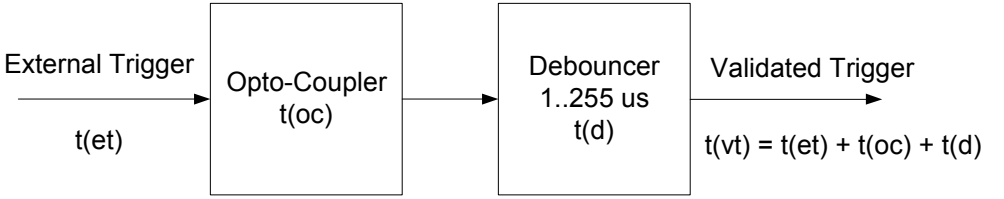


Figure 1: External Trigger Input Validation & Delay

Let	t(et) = time of external trigger in µs t(vt) = time of validated trigger in µs t(oc) = time opto-coupler takes to change state t(d) = debouncing duration from 1 to 255µs
trigger high	For an active high external trigger, t(oc) = 0.5µs: t(vt) = t(et) + 0.5µs + t(d)
trigger low	For an active low external trigger, t(oc) = 4.2µs: t(vt) = t(et) + 4.2µs + t(d)

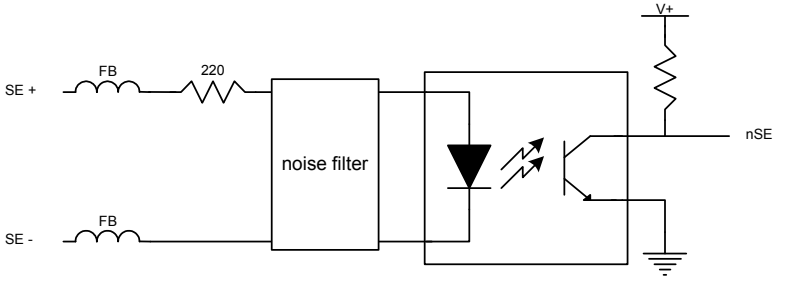
Note: DALSA recommends using an active high external trigger, which minimizes the time it takes the opto-coupler to change state. Specifically, the opto-coupler response time is 0.5µs for active high compared to 4.2µs for active low.

If the duration of the external trigger is > t(oc) + t(d), then a valid acquisition trigger is detected. Therefore, the external pulse with active high polarity must be at least 1.5µs (if debounce time is set to 1) in order to be acknowledged. Any pulse larger than 5.2µs is considered valid.

It is possible to emulate an external trigger using the software trigger, generated by a function call from an application.

Note 4: Shaft Encoder Input Specifications

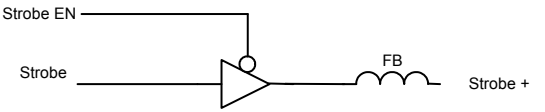
Dual Quadrature Shaft Encoder Inputs (phase A and phase B) are opto-coupled and able to connect to differential signals (RS422) or single ended TTL source signals. The following figure is typical for each input.



- For single ended TTL signals, the SE- pin is connected to ground. The threshold 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.
- Opto-coupler response time is 0.25µs for a rising signal.
- Opto-coupler response time is 2.8µs for a falling signal.
- See "Line Trigger Source Selection for Line Scan Applications" on page 61 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 RS-422)
  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

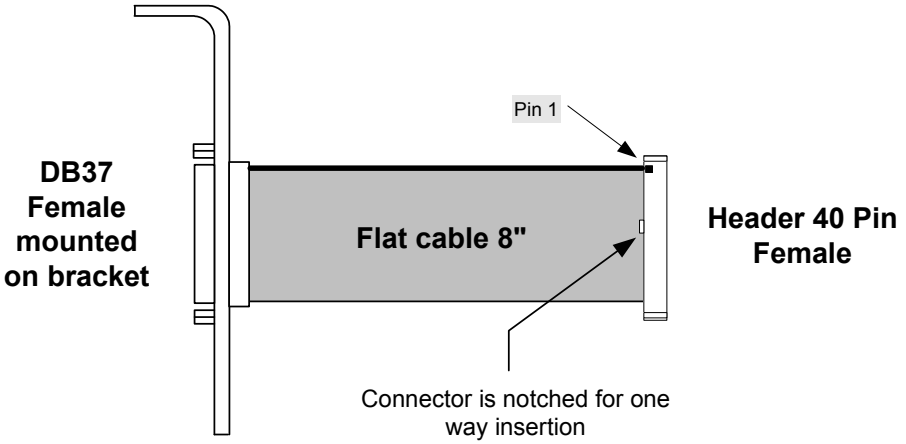
- Connect the PC floppy drive power connector to J7 so as to supply DC power to a camera. Both 5Vdc and 12Vdc are available on J3 or on the DB37 External Signals Bracket Assembly.
- Both the 5Volt and 12Volt power pins have a 1.5 amp re-settable fuse on the board. If the fuse is tripped, turn off the 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 J3 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 J3 header. When connecting to J3, make sure that the cable pin 1 goes to J3 pin 1 (see the layout drawing "X64 Xcelera-LVDS PX4 Board Layout Drawing" on page 84).

Note: For additional independent I/O signals use the optional X-I/O module. See "Appendix: X-I/O Module Option" on page 113.

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 "J3: External Signals Connector " on page 101 for signal descriptions.

DB37 Pin Number	Signal	J3 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

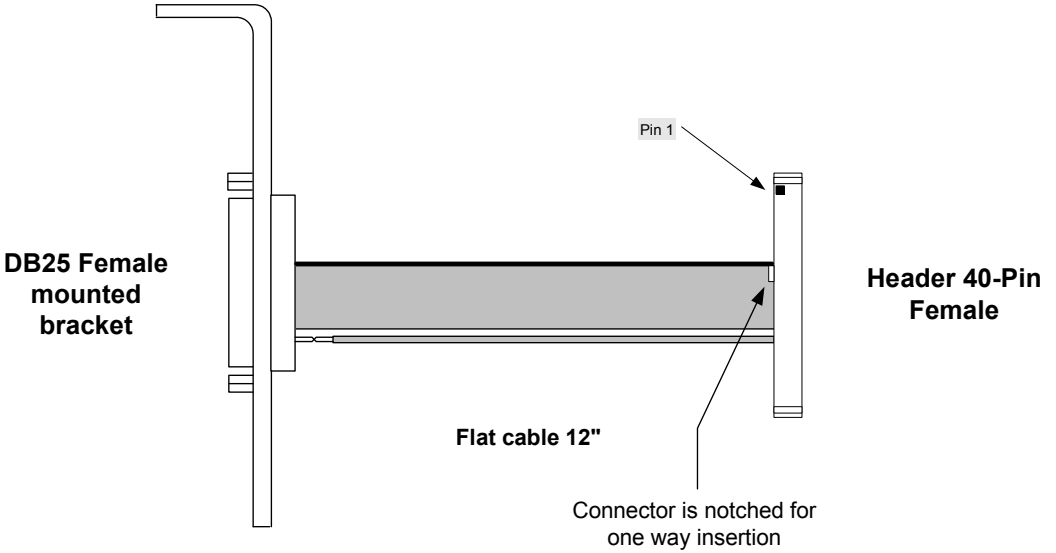
External Signals Connector Bracket Assembly (Type 2)

The External Signals bracket (OR-X4CC-0TIO2) provides a simple way to bring out the signals from the External Signals Connector J3 to a bracket mounted DB25. Connect external cables designed for the DALSA X64-CL iPro directly.

Install the bracket assembly into an adjacent PC expansion slot and connect the free cable end to the board's J3 header. When connecting to J3, make sure that the cable pin 1 goes to J3 pin 1 (see the layout drawing "X64 Xcelera-LVDS PX4 Board Layout Drawing" on page 84).

Note: For additional independent I/O signals use the optional X-I/O module. See "Appendix: X-I/O Module Option" on page 113.

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 "J3: External Signals Connector " on page 101 for signal descriptions.

DB25 Pin Number	Signal	J3 Connector Pin Number
6	External Trigger Input 0 +	19
19	External Trigger Input 0 -	20
7	External Trigger Input 1 +	21
20	External Trigger Input 1 -	22
8	Shaft Encoder Phase A +	23
21	Shaft Encoder Phase A -	24
9	Shaft Encoder Phase B +	25
22	Shaft Encoder Phase B -	26
11	Strobe Output 0	28
24	Ground	29
10	Strobe Output 1	30
14	Ground	31
15	Ground	38
16	Ground	39
25	Ground	40

J14: Board Sync

Board Sync interconnects multiple X64 Xcelera boards to synchronize acquisitions to one trigger or event. The trigger source can be either an external signal or internal software trigger. The board receiving the trigger is the Master board, while the boards receiving the control signal from the Master board are Slaves.

- **Hardware Connection:** Interconnect two, three, or four X64 Xcelera boards via their J14 connector. The 4 pin cable is wired one to one — i.e. no crossed wires. The cable must be as short as possible and the boards must be in the same system.
- **Master Board Software Setup:** Choose one X64 Xcelera as master. The Sapera parameter CORACQ_PRM_EXT_TRIGGER_SOURCE is set to either *Mode 1–Output to Board Sync* or *Mode 2–Control pulse to Board Sync*. See Sapera documentation for more details.
- **Slave Board Software Setup:** The Sapera parameter CORACQ_PRM_EXT_TRIGGER_SOURCE is set to *From Board Sync*.
- **Test Setup:** The control application starts the acquisition on all slave boards. The acquisition process is now waiting for the control signal from the master board. The master board acquisition is triggered and the acquisition start signal is sent to each slave board (with ~0.8µs delay max).

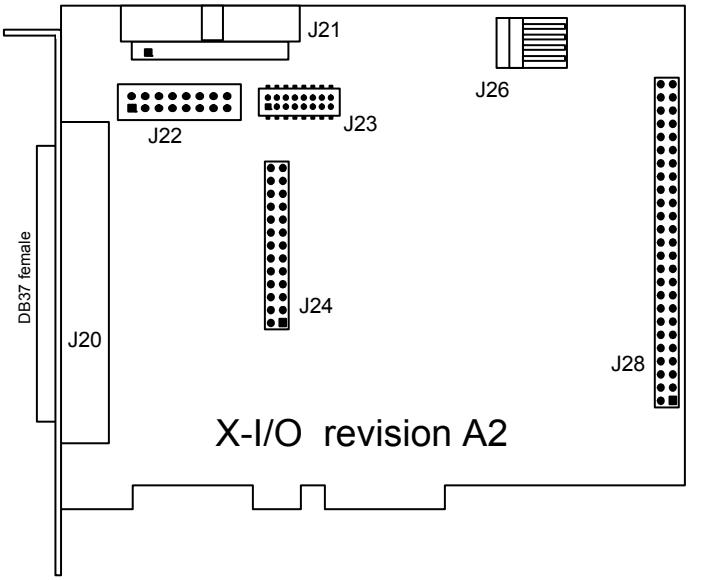
Contact Technical Support for additional information.

Appendix: X-I/O Module Option

X-I/O Module Overview

- The X-I/O module requires X64 Xcelera-LVDS PX4 board driver version 1.00 (or later) and Sapera LT version 5.30 (or later).
- Occupies an adjacent slot to the X64 Xcelera-LVDS PX4 Full. Slot can be either PCI-32 or PCI-64—no PCI signals or power are used.
- Connects to the X64 Xcelera-LVDS PX4 via a 16 pin flat ribbon cable. J23 on X-I/O to J11 on X64 Xcelera-LVDS 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 outputs software selectable as NPN (current sink) or PNP (source driver) type drivers. See "Outputs in NPN Mode: Electrical Details" on page 117 and "Outputs in PNP Mode: Electrical Details" on page 118.
- X-I/O provides 2 opto-coupled inputs. See "Opto-coupled Input: Electrical Details" on page 119.
- X-I/O provides 6 TTL level inputs with software selectable transition point. See "TTL Input Electrical Details" on page 119.
- 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-LVDS 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-LVDS 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-LVDS PX4 board.
- Insert the X-I/O module into any free PCI slot (no PCI electrical connections are used), securing the bracket.
- Connect the supplied X-I/O module 16 pin ribbon cable from J23 to the X64 Xcelera-LVDS PX4 board J11.
- Power on the computer again.
- For new X64 Xcelera-LVDS PX4 and X-I/O module installations, simply follow the procedure to install Sapera and the X64 Xcelera-LVDS PX4 driver (start with "Installing X64 Xcelera-LVDS PX4" on page 11).

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

- If both Sapera and X64 Xcelera-LVDS PX4 driver need to be installed, follow the procedure "Sapera LT Library Installation" on page 11. 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-LVDS 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 14.

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 thumbscrews for a secure fit. Wiring type should meet the needs of the imaging environment.

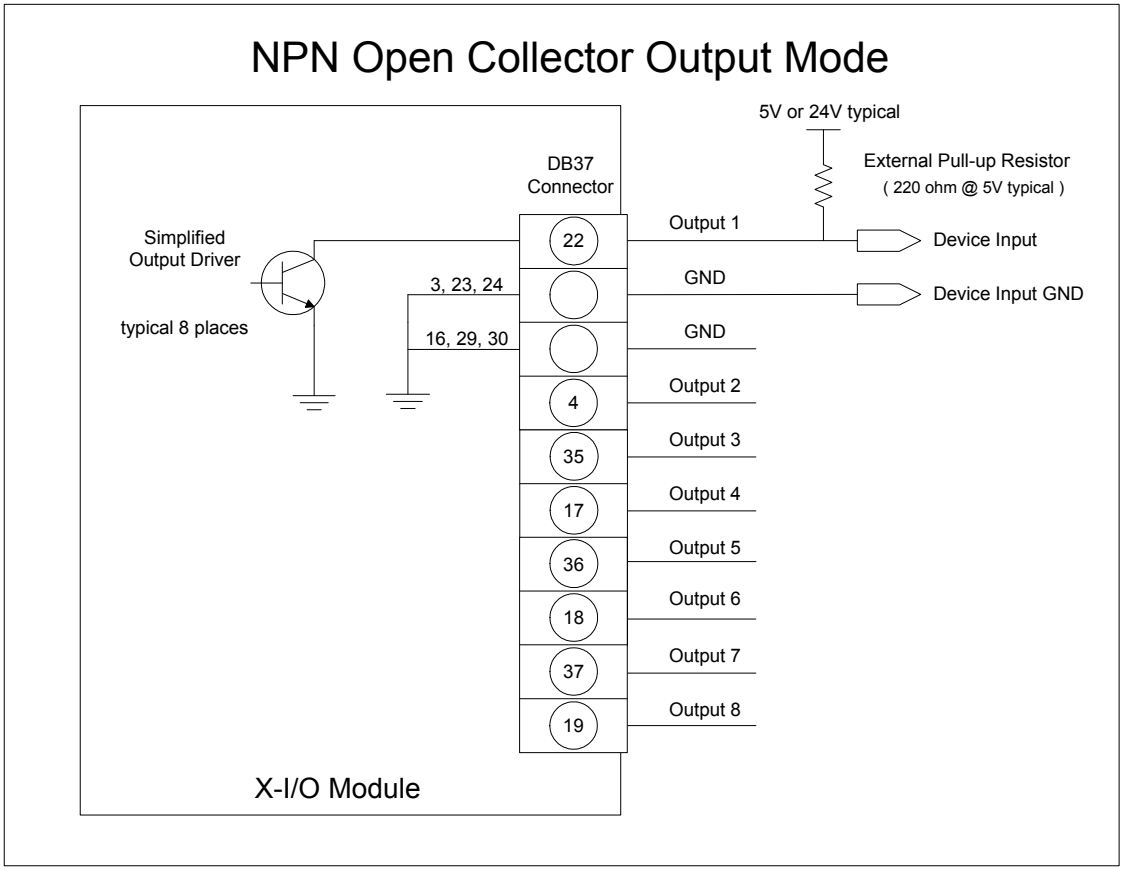
For the external signals Trigger Input, Shaft Encoder Input, and Strobe output, now available on the DB37, refer "J3: External Signals Connector" on page 101 to for signal details.

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	output #1
4	OUT_TTL_2	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	output 3
17	OUT_TTL_4	output 4
36	OUT_TTL_5	output 5
18	OUT_TTL_6	output 6
37	OUT_TTL_7	output 7
19	OUT_TTL_8	output 8

Outputs in NPN Mode: Electrical Details

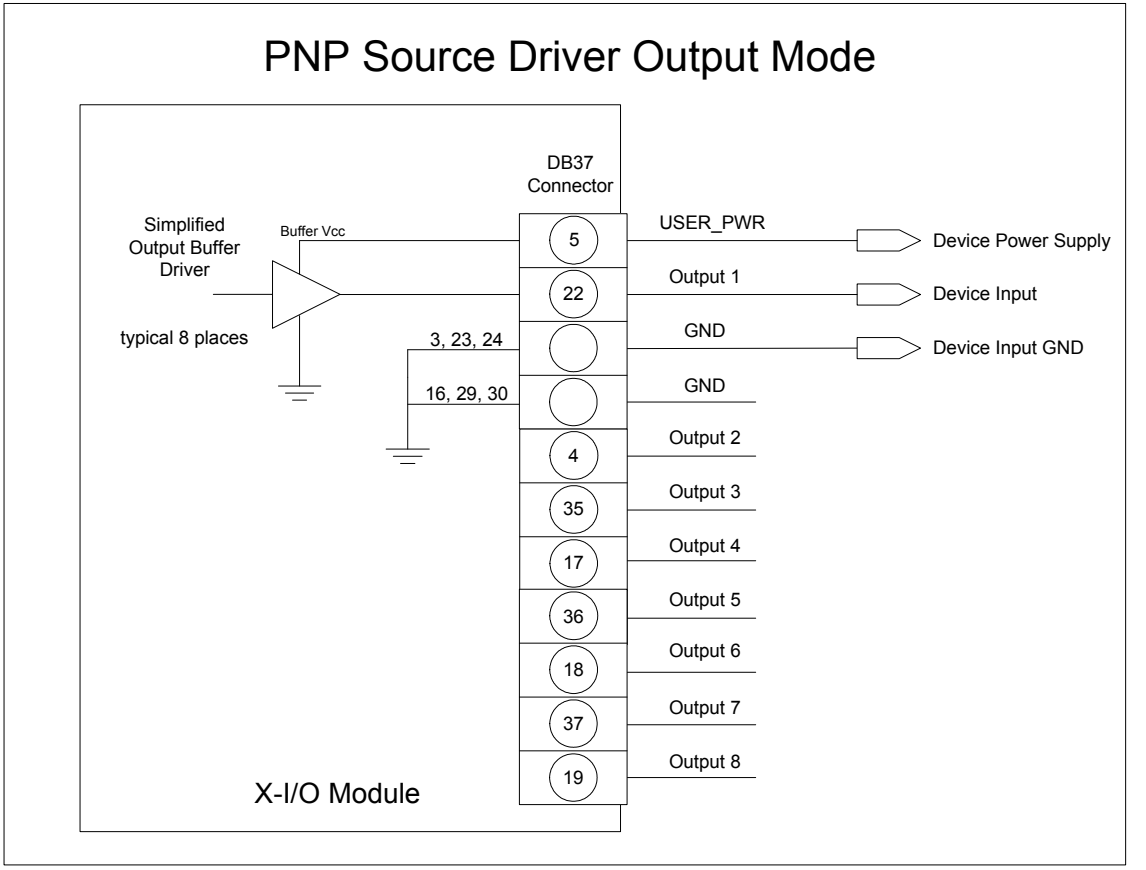
When the outputs are configured for NPN mode (open collector - sink mode) the user is required to provide an external input pull-up resistor on the signal being controlled by the X-I/O output. A simplified schematic and important output specification follow:



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

Outputs in PNP Mode: Electrical Details

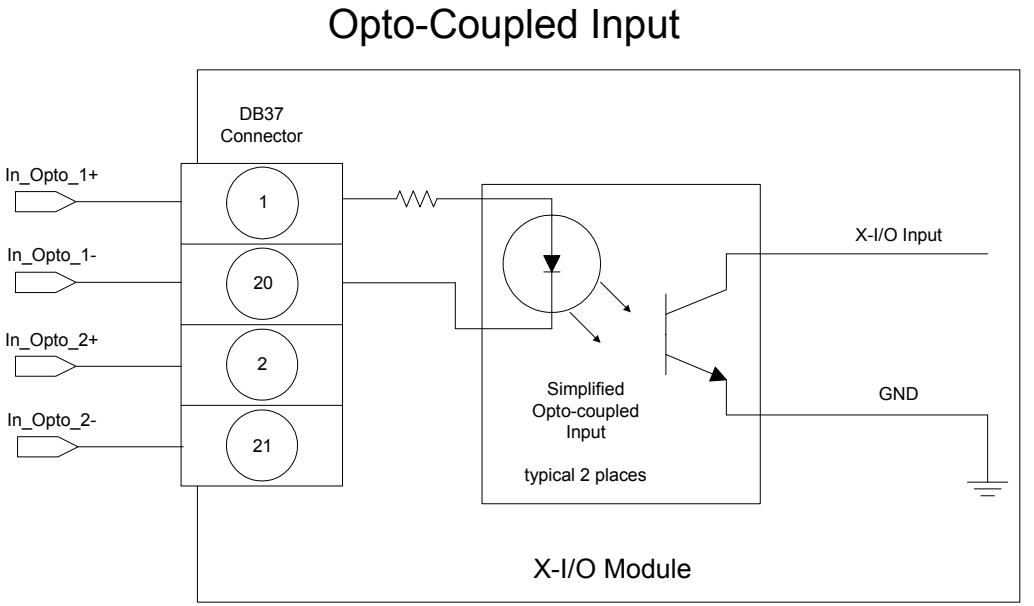
When the outputs are configured for PNP mode (source driver), an external power supply is required to provide the buffer output supply voltage (USER_PWR). A simplified schematic and important output specification follow:



- 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

Use the two opto-coupled inputs with either TTL or RS422 sources. A simplified input schematic and important electrical specification follows.



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 120) 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

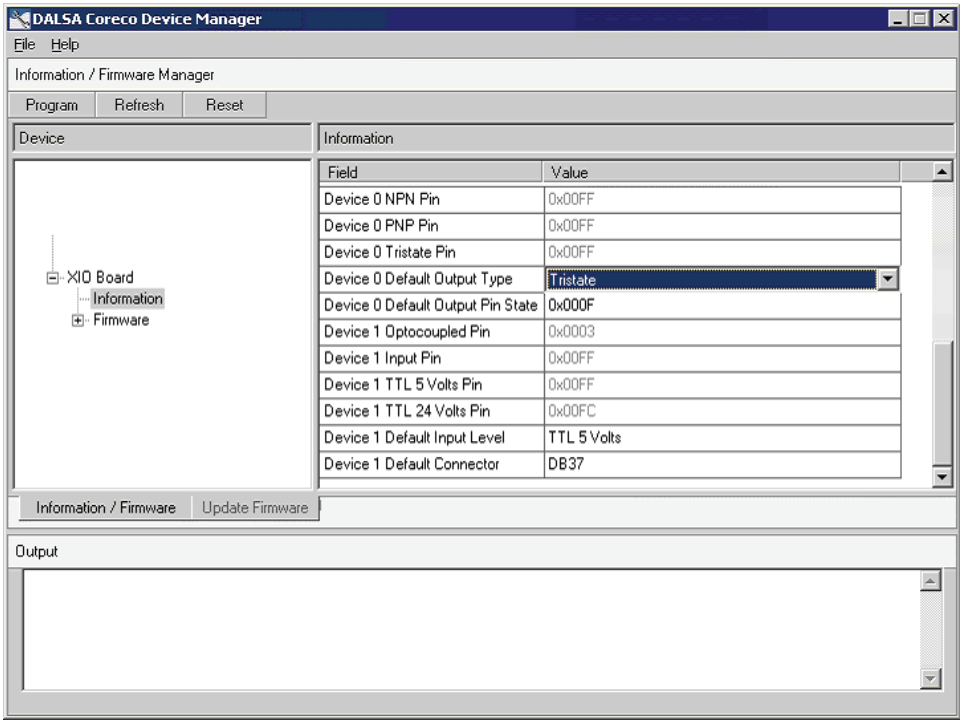
Sapera version 5.30 (or later) provides support for the X-I/O module via an I/O class and demonstration program. Users can use the demonstration program as is, or use the demo program source code to implement X-I/O controls within the custom imaging application.

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. Use the Device Manager program to configuration the initial state. Run the program via the windows start menu: (**Start • Programs • DALSA • X64 Xcelera-LVDS PX4 Device Driver • Device Manager**).

The Device Manager provides information on the installed X64 Xcelera-LVDS 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 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 displays 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 24 Volts, dependent on the signal type 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 Output message 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 controls the I/O capabilities of the X-I/O module on the Sapera board product. The demo will present to the user only the controls pertaining to the selected hardware (in the case of multiple installed boards).

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-LVDS PX4 board is selected and click OK to continue.

General I/O Module Control Panel

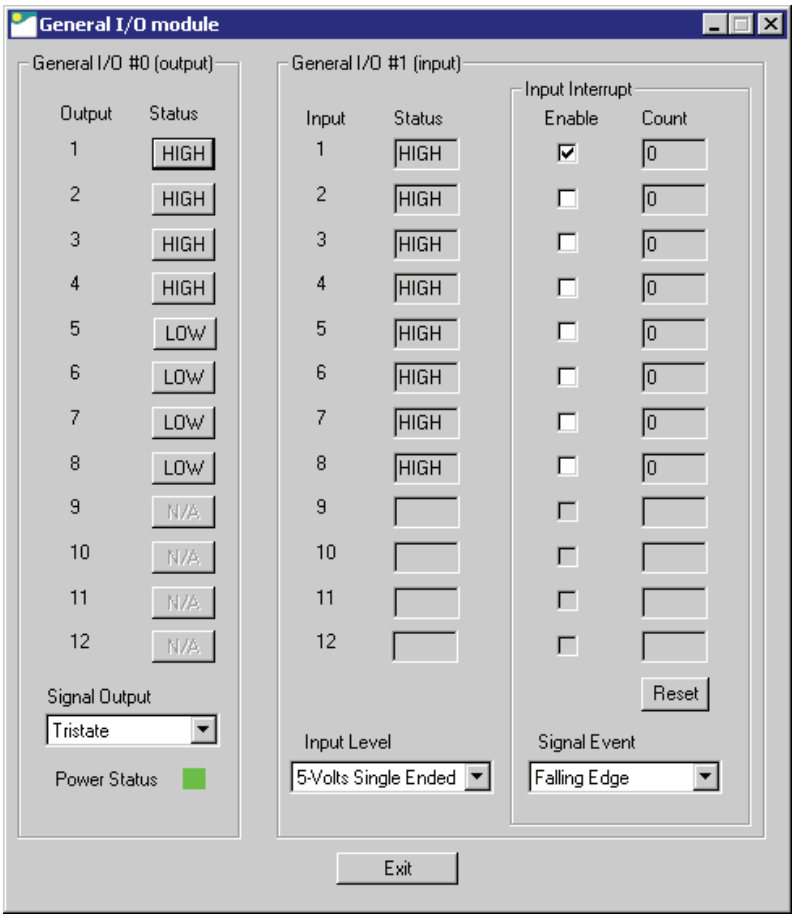
The I/O module control demo presents the I/O capabilities of the installed hardware. The following figure shows the X-I/O module connected to the X64 Xcelera-LVDS PX4 board.

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 is 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()]++;
}
```

Contact Information

Sales Information

Visit our web site: www.teledynedalsa.com/mv

Email: mailto:info@teledynedalsa.com

Canada

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

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

USA Sales

Teledyne DALSA — Billerica office
700 Technology Park Drive
Billerica, Ma.
01821

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

Asia Sales

Teledyne DALSA Asia Pacific
Ikebukuro East 13F
3-4-3 Higashi Ikebukuro,
Toshima-ku, Tokyo
Japan

Tel: +81 3 5960 6353
Fax: +81 3 5960 6354

European Sales

Teledyne DALSA Europe
Breslauer Str. 34
D-82194 Gröbenzell (Munich)
Germany

Tel: +49 - 8142 – 46770
Fax: +49 - 8142 - 467746

Technical Support

Submit any support question or request via our web site:

Technical support form via our web page: Support requests for imaging product installations, Support requests for imaging applications	http://www.teledynedalsa.com/mv/support
Camera support information	
Product literature and driver updates	

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.

FPN (Fixed Pattern Noise):
FPN is the unwanted static variations in response for all pixels in the image.

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.

PCI Express
PCI Express is the serial bus addition to the PCI series of specifications. However PCI Express is not compatible with PCI products or expansion slots. PCI Express products or expansion slots are designated as 1x, 4x, 8x, or 16x, which defines the total data bandwidth and the physical size of the bus connector.

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.

PRNU (Photo Response Non Uniformity):
PRNU is the variation in response between sensor pixels.

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

L

M

N

O

P

Q

R

S

T

U

V

W

X