Xtium3-CLHS-PX8

User Manual
Edition 1.00

sensors | cameras | **frame grabbers** | processors | software | vision solutions

P/N: OC-B8SM-PX8U0
REV: 0001
www.teledynevisionsolutions.com

TELEDYNE

Notice

Document Date: Edition 1.00, Revision 1 released November 18, 2025

About Teledyne DALSA, a business unit of Teledyne Digital Imaging Inc.

Teledyne DALSA is a leader in the design, manufacture, and deployment of digital imaging components and solutions for machine vision. From image sensors, cameras, smart cameras, and frame grabbers to sophisticated vision software and intelligent vision systems, our products define the top end of performance in the world's most demanding digital imaging applications.

Teledyne DALSA is also part of Teledyne Vision Solutions which offers the world's most comprehensive, vertically integrated portfolio of industrial and scientific imaging technology. Aligned under one umbrella, Teledyne DALSA, e2v CMOS image sensors, FLIR IIS, Lumenera, Photometrics, Princeton Instruments, Judson Technologies, Acton Optics, and Adimec form an unrivalled collective of expertise across the spectrum with decades of experience and best-in-class solutions.

Contents

OVERVIEW 7
Series Key Features 7
PRODUCT PART NUMBERS 8
XTIUM3-CLHS PX8 BOARD SPECIFICATIONS 9
HOST SYSTEM REQUIREMENTS 12
DEVELOPMENT SOFTWARE OVERVIEW 12
Sapera LT Library 12
Sapera Processing Library 12
INSTALLATION PROCEDURE 12

XTIUM3-CLHS PX8 ADVANTAGES 13
TRIGGER-TO-IMAGE RELIABILITY (T2IR) 13
DATA FORWARDING FOR PARALLEL PROCESSING 14
MULTI-CAMERA SYNCHRONIZATION USING BOARD SYNC 14
TDI MULTI-PLANE IMAGE PROCESSING 15
TDI VERTICAL SPATIAL CORRECTION 16
FLAT FIELD CORRECTION 16
LOOKUP TABLE (LUT) 16
HORIZONTAL & VERTICAL FLIP OPERATIONS 17

TECHNICAL SPECIFICATIONS 18
XTIUM3-CLHS FLOW DIAGRAM 18
CONNECTORS AND BOARD LAYOUT 19
Xtium3-CLHS PX8 Board Layout Drawing 19
Connector / LED Description List 19
Xtium3-CLHS PX8 End Bracket Detail 20
Status LEDs Functional Descriptions 20
J1: External I/O Signals Connector 22
J2: Camera Link HS Connector 27
J3: Data Forwarding Connector 27
J4: Internal I/O Signals Connector 28
J5: Multi-Board Sync / Board-to-Board Bi-directional I/Os 33
INPUT ELECTRICAL SPECIFICATIONS 34
General Purpose Inputs 34
Rotary (Shaft) Encoder/ External Line Trigger Input Specifications 36
OUTPUT ELECTRICAL SPECIFICATIONS 39
LVTTL Output Specifications 39
Open Collector Output Specifications 41
EMC DECLARATIONS OF CONFORMITY 42
FCC Statement of Conformance 42
EU and UKCA Declaration of Conformity 42

ACQUIRING IMAGES 43
BASIC SETUP 43
Basic Acquisition Parameters 43
TDI Multi-Plane Acquisition 44
USING TRIGGERS TO ACQUIRE IMAGES 45
Area Scan Applications 46
Line Scan Applications 47
Variable and Fixed Length Frames 50
Encoder Interface Timing 54
Encoder Averaging Engine 55

Shaft Encoder Direction and Count *59*

XTIUM3-CLHS PX8 SAPERA PARAMETERS **63**
BASIC TIMING CATEGORY 64
Parameter Descriptions *64*
ADVANCED CONTROL CATEGORY 66
Parameter Descriptions *67*
Sapera Acquisition Methods *68*
EXTERNAL TRIGGER CATEGORY 69
Parameter Descriptions *70*
IMAGE BUFFER AND ROI CATEGORY 73
Parameter Descriptions *73*
Horizontal and Vertical Flip *74*

USING THE XTIUM3-CLHS PX8 WITH SAPERA **75**
SAPERA SERVERS AND RESOURCES 75
CAMEXPERT QUICK START 76
CamExpert Interface *77*
Sapera Camera Configuration Files *79*
Lookup Table (LUT) Configuration *80*
Multi-Planes Image Processing *82*
2-Plane Processing Operations *83*
3-Plane Processing Operations *85*
4-Plane Processing Operations *88*
Using the Flat Field Correction Tool *91*
SAPERA DEMO APPLICATIONS 96
Grab Demo Overview *97*
General I/O Demo Overview *98*
TRIGGER TO IMAGE RELIABILITY 99
Supported Events and Transfer Methods *100*

XTIUM3-CLHS PX8 UTILITIES **104**
DEVICE MANAGER 105
Device Manager – Board Viewer *105*
Information Field Description *106*
Firmware Information *108*
Parameter Information *109*
Changing Device Settings *110*
Device Information Report *111*
FIRMWARE LOADER 112
Firmware Update: Automatic Mode *112*
Firmware Update: Manual Mode *112*
Executing the Firmware Loader from the Start Menu *113*
TELEDYNE LOG VIEWER 114
Log Viewer Options *115*
Log Server Settings *115*
XTIUM3 FRAME GRABBER DIAGNOSTIC TOOL 116
Diagnostic Tool Main Window *116*
Diagnostic Tool *117*
PCI DIAGNOSTIC TOOL 119
SAPERA MONITOR 120
Using Sapera Monitor *120*
Sapera Monitor Window *121*
Sapera Monitor Menu Commands *122*
SAPERA CONFIGURATION 124
Viewing Installed Sapera Servers *124*
Increasing Contiguous Memory for Sapera Resources *125*

Multi-Threaded Transfer Callback Optimization *126*

INSTALLING XTIUM3-CLHS PX8 ..... **127**
Supported Windows Version *127*
Software Download *127*
Secure Boot *128*
INSTALLATION OVERVIEW 128
Additional Installation Types: *128*
QUICK START SETUP & INSTALLATION 129
HARDWARE INSTALLATION 134
Warning! (Grounding Instructions) *134*
Hardware Installation Procedure: *134*
Xtium3-CLHS PX8 Driver Installation on Windows *135*
Multi-board Data Forwarding Hardware Setup *136*
Multi-board Sync & I/O Setup *139*
CABLES & ACCESSORIES 141
Camera Link HS Cables Overview and Resources *141*
DH40-27S Cable to Blunt End (OR-YXCC-27BE2M1, Rev B1) *142*
DH40-27S Connector Kit for Custom Wiring *143*
Board Sync Cable Assembly *144*

APPENDIX A: ADDITIONAL INSTALLATION TYPES **145**
UPGRADING SAPERA OR BOARD DRIVER 145
Board Driver Upgrade Only *145*
Upgrading both Sapera and Board Driver *145*
Preserving Board Parameters during Driver Upgrade *146*
PRESERVING BOARD PARAMETERS DURING BOARD REPLACEMENT OR SYSTEM CLONING 147
SILENT INSTALLATION 148
Creating a Response File *148*
Running a Silent Mode Installation *149*
SILENT MODE UNINSTALL 149
Creating a Response File *149*
Running a Silent Mode Uninstall *149*
SILENT MODE INSTALLATION RETURN CODE 150
INSTALLATION SETUP WITH CORAPPLAUNCHER.EXE 150
CUSTOM DRIVER INSTALLATION USING INSTALL.INI 151
Creating the install.ini File *151*
Run the Installation using install.ini *151*

APPENDIX B: TROUBLESHOOTING PROBLEMS **152**
OVERVIEW 152
PROBLEM TYPE SUMMARY 152
First Step: Check the Status LED *152*
Possible Installation Problems *152*
Possible Functional Problems *153*
TROUBLESHOOTING PROCEDURES 154
Checking for PCI Bus Conflicts *154*
Windows Device Manager *155*
BSOD (blue screen) Following a Board Reset *155*
Sapera and Hardware Windows Drivers *155*
Recovering from a Firmware Update Error *156*
Driver Information via the Device Manager Program *157*
Teledyne DALSA Log Viewer *158*
On-board Image Memory Requirements for Acquisitions *158*
Symptoms: CamExpert Detects no Boards *158*
Symptoms: Xtium3-CLHS PX8 Does Not Grab *159*
Symptoms: Card grabs black *159*

Symptoms: Card acquisition bandwidth is less than expected 160

APPENDIX C: SAPERA PARAMETER REFERENCE 161
XTIUM3-CLHS PX8 SUPPORTED SAPERA PARAMETERS 161
Camera Related Capabilities 161
Camera Related Parameters 161
VIC Related Parameters 162
ACQ Related Parameters 167
Transfer Related Capabilities 168
Transfer Related Parameters 168
General Outputs #1: Capabilities (for GIO Module #0) 169
General Outputs #1: Parameters (for GIO Module #0) 169
General Inputs #1: Capabilities (for GIO Module #1) 169
General Inputs #1: Parameters (for GIO Module #1) 169
Bidirectional Board-to-Board I/Os: Capabilities (for GIO Module #2) 170
Bidirectional Board-to-Board I/Os: Parameters (for GIO Module #2) 170
Open Collector General Outputs: Capabilities (for GIO Module #3) 170
Open Collector General Outputs: Parameters (for GIO Module #3) 170
Multi-Plane Input Processing Parameters 171

APPENDIX D: XTIUM FRAME GRABBER USER COMMANDS 175
OVERVIEW 175
Demo Program 175
Generic Function 177
CorManControl 177
USER COMMANDS 177
CORCMD_USER_DEVICE_PARAMETER_READ 177
CORCMD_USER_DEVICE_PARAMETER_READ_EX 177
CORCMD_USER_DEVICE_PARAMETER_WRITE 178
USER FUNCTIONS 179
CORHW_USER_DEVICE_PRM_GET_BANDWIDTH 180
CORHW_USER_DEVICE_PRM_GET_CAMERA_PORT_READ 180
CORHW_USER_DEVICE_PRM_GET_LANES_STATS 181
CORHW_USER_DEVICE_PRM_GET_NB_LANES 182
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_BIT_TRANSFER_RATE 182
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NB_LANES 182
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NUMBER 183
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_PAYLOAD_SIZE 183
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_REQUEST_SIZE 183
CORHW_USER_DEVICE_PRM_GET_PCI_FUNCTION_NUMBER 184
CORHW_USER_DEVICE_PRM_GET_PCI_SLOT_NUMBER 184
CORHW_USER_DEVICE_PRM_GET_TEMPERATURE 184
CORHW_USER_DEVICE_PRM_GET_THEORETICAL_BANDWIDTH 185
CORHW_USER_DEVICE_PRM_GET_TRANSMISSION_ENCODING 185
CORHW_USER_DEVICE_PRM_GET_USER_DEVICE_INFO_VALUE 186
CORHW_USER_DEVICE_PRM_RESET_LANES_STATS 186
CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_CONNECT 187
CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_DISCONNECT 187
CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_WRITE 188
CORHW_USER_DEVICE_PRM_SET_DIAGNOSTIC_MODE 189
CORHW_USER_DEVICE_PRM_SET_USER_DEVICE_INFO_VALUE 189

APPENDIX E: ADVANCED PROCESSING LICENCES 190
INSTALLING AN ADVANCED PROCESSING LICENSE 191

CONTACT INFORMATION ____ **193**
SALES INFORMATION 193
TECHNICAL SUPPORT 193

Figures

Figure 1: Multi-camera Synchroization using Board Sync *14*
Figure 2: Horizontal and Vertical Flip Operations *17*
Figure 3: Xtium3-CLHS Flow Diagram *18*
Figure 4: Board Layout *19*
Figure 5: End Bracket Details *20*
Figure 6: End Bracket Staus LEDs *20*
Figure 7:External Signals to J1 Connection Diagram *24*
Figure 8:Output Signals to J1 Connection Diagram *25*
Figure 9:External RS-422 Signals to J1 Connection Diagram *26*
Figure 10: External Signals to J4 Connection Diagram *30*
Figure 11:Output Signals to J4 Connection Diagram *31*
Figure 12: Output Signals to J4 Connection Diagram *32*
Figure 19: General Inputs Electrical Diagram *34*
Figure 15: Encoder Input Electrical Diagram *36*
Figure 16: Interfacing to a Line Driver Output *37*
Figure 17: Interfacing to an Open Collector Output *37*
Figure 18: Interfacing TTL to TTL Shaft Encoder Inputs *38*
Figure 19: General Outputs LVTTL Electrical Diagram *39*
Figure 20: General Outputs Open Collector Electrical Diagram *41*
Figure 21: Spatial Correction *44*
Figure 22: Line scan, Fixed Frame, No Trigger *51*
Figure 23: Line scan, Fixed Frame, Edge Trigger *51*
Figure 24: Line scan, Fixed Frame, Level Trigger (Roll-Over to Next Frame) *52*
Figure 25: Line scan, Variable Frame, Edge Trigger (Active High determines Frame Length) *52*
Figure 26: Line scan, Variable Frame, Level Trigger (Roll-Over) *53*
Figure 27: Encoder Input with Pulse-drop Counter *54*
Figure 28: Using Shaft Encoder Direction Parameter *59*
Figure 29: CamExpert Program *77*
Figure 30: CamExpert Histogram of Dark Image *92*
Figure 31: CamExpert Histogram of Bright Image *93*
Figure 32: CamExpert Flat Field Correction Menu Command *94*
Figure 33: CamExpert Flat Field Correction Dialog *94*
Figure 34: The Sapera Explorer application *96*
Figure 35: Board Information via Device Manager *105*
Figure 36: Firmware Information via Device Manager *108*
Figure 37: Manually Loading Firmware via Device Manager *108*
Figure 38: Device Manager Parameter Panel *109*
Figure 39: Automatic Firmware Update *112*
Figure 40: Manual Firmware Update *113*
Figure 41: Start Menu Firmware Update Shortcut *113*
Figure 42: Diagnostic Tool User Interface *116*
Figure 43: Diagnostic Tool “Resource in use” *117*
Figure 44: Diagnostic Tool Self-Test Window *117*
Figure 45: Data Forwarding Block Diagram *136*
Figure 46: Dual Camera Output Connection Block Diagram *138*
Figure 47: DH60-27P Cable No. OR-YXCC-27BE2M1 Detail *142*
Figure 48: Photo of cable OR-YXCC-27BE2M1 *142*
Figure 49: Photo of cable OR-YXCC-BSYNC40 *144*
Figure 50: Device Manager Parameter Setting Differences *146*
Figure 51: Firmware Update Status *147*

Figure 52: Same Firmware For All Devices Checkbox *147*
Figure 53: Create an install.ini File *151*
Figure 54: PCI Diagnostic Program *154*
Figure 55: PCI Diagnostic Program – PCI bus info *155*
Figure 56: Using Windows Device Manager *155*
Figure 57: Board Firmware Version *157*
Figure 58: PCI Diagnostic Tool BM Button *159*

Overview

Series Key Features

Compliant with Camera Link HS (CLHS) specification version 1.1 (X-Protocol)

(visit http://www.automate.org for details on industry standards)

- Uses a PCIe x8 Gen4 slot to maximize transfers to host computer buffers
- Supports up to 7 lanes.
- Bit transfer rate of 10.3125 Gbps per lane: up to 72.187 Gbps with 7 lanes.
- Acquire from Monochrome cameras, both area scan and line scan
- Output lookup tables
- Vertical and Horizontal Flip supported on board
- External Input Triggers and Rotary Encoder inputs, along with Strobe outputs
- Supports Multi-board Sync for trigger events, to simultaneously acquire from multiple cameras.
- Supports Data Forwarding Mode, where camera image data is automatically transferred to one or more Xtium3-CLHS boards (each installed in a separate computer), allowing distributed processing of the acquisition. The specification defines a device discovery methodology that can be automated and which provides plug and play capability
- Camera Link HS cameras implement GenICam and associated GenCP, thus resulting in ease of use for Teledyne DALSA or third party cameras
- Supports a number of acquisition events in compliance with “Teledyne DALSA’s Trigger to Image Reliability”

Product Part Numbers

Xtium3-CLHS PX8 Board

Item	Product Number
Xtium3-CLHS PX8	OR-B8S0-PX870

Xtium3-CLHS PX8 Software

Item	Availability
Sapera LT image acquisition SDK (API) version 9.10 or later for full feature support (required)	Free download from Teledyne Visions Solutions website
(optional) Sapera Processing Imaging Development Library includes over 600 optimized image-processing routines.	Free download from Teledyne Visions Solutions website

Optional Xtium3-CLHS PX8 Cables & Accessories

Item	Product Number
DH40-27S cable assembly to blunt end: 6 ft. cable I/O 27 pin Hirose connector to blunt end. This cable assembly connects to J1.	OR-YXCC-27BE2M1
External Signals bracket provides a simple way to bring out the signals from the External Signals Connector J4 to a bracket mounted DB37.	OC-X4CC-IOCAB Cable assemblies for I/O connector J4
DH40-27S Connector Kit for Custom Wiring: Comprised of a DH40-27S connector plus screw lock housing kit	OR-YXCC-H270000
Cable assembly to connect to J5 (Board Sync) Connecting 2 boards Connectiing 3 or 4 boards Connectiing 5 or 6 boards	 OR-YXCC-BSYNC20 OR-YXCC-BSYNC40 OR-YXCC-BSYNC60
Data Cable (10M): Active Optical Cable (AOC) cable 10 meters, screw lock CX4 connector	AC-CA-00007-00-R

Xtium3-CLHS PX8 Board Specifications

Frame Grabber	Xtium3-CLHS PX8
Board Input Interface	Camera Link High Speed (CLHS)
Part Numbers	OR-B8S0-PX870
Input Camera Ports	1 camera
Maximum Number of Lanes	7
Transmission Rate	Up to 7-lanes x 10.3125 Gbps (72.187 Gbps total)
Input Camera Bandwidth	Up to 9.32 GB/s (8.68 GB /sec binary) in Frame Grabber memory
On-Board Memory	4 GB on-board memory image buffers 4 GB on-board memory for processing functions
Advanced Hardware Functionality	Automatic Vertical Spatial Correction for TDI multi-plane cameras Supports camera Data Forwarding up to 5 additional boards for distributed image processing Supports multi-camera synchronization of 2 to 4 boards to grab images from multiple independent cameras in one image buffer Data conversion Real-time, user selectable image processing Supports T2IR (Trigger2Image Reliability) framework with comprehensive event notifications Rotary (shaft) encoder averaging engine, drop/multiplier, count and direction controls
PCI Bus related	
Host Bus	PCIe Gen4 x 8 slot
PCIe Payload	Up to 1024 bytes
Bandwidth to Host System (in a PCIe Gen4 x8 slot)	PCIe bus Output: up to 13.2 GB/sec (12.3GB/sec binary) sustained (PCIe Payload @ 512 bytes) PCIe bus Output: up to 12.8 GB/sec (12 GB/sec binary) sustained (PCIe Payload @ 256 bytes) PCIe Payload and maximum bandwidth obtained is dependent on PC characteristics and image size.

NOTE

When calculating bandwidth requirements for memory and throughput ensure that units such as GB/s use the same base, decimal or binary. For example, a binary GB is 1073741824 bytes. This should be considered when translating from Gbps to GB where a Gbps is 1000000000 bits; GB refers to the amount of data stored, while Gb refers to the speed of data transfer.

Features	
Image Processing	TDI Multi-Plane Image Processing functions such as summation and averaging
	Flat Field/Flat Line Correction / Pixel Correction
	Lookup Table (LUT)
Image Flip	Horizontal and Vertical
	Image Timestamp
Line Scan Acquisition Control	Fixed or variable frame lengths (using external pulse(s))
Trigger Generator	Line Trigger: maximum frequency (to camera) = 1 MHz, in steps of 4 ns. Frame Trigger: maximum frequency (to camera) = 41 KHz, in steps of 1 us. Strobe Signal: up to 4 strobe pulses (for either linescan or areascan)
Synchronization Source	External Trigger (line or frame), Software, Board Sync
Mechanical Interface	
Data Input Connectors	Data input: 1 x CX4 thumbscrew, AOC ready Data forward: 1 x CX4 thumbscrew, AOC ready
I/O connectors	I/O available on DH60-27P connector (J1) and on 40-pin TST-120-01-G-D (J4)
Electrical Interface	
Inputs Voltage	RS-422 (LVDS) / TTL /12V / 24V
General Purpose Inputs/Outputs	Differential Input = 2 Rotary encoder (quadrature phase A + B) or line trigger input • Differential maximum input frequency up to 8 MHz Single ended = 6 inputs General Purpose Inputs (including 2 external frame triggers) • maximum input frequency LVDS/TTL of up to 8 MHz • maximum input frequency 12/24V of up to 2 MHz • maximum interrupts (up to 10 000 software events total)
	12 General Purpose Outputs • 4 LVTTL General Purpose Outputs (including 2 Strobe Outputs) • 8 Open Collector General Purpose Outputs (including 2 Strobe Output)s
Board-to-Board IOs	8 Bi-Directional IOs (including 2 Board Sync) Supports multi-camera synchronization up to 6 boards
Power Requirement (during acquisition)	24 mW on +3.3V 22.65 W on +12V
Environmental condition	
Ambient Temperature:	10° to 50°C (operation) -40° to 75°C (storage)
Relative Humidity:	5% to 90% non-condensing (operating) 0% to 95% (storage)
MTBF @40°C	21 years

Supported Camera Specifications	
CLHS Standard	Version 1.1 and 2.x compliant
Camera Scan Type	Area scan & Line scan
Camera Format	CLHS X-protocol (64/66 Bit Encoding)
Bit Transfer Rate	Up to 7-lanes @ 10.3125 Gb/s (72.187 Gbps total)
Supported Input Pixel Format	8, 10, 12-bit/pixel
Horizontal Resolution	Minimum: 32 Pixels per lane Maximum 8-bits/pixel x 64k Pixels/Line/Plane 16-bits/pixel x 32k Pixels/Line/Plane
Vertical Resolution	Minimum: 1 line Maximum: 65536 lines for area scan cameras Infinite line count for line scan cameras (limited by host memory)
Software Requirements	
Operating System Support	x86 Platform : • Windows 11 (64-bits) • Ubuntu: 20.04.1 LTS (kernel 5.4), up to 22.04.5 LTS (kernel 6.8), up to 24.04.3 LTS (kernel 6.14) • Redhat: RHEL 8.10 (kernel 4.18), RHEL 9.6 (kernel 5.14)), RHEL 10.0 (kernel 6.12) • SUSE : SLES 15 SP3 (kernel 5.3) • Debian 13 (kernel 6.12) Arm64 Platform : • Ubuntu for ARM64 (22.04) – Nvidia Jetson ORIN Contact Teledyne sales for Linux driver availability
Software compatibility	Teledyne Sapera LT SDK v9.10 or greater Contact Teledyne sales for GenTL compatibility (.CTI file)

Host System Requirements

Xtium3-CLHS PX8 Dimensions

Approximately 5 in. (13 cm) wide by 4 in. (10 cm) high

General System Requirements for the Xtium3-CLHS PX8

- PCI Express Gen4 x8 slot compatible;
- Will work in Gen1, Gen2 or Gen3 x8 slot with reduced bandwidth to host.
- On some computers the Xtium3-CLHS PX8 may function installed in a x16 slot. The computer documentation or direct testing by the user is required.
- Xtium3-CLHS PX8 operates correctly when installed in a multi-processor system (including Hyper-Threading multi-core processors).

Operating System Support

Windows 11 64-bit

NOTE

Ensure adequate airflow for proper functioning of the board across the entire temperature range of 10 – 50°C. Airflow measuring 220 LFM (linear feet per minute) across the surface of the board is recommended.

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 Teledyne 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. Applications are developed using either C++ or .NET frameworks.

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

Sapera LT is available as a free download from the Teledyne Visions Solutions website.

Sapera Processing Library

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

Installation Procedure

See the Installing Xtium3-CLHS PX8 section.

Xtium3-CLHS PX8 Advantages

The Xtium3-CLHS PX8 offers unique multi-board acquisition capabilities, hardware-based processing and other features that provide industry-leading advantages for high-bandwidth applications such as:

- Trigger-to-Image Reliability (T2IR)
- Data Forwarding for Parallel Processing
- Multi-camera Synchronization using Board Sync
- TDI Vertical Spatial Correction
- TDI Multi-Plane Image Processing
- Lookup Table (LUT)
- Horizontal & Vertical Flip Operations

Trigger-to-Image Reliability (T2IR)

For more information, see the Trigger to Image Reliability section. Also, visit the Teledyne website, to access the T2IR primer and FAQ:

https://www.teledynevisionsolutions.com/learn/learning-center/machine-vision/trigger-to-image-reliability-t2ir/

Overview

All Teledyne image acquisition devices implement what is referred to as Trigger-to-Image Reliability (T2IR) using a variety of techniques, from user-defined trigger parameters to buffer management, image metadata and numerous types of event flags, to ensure that images are acquired, transferred and processed with robustness and traceability.

Teledyne's T2IR framework includes powerful GUI based tools for continuous monitoring and rapid pinpoint of errors that are hard to trace back. This continuous system monitoring and deep debugging tools help reduce downtime. Refer to the Xtium3-CLHS PX8 Utilities section for more information on these tools.

Data Forwarding for Parallel Processing

For complete information, see the Data Forwarding Setup, L3: Data Forwarding Status LED, and J3: Data Forwarding Connector sections.

Overview

The CLHS data forwarding feature allows for the automatic transfer of camera image data to one or more Xtium3-CLHS boards, enabling distributed processing of image acquisition.

When acquiring images at the maximum bandwidth, the amount of image data that can be processed on one computer, even with multithreading, may not be adequate. In these cases, the frame grabber connected to the camera (the “master’) is connected in series to several Xtium3-CLHS PX8 boards in separate computers and the image data forwarded to all frame grabbers simultaneously.

Each frame grabber can then perform processing on the image such that the necessary operations can be performed within the application’s time requirements.

Multi-camera Synchronization using Board Sync

For complete information, see J5: Multi-Board Sync / Bi-directional General I/Os and Multi-board Sync & I/O Setup.

Overview

This setup supports multi-board sync for trigger events, allowing simultaneous acquisition from multiple cameras. A single trigger to one board is sent to the boards attached through the board sync connector allowing them to use this signal to trigger an acquisition from cameras connected to them through the frame grabber IO connector.

Boards synchronized in this manner must reside in the same computer; ensure that the PCIe slots have adequate lanes for the expected bandwidth (if necessary synced boards can also use data forwarding to boards in separate PCs as required).

As synchronized boards reside in the same computer, each board has access to the same acquisition memory space, allowing for all of them to acquire into child buffers of a single parent buffer. That is, synchronized board cameras can all grab into the same single buffer to create one image.

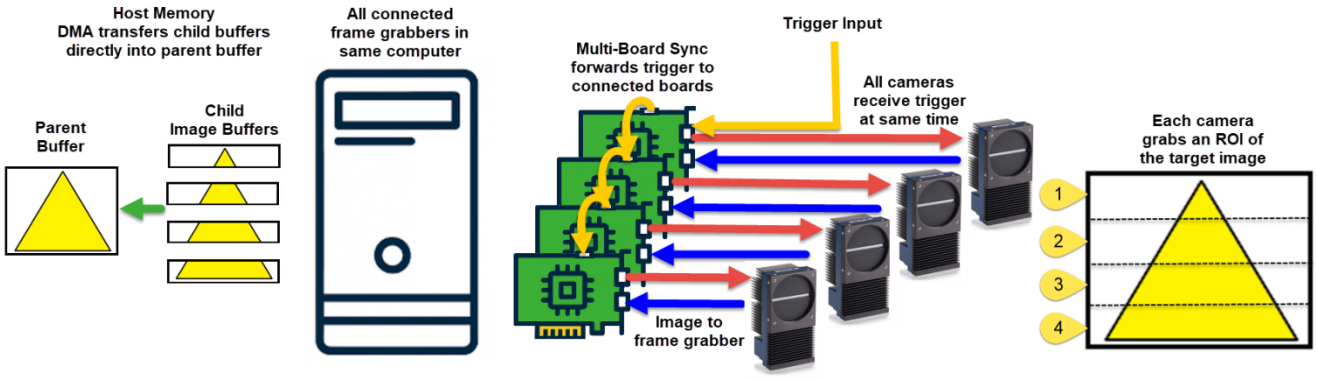


Figure 1: Multi-camera Synchroization using Board Sync

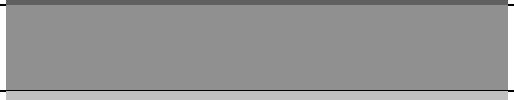
TDI Multi-Plane Image Processing

Complete information is available in the Multi-Planes Image Processing section.

Overview

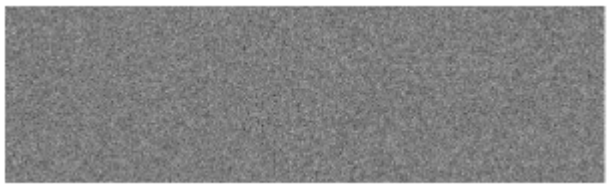
The Xtium3-CLHS PX8 supports the following custom on-board functions for real-time processing of multi-plane images:

- **Plane summation with saturation:** This effectively increases the exposure by combining the light gathered by multiple lines into one line to create a brighter image and is useful in low light situations. This allows for using shorter exposure times and faster line rates.

Input Planes	Summation	
 48 DN	2-Plane: 96 DN	
	3-Plane: 144 DN	
	4: Plane: 192 DN	

This option supports dual output of a “passthrough” (unprocessed or “dark” compared to the summed image) image of a selected input plane (camera output row) into the second plane of a Mono8P2 buffer.

- **Plane summation**: Only available for 3-plane 8-bit input. This effectively increases the exposure by combining the light gathered by multiple lines into one line to create a brighter image. This allows for using shorter exposure times and faster line rates instead of increasing the exposure time. The summed image is transferred to a 16-bit buffer. This option is used to avoid data loss when the summation of the 3-planes exceeds 8-bits (instead of saturating).
- **Plane averaging:** This is used to reduce random noise in the acquisition, increasing the signal-to-noise ratio. The effect of the averaging depends on the number of planes averaged (2, 3 or 4).

Input Planes Images	Averaged Result
	

For 3 and 4-plane input, this option supports ouput of an optional “passthrough” (unprocessed) image of a selected input plane (camera output row) into a second plane of a Mono8P2 buffer.

Support for 2 and 3-plane image processing functions is included as part of the standard Xtium3-CLHS PX8 feature set.

Support for 4-plane image processing functions require an Advanced Processing license; for information on obtaining and installing a license see Appendix E: Advanced Processing Licences.

TDI Vertical Spatial Correction

Complete information is available in the TDI Multi-Plane Acquisition section.

Overview

When acquiring multiiple planes from a TDI (Time Delay Integration) CLHS camera, the Xtium3-CLHS PX8 automatically applies a vertical spatial correction when starting the acquisition grab. This ensures that only lines that have accumulated charge in all the arrays (each plane) are output. Spatial correction is necessary when using multiple array output, such as when using cameras with HDR or high full well modes.

Flat Field Correction

For complete information see the Flat Field Correction Tool.

Overview

The Xtium3-CLHS PX8 supports hardware-based real-time Flat field correction (FFC). Flat field correction (for 1D cameras this is also referred to as Flat Line Correction (FLC)) eliminates small gain differences between pixels in a sensor, removes sensor hotspots by automatically doing pixel replacement, and also compensates for light distortion caused by a lens. Flat field correction data is composed of gain and offset coefficients for each pixel. These coefficients are saved as a *.tiff* file which can then be loaded by the frame grabber.

When a properly calibrated flat field correction is applied to the image, a sensor exposed to a uniformly lit field will have no graylevel differences between pixels.

Sapera LT's CamExpert includes a Flat Field Correction Tool to perform flat field (line) calibration to generate coefficients and save them to file.

Lookup Table (LUT)

For complete information on implementation see Lookup Table (LUT) Configuration.

Overview

The Xtium3 PX8 includes an on-board output look-up table LUT for mapping input pixel values to corresponding output pixel values, replacing computationally expensive operations (like mathematical transformations) with real-time hardware-based processing.

LUTs can adjust brightness, contrast, and gamma to improve visibility of features and enhance edges. LUTs can be used to quickly classify pixel values (for example, binary thresholding) or speed up segmentation tasks like separating objects from the background.

Sapera LT's CamExpert provides a number of standard LUTs and allows you to create custom LUTs; see Lookup Table (LUT) Configuration.

Horizontal & Vertical Flip Operations

For complete information on implementation using Sapera LT, see Horizontal and Vertical Flip.

Overview

The Xtium3 PX8 supports real-time on-board horizontal and vertical flip operations; performing these operations on the frame grabber frees the CPU for other processing tasks.

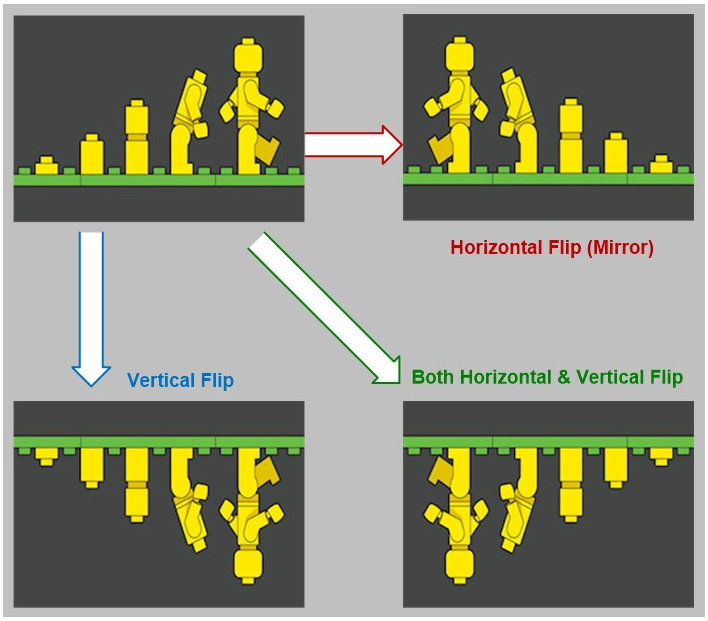


Figure 2: Horizontal and Vertical Flip Operations

Technical Specifications

Xtium3-CLHS Flow Diagram

The following diagram represents the processing sequence through the Xtium3-CLHS.

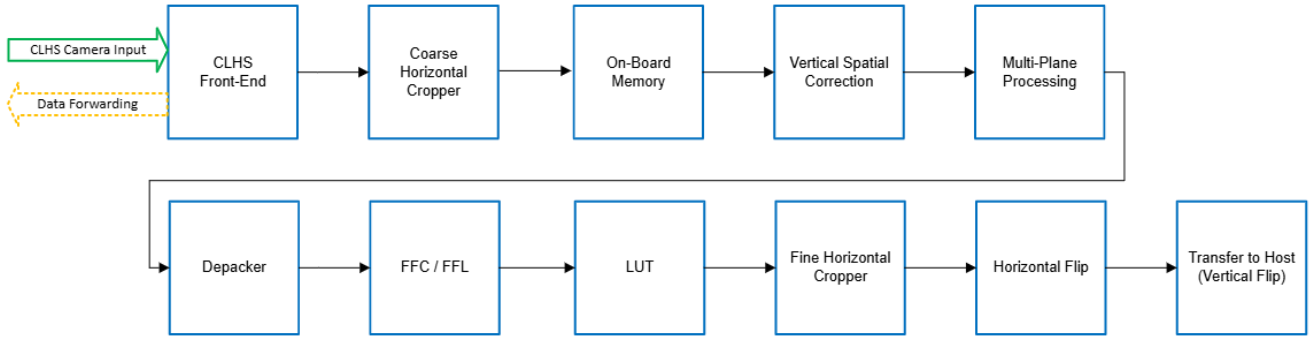


Figure 3: Xtium3-CLHS Flow Diagram

- **Camera Link HS Front End:** Extracts the image data packets from the Camera Link HS port.
- **Horizontal Cropper (Coarse):** Horizontal cropper used before writing into the memory. This aligned to 64-bits with the image width parameter setting.
- **On-Board-Memory:** Stores the image data.
- **Vertical Spatial Correction:** Depending on the line scan camera, a spatial correction might need to be applied by the frame grabber when acquiring multiple image planes. For more details, see TDI Acquisition: Vertical Spatial Correction.
- **Multi-Plane Processing:** Performs selected processing operations on multiple plane input images. For more details, see TDI Multi-Plane Image Processing.
- **Depacker:** Unpacks 10 and 12-bit packed pixel formats to 16-bits.
- **FFC/FLC:** Flat Field/Flat Line correction. Applies to Monochrome data only.
- **LUT:** Apply lookup table transformation to the image.
- **Horizontal Cropper (Fine):** Crops the resulting image when used, using a 4 pixel resolution.This specified using the image width parameter.
- **Horizontal Flip:** Performs the line data flip process.
- **Transfer to Host** (and optional vertical flip)**:** Transfers the data using direct memory access (DMA) to the host buffer memory. This module will also perform the vertical flip if enabled.

Connectors and Board Layout

Xtium3-CLHS PX8 Board Layout Drawing

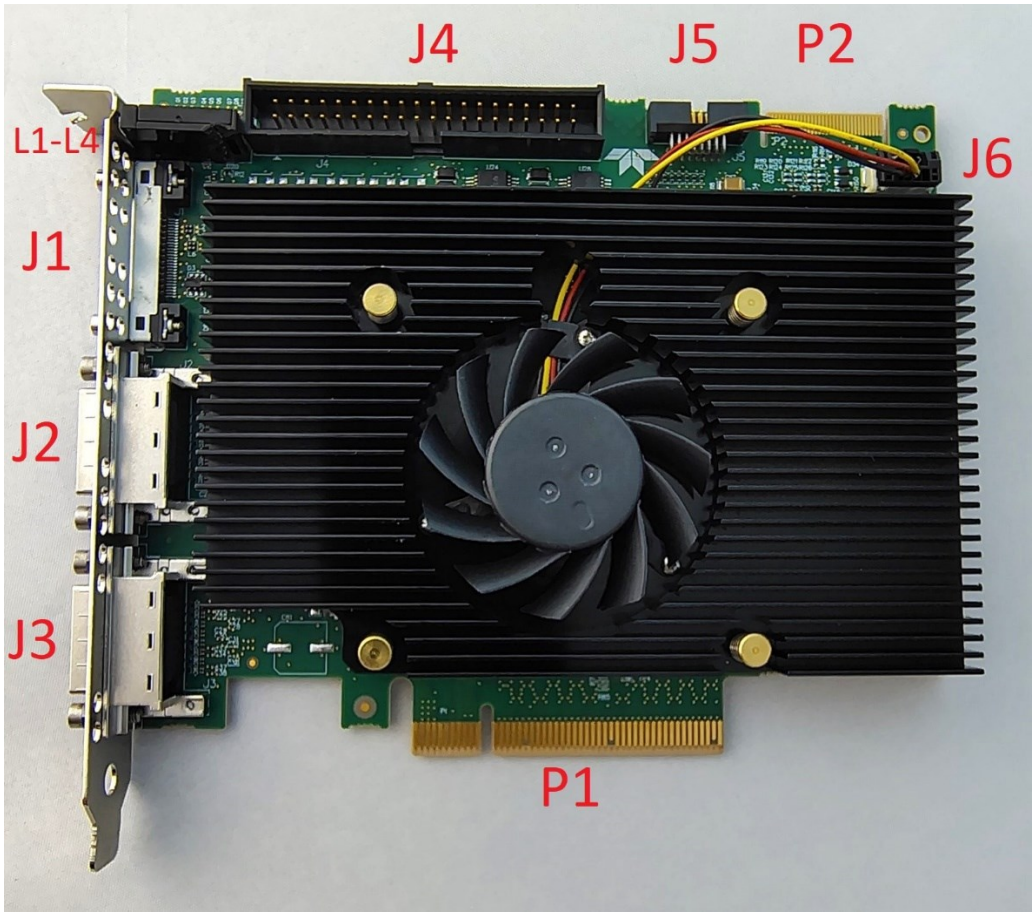


Figure 4: Board Layout

Connector / LED Description List

The following table lists components on the Xtium3-CLHS PX8 board. Detailed information concerning the connectors/LEDs follows this summary table.

Location	Description	Location	Description
J1	User Port: External I/O Signals connector	J4	Expansion Port: Internal I/O Signals connector
L1	Boot-up/PCIe Status LED (refer to text)	J5	Multi Board Sync: Board-to-Board I/O Signals connector
J2	Camera Link HS Input Connector	P1	PCIe x8 computer bus connector (Gen4 compliant slot preferred)
L2	Camera Link HS Input status LED	P2	Reserved
J3	Camera Link HS Output Connector (used for Data Forwarding)	J6	Power connector for fan
L3	Data Forwarding Output status LED		

Xtium3-CLHS PX8 End Bracket Detail

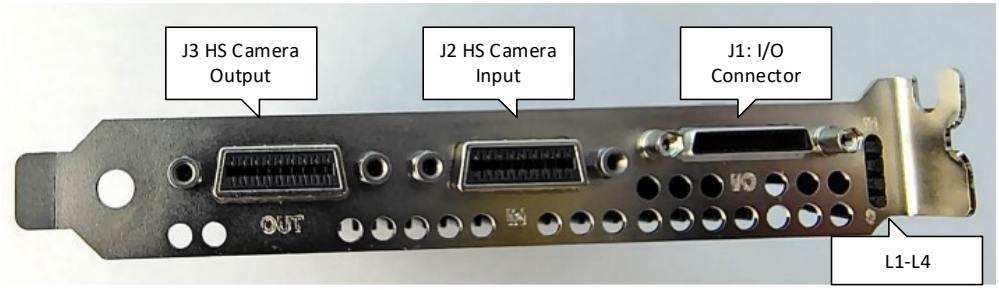


Figure 5: End Bracket Details

The hardware installation process completes with the connection of a supported camera to the Xtium3-CLHS PX8 board using a Camera Link HS cable (see the Camera Link HS Cables Overview and Resources section).

- The Xtium3-CLHS PX8 board supports one Camera Link HS camera output.
- Connect the camera to the J3 connector with a Camera Link HS cable.

Status LEDs Functional Descriptions

Three LED indicators, mounted on the board bracket, provide information on board and connection status as per the tables below.

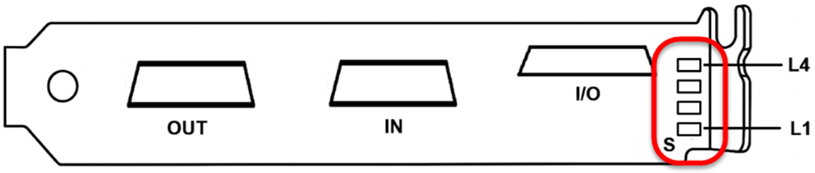


Figure 6: End Bracket Staus LEDs

L1: Boot-up/PCIe Status LED — Provides general boar status information

L2: CLHS status LED — Indicates data status for J2 when connected to a camera or when receiving forwarded data from another Xtium3-CLHS system.

L3: CLHS Forwarded Data status LED — Indicates connection status when the data forwarding connector J3 is connected to another Xtium3-CLHS. ▪

L4: Not Used.

L1: Boot-up/PCIe Status LED

Color	State	Description
Yellow		
Yellow		
Red	Solid	FPGA firmware not loaded
Green	Solid	Normal FPGA firmware loaded, Gen4 speed, link width x8
Green	Flashing	Normal FPGA firmware loaded, Gen1/Gen2/Gen3 speed, link width x8
	Solid	Normal FPGA firmware loaded, Gen4 speed, link width not x8
	Flashing	Normal FPGA firmware loaded, Gen1/Gen2/Gen3 speed, link width not x8
Blue	Solid	Safe FPGA firmware loaded, Gen4 speed
Blue	Flashing	Safe FPGA firmware loaded, Gen1/Gen2/Gen3 speed
Red	Flashing	PCIe Training Issue – Board will not be detected by computer

L2: Camera Link HS Status LED (when a camera is connected)

The Xtium3-CLHS PX8 implements the mandatory LED states defined by the Camera Link HS Specification v1.0 RC5. The first column – Priority, defines the signaling order when multiple events are reported simultaneously.

This LED status table reflects activity on input connector J2 when a camera is connected (that is, the Xtium3 is the Data Forwarding Master).

LED State	Description
Off	Device not powered and/or waiting for software.
Flashing Orange Medium ~2Hz	The devices have established communication and determined that they are not interoperable. This would be the case when the frame grabber and the camera have different pixel depth, pixel format and/or number of lanes.
Flashing Green Medium ~2Hz	Hardware is fine, but connection not established or recently broken.
Constant Green	Link established and data transfer may take place.
Flashing Green Fast ~4Hz	Acquisition in progress.

L2: Camera Link HS Status LED (as Data Forwarding receiver)

When the Xtium3-CLHS PX8 does not have a camera connected but is used as a Data Forwarding Slave, (that is, the board receives data on J2 from another Xtium3 in a separate computer), the LED L2 describes these two additional conditions.

LED Color	State	Description
Green	Flashing	Connection not established on input connector J2
Green	Constant	Link established on input connector J2 and data transfer may take place.

L3: Data Forwarding Status LED

L3 defines the connection status for when the Camera Link HS output J3 is forwarding acquisition data to another Xtium3-CLHS in a separate computer. Note that the Xtium3-CLHS PX8 data forwarding feature is not part of the standard CLHS specifications.

LED Color	State	Description
Off	Constant	Connection not established on input connector J3 of the receiving Xtium3-CLHS
Green	Constant	Link established with the receiving Xtium3-CLHS and data transfer may take place.

J1: External I/O Signals Connector

J1 user port allows the acquisition card to control and monitor the status of an external device. The J1 connector is located on the front bracket of the acquisition card.

The connector is a Hirose DH60-27P (female). The user interface connector is of the 'Screw lock' type and can withstand up to 1000 cycles. Each contact can handle a maximum current of 500mA.

This connector can accommodate two versions of male connectors for wiring purposes:

- DH30B-27S: version with terminals that accept a flat cable
- DH40-27S: version with terminals that accept a soldered cable

J1 includes the following I/Os:

- 4 Inputs (RS-422 (LVDS), TTL (LVTTL), 12V and 24V)
- 8 Outputs (4 LVTTL, 4 Open Collector)
- 1 dual phase Rotary Encoder (A and B)
- 2 power outputs (+5V and +12V)

WARNING

J1 and J4 have the same signal assignments. Signals are routed to both connectors directly from their internal circuitry. Therefore, never connect both J1 and J4 to external devices at the same time.

See the following sections for more details:

- Using J1: Connecting External Drivers to General Inputs
- Using J1: Connecting External Receivers to the General Outputs
- Using J1: Interfacing to an RS-422 Driver Output
- Input Specifications
- Output Specifications

J1 Pinout

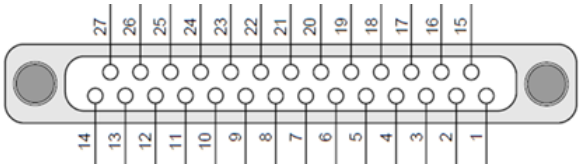


Table 1: DH60-27P Connector Signals

Pin	Description
1	Ground
2	RS-422 Rotary (Shaft) Encoder Phase A (-) / External Line Trigger (see Note 1)
3	LVTTL/RS-422 Rotary (Shaft) Encoder Phase A (+) / External Line Trigger (see Note 1)
4	Ground
5	RS-422 Rotary (Shaft) Encoder Phase B (-) / External Line Trigger (see Note 1)
6	LVTTL/RS-422 Rotary (Shaft) Encoder Phase B (+) / External Line Trigger (see Note 1)
7	External Frame Trigger Input 1/General Input 1 (-) (see Note 2)
8	External Frame Trigger Input 1/General Input 1 (+) (see Note 2)
9	External Frame Trigger Input 2/General Input 2 (+) (see Note 2)
10	Ground
11	Strobe 1 */ General Output 1 (LVTTL) (see Note 3)
12	Strobe 2 */ General Output 2 (LVTTL) (see Note 3)
13	Ground
14	Power Output 12 Volts, 450mA max
15	General Input 3 (+)
16	General Input 4 (+)
17	General Input 4 (-)
18	General Input 3 (-)
19	Power Output 5 Volts, 250mA max
20	External Frame Trigger Input 2/ General Input 2 (-) (see Note 2)
21	General Output 3 (LVTTL)
22	General Output 4 (LVTTL)
23	Strobe 5 */ General Output 5 (Open Collector) (see Note 3)
24	Strobe 6 */ General Output 6 (Open Collector) (see Note 3)
25	General Output 7 (Open Collector)
26	General Output 8 (Open Collector)
27	NC

Note 1: Assigned to either encoder or external line trigger (mutually exclusive). The external line trigger allows for exposure control using the pulse width.

Note 2: By default, General Input 1 and General Input 2 are reserved as External Frame Trigger Inputs. To use these inputs for purposes other than external frame triggers, the “reserved” status can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

Note 3: By default, General Output 1 is reserved as a Strobe Ouput. The “reserved” status for outputs to be used as Strobe Outputs can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

Using J1: Connecting External Drivers to General Inputs

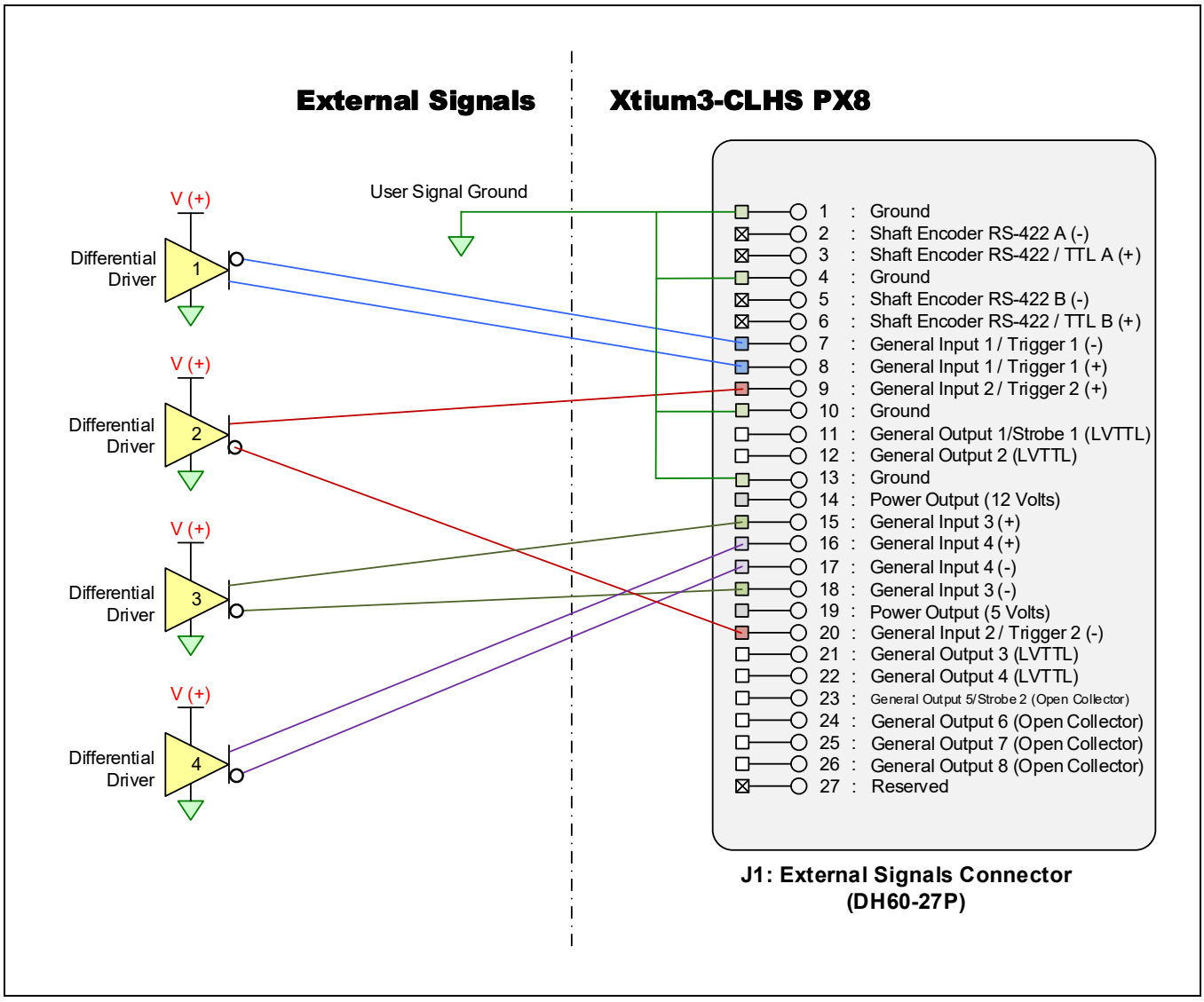


Figure 7:External Signals to J1 Connection Diagram

Using J1: Connecting External Receivers to the General Outputs

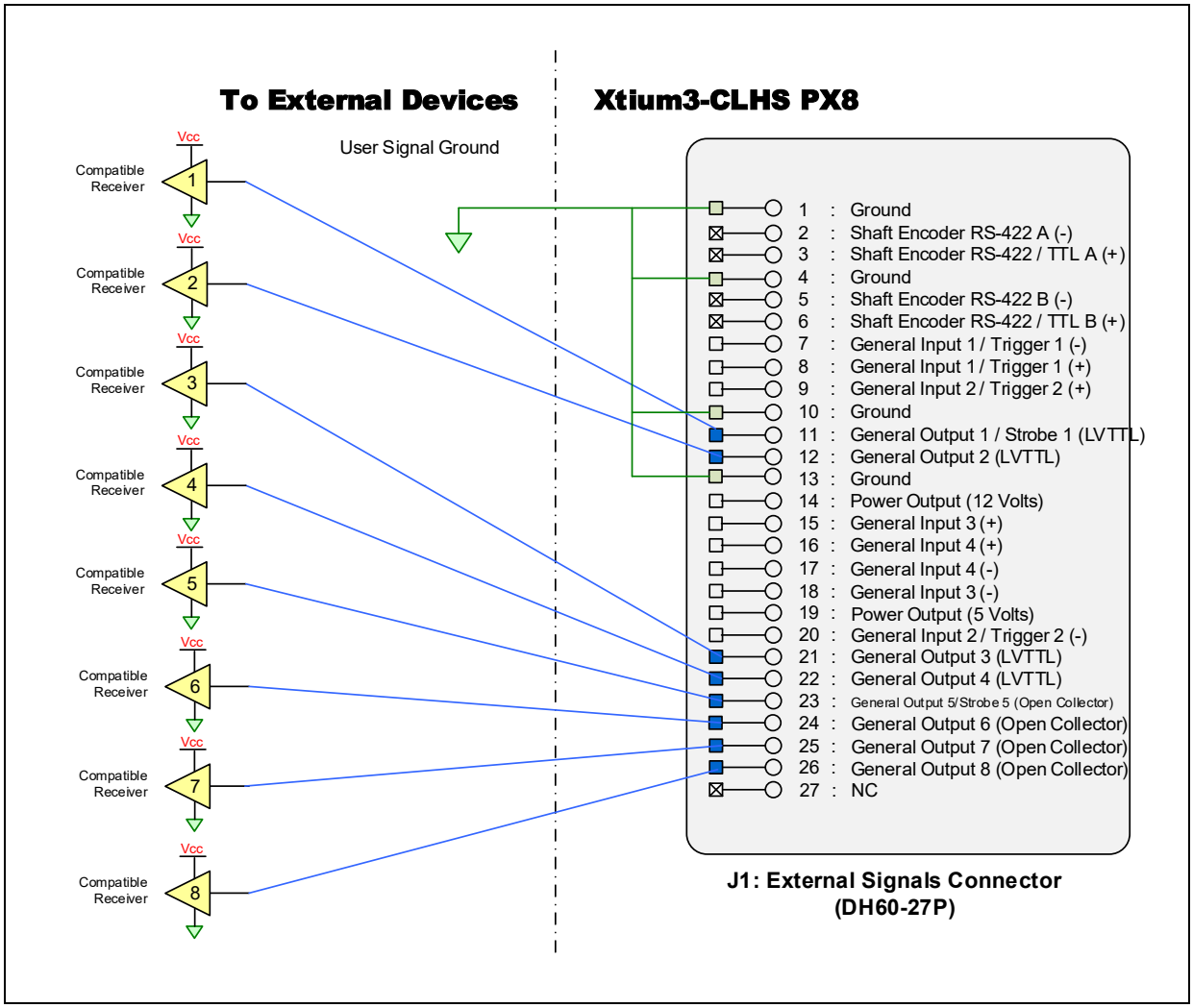


Figure 8:Output Signals to J1 Connection Diagram

Using J1: Interfacing to an RS-422 Driver Output

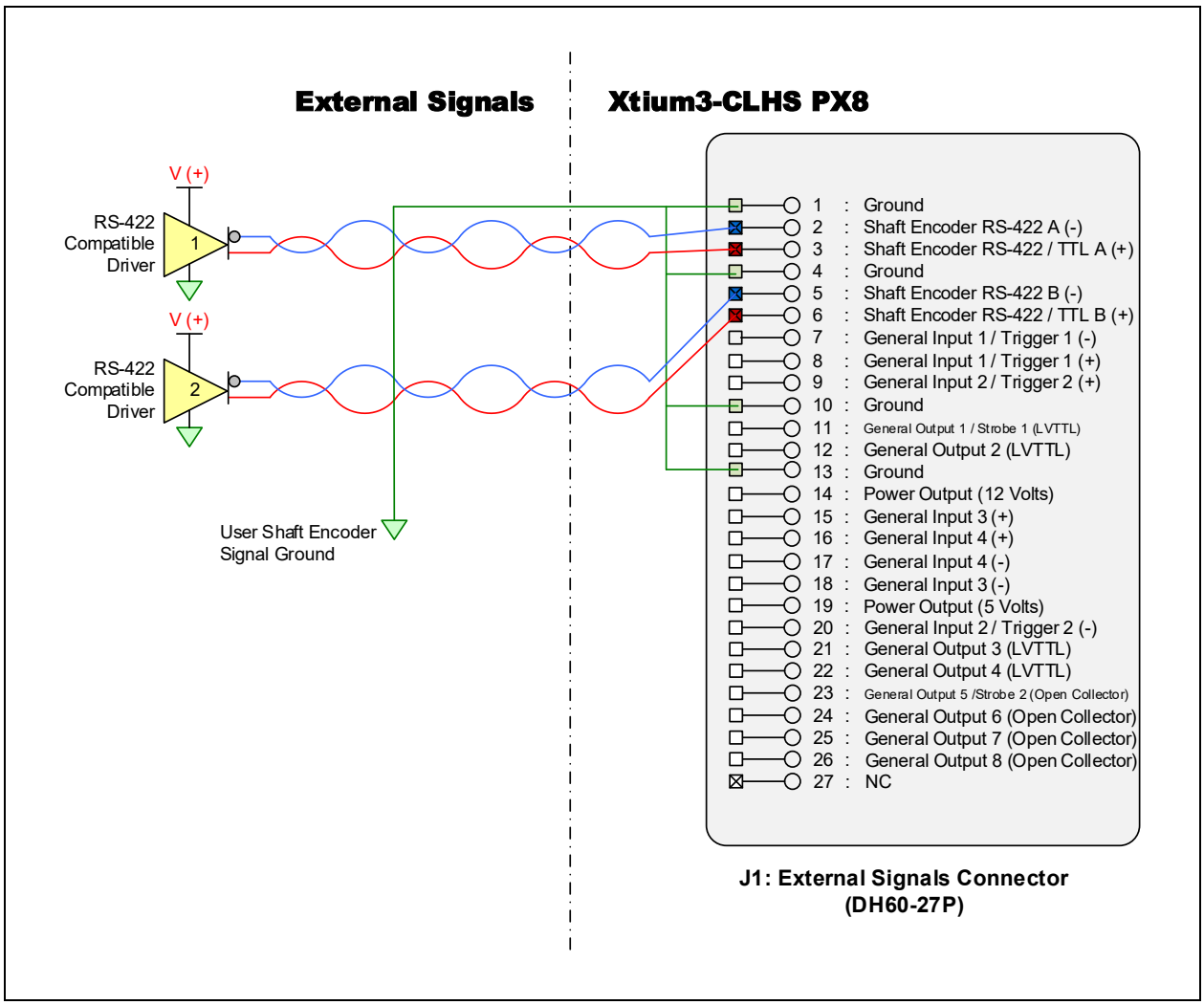


Figure 9:External RS-422 Signals to J1 Connection Diagram

J2: Camera Link HS Connector

NOTE

The Camera Link HS camera connector is defined in the AIA document “Specifications of the Camera Link HS Interface Standard for Digital Cameras and Frame Grabbers” version 1.2, ©2022 AIA.

Typically there is no need to be concerned with the physical pinout of the connector or cables. Refer to their site https://www.automate.org/ for additional information.

J3: Data Forwarding Connector

NOTE

The Data Forwarding connector on the Xtium3-CLHS PX8 has the same specifications as the Camera Link HS camera connector (J2) defined in the AIA document “*Specifications of the Camera Link HS Interface Standard for Digital Cameras and Frame Grabbers*” *version 1.2*, ©2022 AIA.

Typically there is no need to be concerned with the physical pinout of the connector or cables. Refer to their site https://www.automate.org/ for additional information.

J4: Internal I/O Signals Connector

WARNING

J1 and J4 have the same signal assignments. Signals are routed to both connectors directly from their internal circuitry. Therefore, never connect both J1 and J4 to external devices at the same time.

The J4 expansion port uses the same signals as the J1 user port but also includes:

- 2 additional inputs
- 4 additional open collector outputs.

The expansion interface uses a 40-pin (20x2) connector (TST-120-01-G-D) and allows connection of a standard 100TH flat cable that is easy to assemble.

The J4 expansion port includes the following I/Os:

- 6 Inputs (RS-422 (LVDS), TTL (LVTTL), 12V and 24V)
- 12 Outputs (4 LVTTL, 8 Open Collector)
- 1 dual phase Rotary Encoder (A and B)
- 2 power outputs (+5V and +12V)

See the following sections for more details:

- Using J4: Connecting External Drivers to General Inputs
- Using J4: Connecting External Receivers to the General Outputs
- Using J4: Interfacing to an RS-422 Driver Output
- Input Specifications
- Output Specifications

J4 Pinout

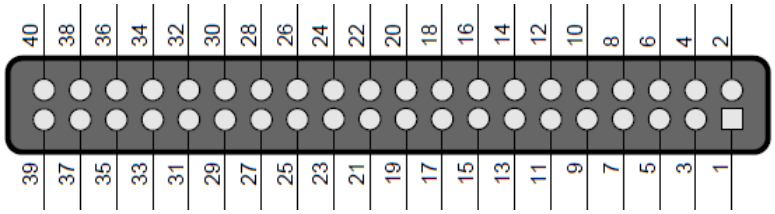


Table 2: 40-pin TST-120-01-G-D Connector Signals

Pin	Description	Pin	Description
1	Power Output 5 Volts, 250mA max	21	External Frame Trigger Input 1/General Input 1 (+) (see Note 2)
2	Power Output 12 Volts, 450mA max	22	External Frame Trigger Input 1/General Input 1 (-) (see Note 2)
3	Ground	23	External Frame Trigger Input 2/General Input 2 (+) (see Note 2)
4	Ground	24	External Frame Trigger Input 2/General Input 2 (-) (see Note 2)
5	LVTTL/RS-422 Encoder Phase A (+) / E xternal Line Trigger (see Note 1)	25	General Input 3 (+)
6	RS-422 Encoder Phase A (-) / External Line Trigger (see Note 1)	26	General Input 3 (-)
7	LVTTL/RS-422 Encoder Phase B (+) / External Line Trigger (see Note 1)	27	General Input 4 (+)
8	RS-422 Encoder Phase B (-) / External Line Trigger (see Note 1)	28	General Input 4 (-)
9	Ground	29	General Input 5 (+)
10	Ground	30	General Input 5 (-)
11	Strobe 1 / General Output 1 (LVTTL) (see Note 3)	31	General Input 6 (+)
12	Strobe 2 / General Output 2 (LVTTL) (see Note 3)	32	General Input 6 (-)
13	General Output 3 (LVTTL)	33	General Output 9 (Open Collector)
14	General Output 4 (LVTTL)	34	General Output 10 (Open Collector)
15	Strobe 5 / General Output 5 (Open Collector) (see Note 3)	35	General Output 11 (Open Collector)
16	Strobe 6 / General Output 6 (Open Collector) (see Note 3)	36	General Output 12 (Open Collector)
17	General Output 7 (Open Collector)	37	NC
18	General Output 8 (Open Collector)	38	NC
19	Ground	39	Ground
20	Ground	40	Ground

Note 1: Assigned to either encoder or external line trigger (mutually exclusive). The external line trigger allows for exposure control using the pulse width.

Note 2: By default, General Input 1 and General Input 2 are reserved as External Frame Trigger Inputs. To use these inputs for purposes other than external frame triggers, the “reserved” status can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

Note 3: By default, General Output 1 is reserved as a Strobe Ouput. The “reserved” status for outputs to be used as Strobe Outputs can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

Using J4: Connecting External Drivers to General Inputs

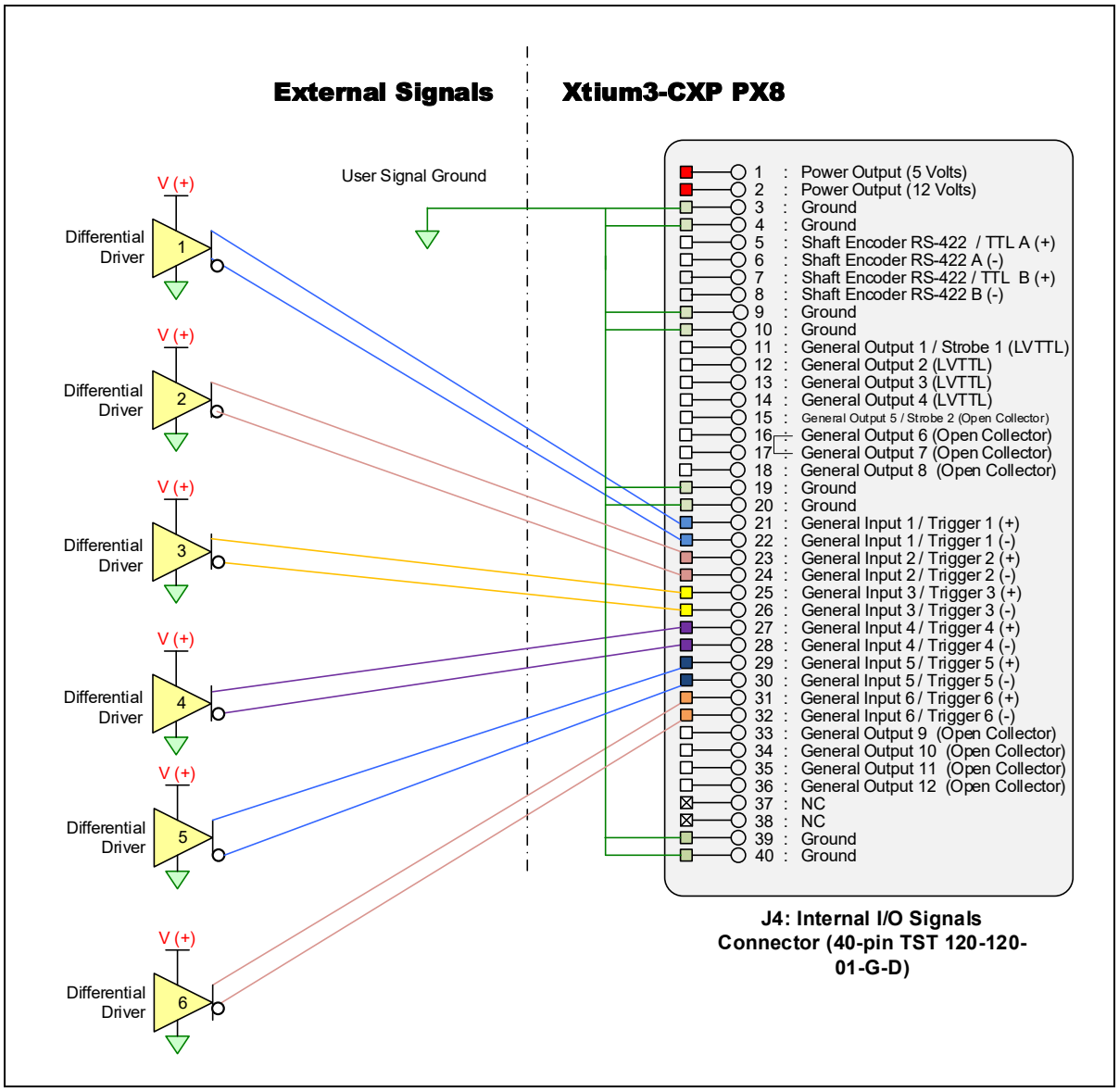


Figure 10: External Signals to J4 Connection Diagram

Using J4: Connecting External Receivers to the General Outputs

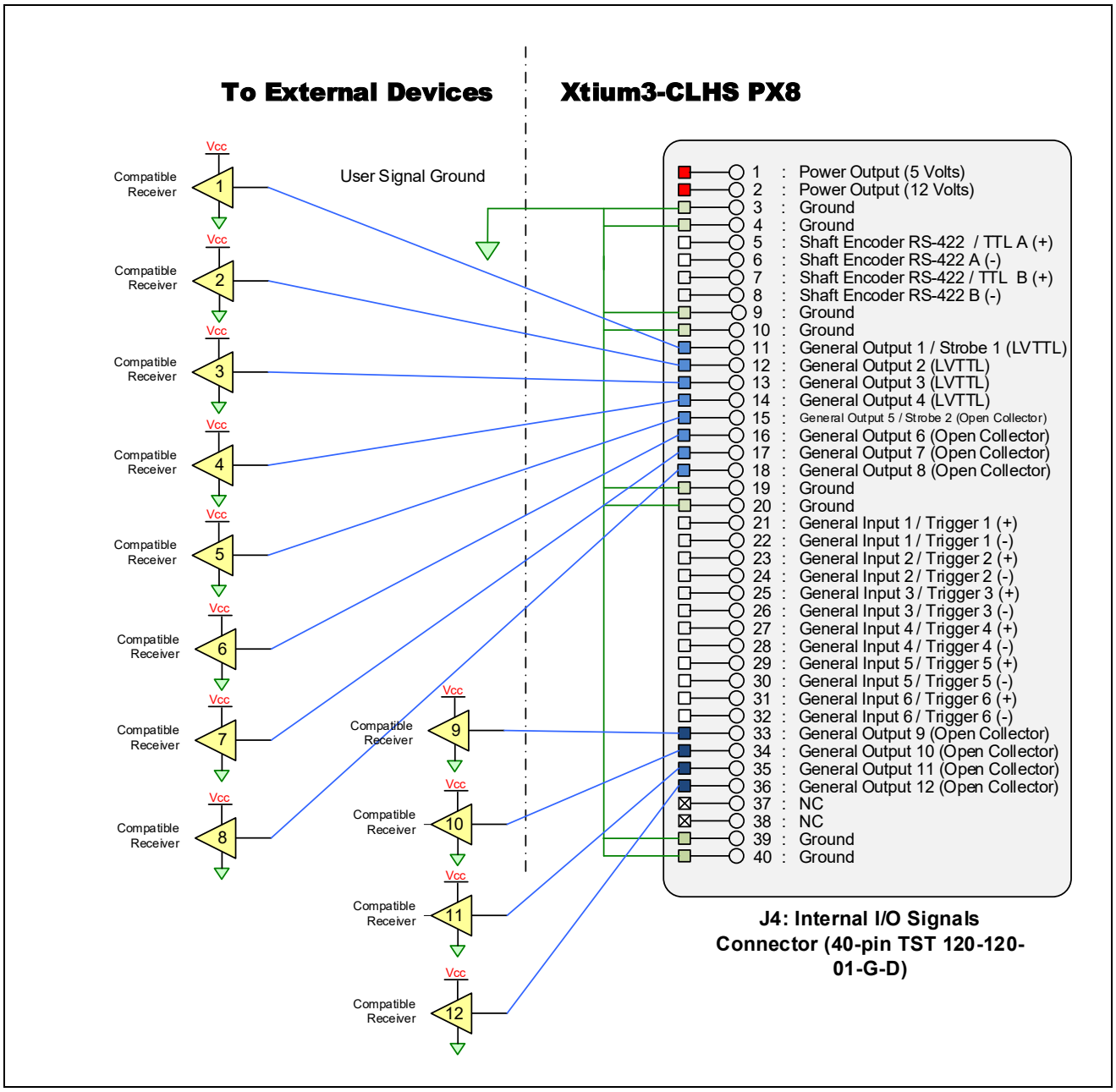


Figure 11:Output Signals to J4 Connection Diagram

Using J4: Interfacing to an RS-422 Driver Output

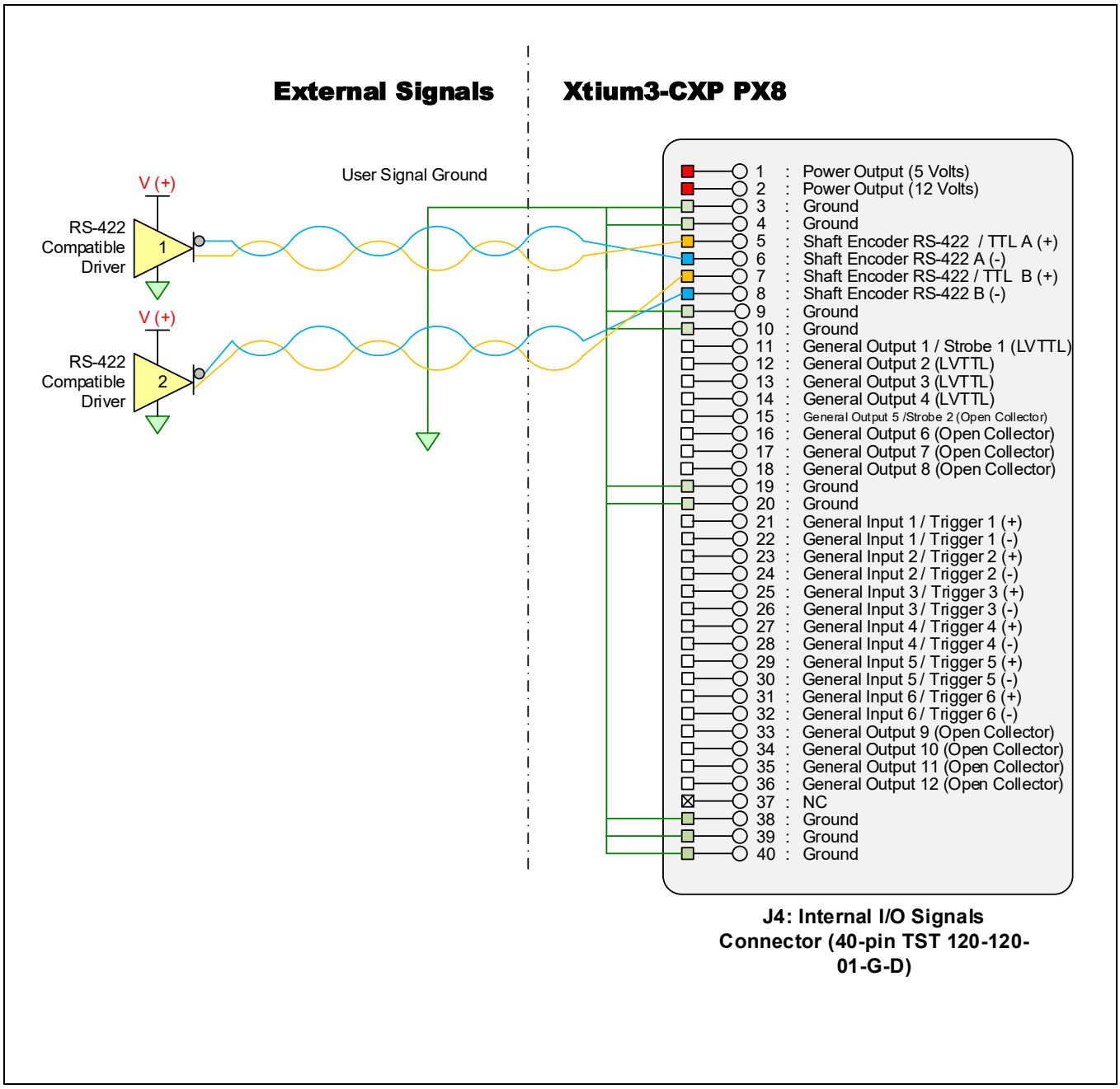


Figure 12: Output Signals to J4 Connection Diagram

J5: Multi-Board Sync / Board-to-Board Bi-directional I/Os

There are 8 bi-directional board-to-board I/Os that can be interconnected between multiple boards.

Interconnect up to 6 Xtium3 boards via their J5 connector using:

- OR-YXCC-BSYNC20 cable for 2 boards
- OR-YXCC-BSYNC40 cable for 3 to 4 boards
- OR-YXCC-BSYNC60 cable for 5 to 6 boards

See Board Sync Cable Assembly.

WARNING
Multi-Board Sync / Bi-directional board-to-board I/Os are only for use with Teledyne DALSA frame grabbers within the same PC, otherwise electrical damage to boards can occur.

For setup information, see Multi-board Sync & I/O Setup.

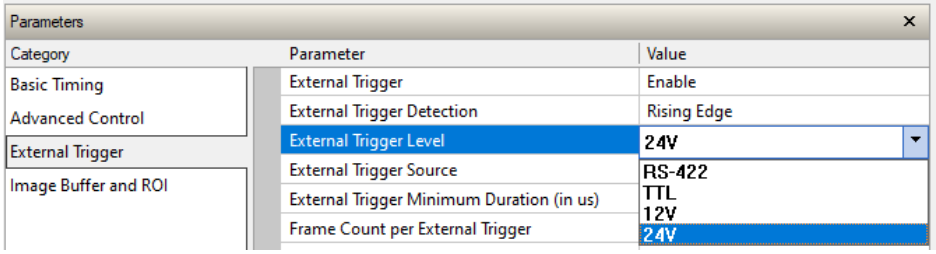
Input Electrical Specifications

General Purpose Inputs

Each of the General Purpose Inputs (GPI) are opto-coupled and able to connect to differential or single ended source signals.

The GPI interface is an isolated interface. This interface supports LVTTL and TTL logic levels. It is also compatible with 12V and 24V environments. This interface can also operate in differential mode compatible with RS-422 (TIA/EIA-422-B standard). The Xtium3-CLHS allows user selected (software programmable) input switching points to support differential (RS-422 (LVDS)) input signals and single ended (TTL (LVTTL), 12V or 24V) input signals.

For example, in CamExpert, this is set using the External Trigger Level parameter



The following diagram shows the simplified circuitry of the GPI interface:

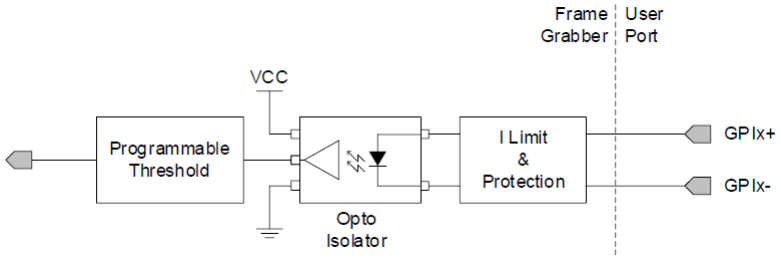


Figure 13: General Inputs Electrical Diagram

External Trigger Inputs Specifications

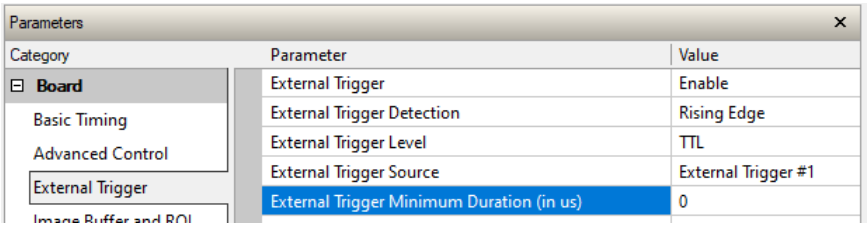
Table 3: External Trigger Input Details

Specification	Definition	Min	Max	External Trigger Level
V_{IH}	High Level Input Voltage (GPI_Px+ - GPI_Px-1)	2V	26V	RS-422 (LVDS) / TTL (LVTTL)
		8V	26V	12V / 24V
V_{IL}	Low Level Input Voltage (GPI_Px+ - GPI_Px-1)	-	0.8V	RS-422 (LVDS) / TTL (LVTTL)
		-	4V	12V / 24V
F_I	Input Frequency (Duty Cycle = 50%0	-	8 MHz	RS-422 (LVDS) / TTL (LVTTL)
		-	2 MHz	12V / 24V

External Trigger Debouncing

Input signal is “debounced” to ensure that no voltage glitch is detected as a valid transition. This debounce circuit time constant can be programmed from 1µs to 255µs. Any pulse smaller than the programmed value is blocked and therefore not seen by the board. If no debounce value is specified (value of 0µs), the minimum value of 1µs will be used.

The debouncing value can be set in CamExpert using the External Trigger Minimum Duration (in µs) feature, available in the External Trigger category,



External Frame Triggers

By default, General Input 1 and 2 are reserved as External Trigger Inputs; as such, these external frame trigger inputs generate individual interrupts and are read by the Sapera application; see the Supported Events and Transfer Methods. External Trigger Input 2 can be used for two pulse external trigger with variable frame length line scan acquisition.

To use these inputs for purposes other than external frame triggers, the “reserved” status can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

Related 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

Sapera LT includes an I/O Demo that demontrates how to use these inputs.

Also see Sapera Servers and Resources and Supported Events and Transfer Methods.

Rotary (Shaft) Encoder/ External Line Trigger Input Specifications

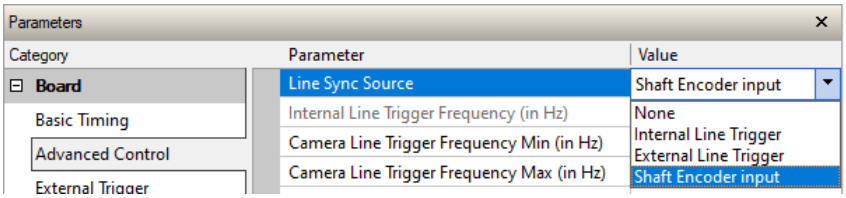
The encoder interface supports a one (single) or two (dual) phase encoder (A and/or B). Dual encoder signals are typically 90 degrees out of phase relative to each other and provide greater encoder motion resolution.

Encoder operating modes:

- Differential (TIA/EIA RS-422 standard)
- Single-ended LVTTL

When used with an external line trigger it allows for exposure control using the pulse width. The external line trigger is connected to either input A or B and can use the rising or falling edge.

In Sapera LT, this is set using the Line Sync Source parameter available in the Advanced Control category.



See J1: External I/O Signals Connector or J4: Internal I/O Signals Connector for connector details.

The Xtium3-CLHS provides ESD filtering on-board. Both inputs have a 100-ohm differential resistor.

WARNING

The encoder interface is not isolated, and the GND must be connected between the external device and the acquisition board.

When connecting a rotary encoder to the Xtium3-CLHS PX8, make sure to connect a common ground between the encoder and the frame grabber. The encoder ground and the Xtium3-CLHS PX8 computer system ground must be connected together. See RED boxed connections in the diagram below.

Failure to follow the described instructions could damage the board resulting in the shaft encoder functionality not working properly.

Ensure that these grounding measures are followed when migrating from boards with opto-coupled shaft encoders (such as the Xcelera).

The figure below shows the simplified representation of these inputs.

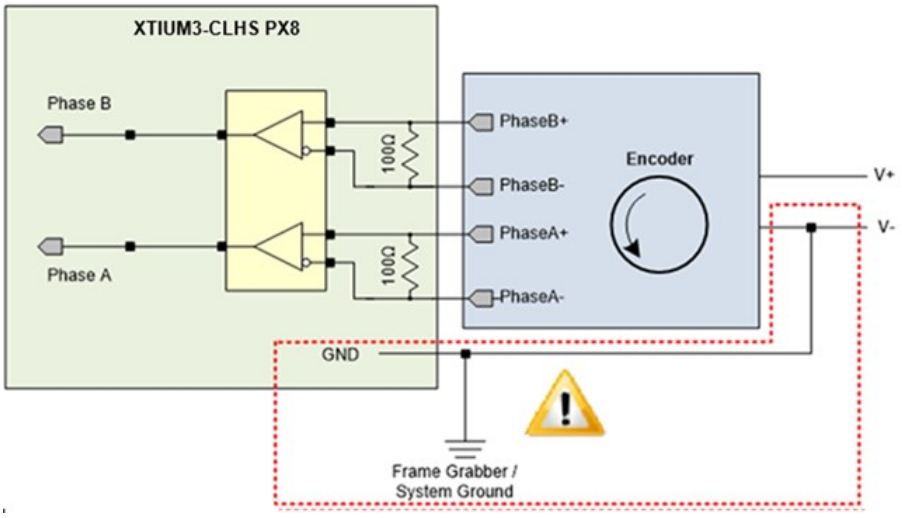


Figure 14: Encoder Input Electrical Diagram

Table 4: Encoder Input Details (TI AM26LV32E)

Specification			Min	Max
RS-422	V_{ICM}	Common-mode input voltage	-7V	7V
	V_{ID}	Differential input voltage	-0.2V	7V
	Input frequency		-	8 MHz
LVTTL	V_{IH}	High level input voltage	2V	7V
	V_{ID}	Low level input voltage	-	0.8V
	Input frequency		-	8 MHz

Interfacing to a Line Driver (Open Emitter) Output

User must select the Encoder RS-422 level when using this mode.

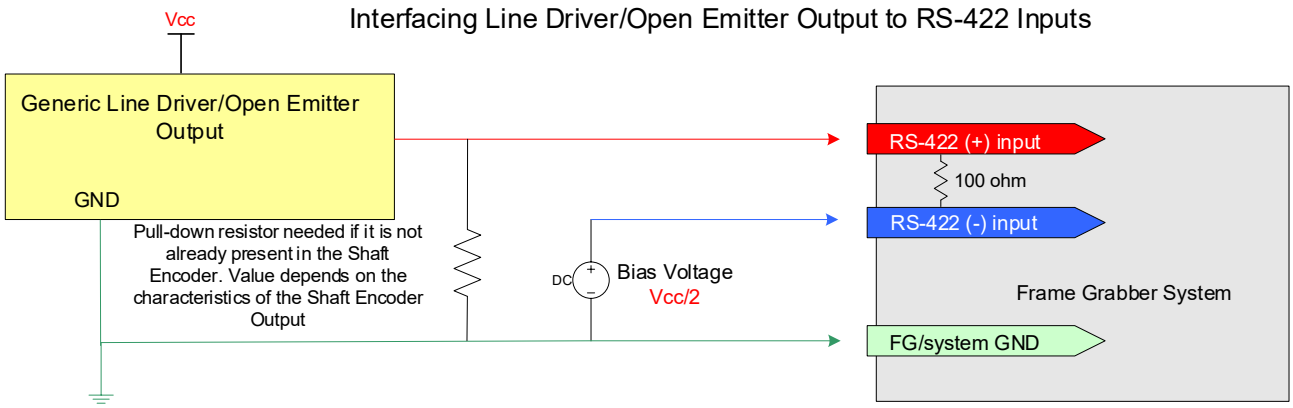


Figure 15: Interfacing to a Line Driver Output

Interfacing to an Open Collector Output

User must select the Encoder RS-422 level when using this mode.

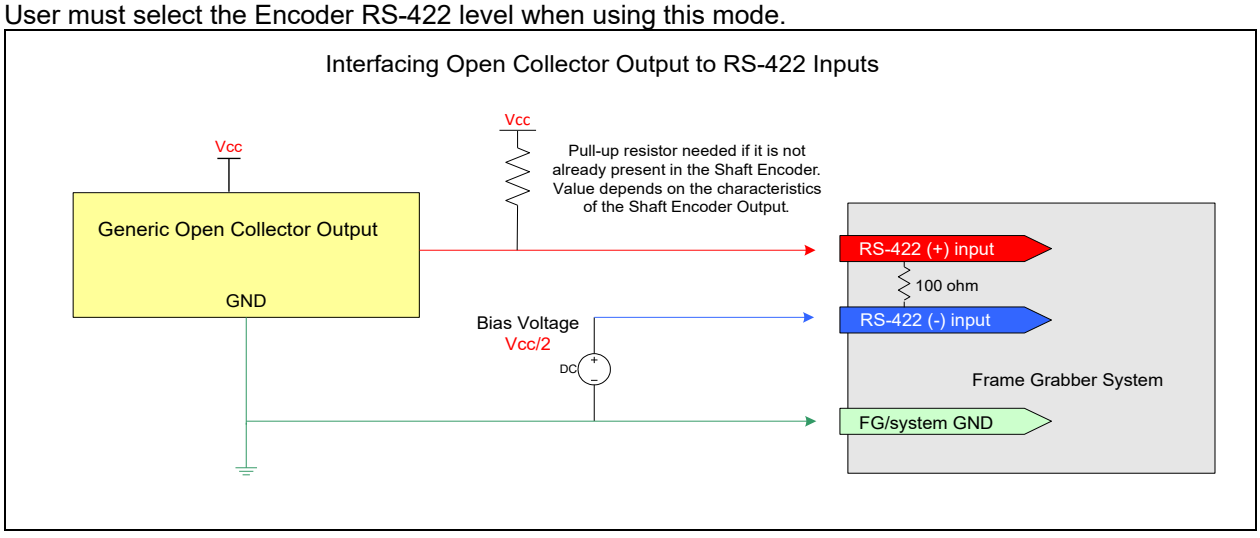


Figure 16: Interfacing to an Open Collector Output

Interfacing directly to a TTL (also called Push-Pull) Output

User must select the Encoder TTL level when using this mode.

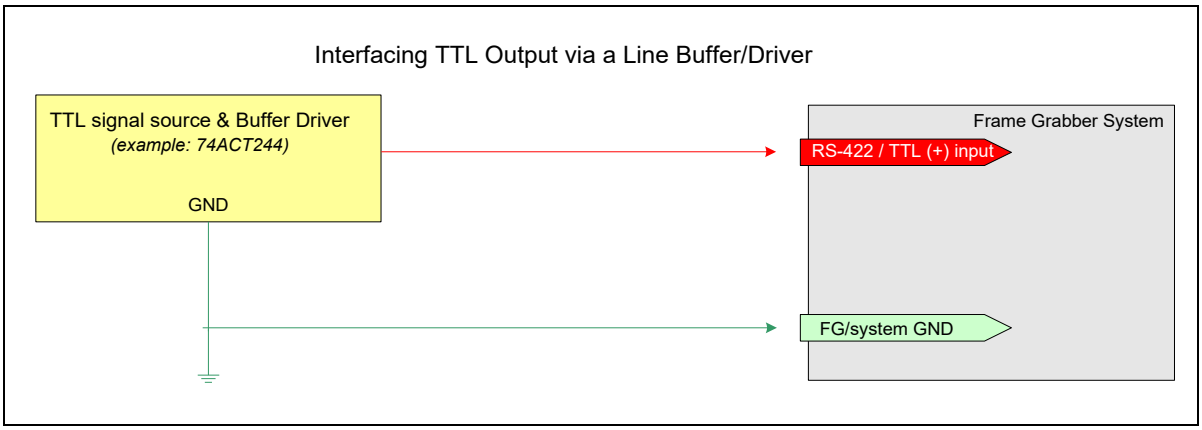


Figure 17: Interfacing TTL to TTL Shaft Encoder Inputs

Related Sapera Parameters

CORACQ_PRM_SHAFT_ENCODER_ENABLE
CORACQ_PRM_SHAFT_ENCODER_LEVEL
CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE
CORACQ_PRM_EXT_TRIGGER_SOURCE

Sapera LT also supports encoder drop and multiply operations, encoder averaging engine and encoder direction and count modes ; see Encoder Interface Timing, Encoder Averaging Engine and Shaft Encoder Direction and Count modes.

Output Electrical Specifications

LVTTL Output Specifications

The 4 LVTTL General Purpose Output (GPO) interface provides a LVTTL-level Push-Pull output. Each output can supply a continuous current of 24 mA. When the frame grabber is powered on, all outputs are in high impedance.

There are 4 LVTTL (3.3V) compatible General Outputs (Output 1 to 4).

The following figure is typical for each General Output.

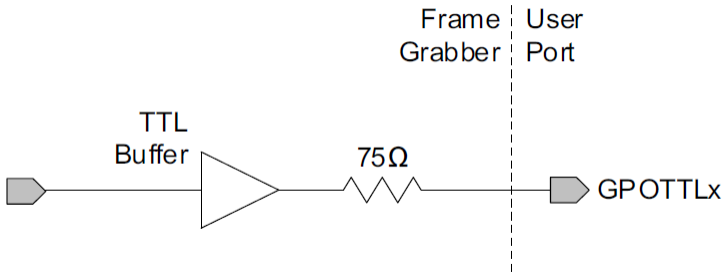


Figure 18: General Outputs LVTTL Electrical Diagram

External Receiver Electrical Requirements

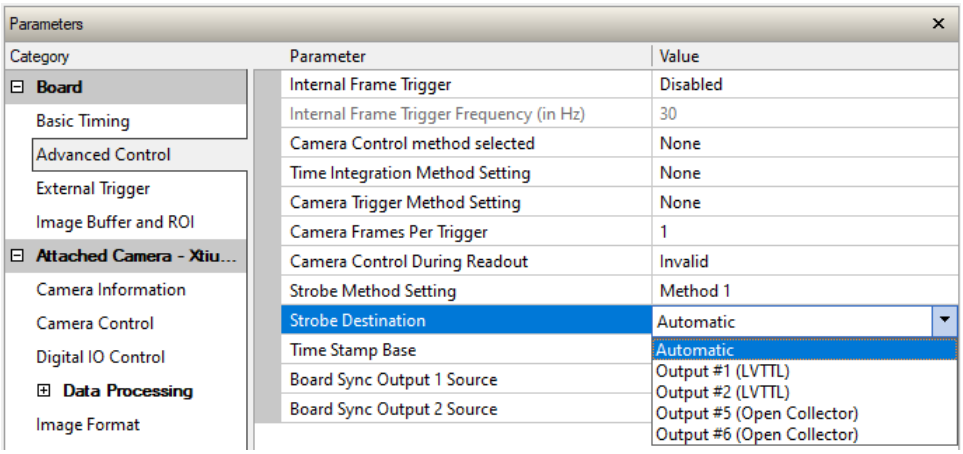
- Xtium3 General Outputs are standard LVTTL logic levels.
- External receiver circuits must be compatible to LVTTL signals.

Xtium3 PX8 Output Level	Description	MIN	MAX	
LVTTL	Output Voltage High (V_{OH})	2.0 V	–	
	Output Voltage Low (V_{OL})	–	0.8 V	
	Output Drive	0	24 mA	
	Output Frequency		10 MHz 10 MHz	RL = 280 (Iout = 12mA); CL = 0.01uF RL = 140 (Iout = 24mA); CL = 0.01uF

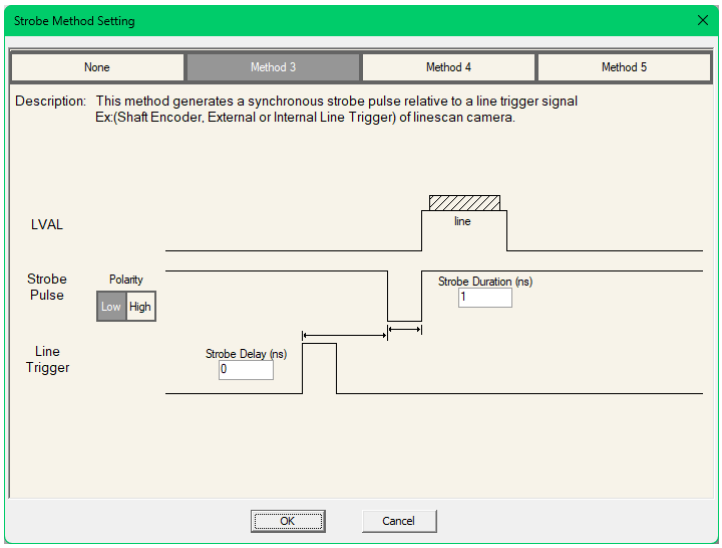
Strobe Outputs

General Output 1 and General Output 2 can also be used as Strobe Outputs controlled by Sapera strobe control functions. By default, General Output 1 is reserved as a Stobe output; to use this output for purposes other than a strobe, the “reserved” status can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

For example, in the CamExpert Advanced Control category, these outputs are available to be assigned as the Strobe Destination when a Strobe Method Setting is enabled.



For the Strobe Method Setting, CamExpert provides a graphical interface for configuration for each of the supported methods



Related Sapera Parameters

CORACQ_PRM_STROBE_ENABLE
CORACQ_PRM_STROBE_POLARITY
CORACQ_PRM_STROBE_LEVEL
CORACQ_PRM_STROBE_METHOD
CORACQ_PRM_STROBE_DELAY
CORACQ_PRM_STROBE_DURATION

Sapera LT includes an I/O Demo that demontrates how to use these outputs.

Also see Sapera Servers and Resources and Supported Events and Transfer Methods.

Open Collector Output Specifications

There are four Open Collector General Outputs on J1 (Output 5 to 8) and eight on J4 (Output 5 to 12).

General Output 5 and General Output 6 can also be used as the Strobe Output controlled by Sapera strobe control functions. To reserve these outputs as Strobe outputs; the “reserved” status can be enabled using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

This interface offers an open-collector output. The component used is the TI-manufactured SN74LS07. This component can supply a continuous current of 40 mA at 26V. When the frame grabber is powered on, all outputs are in high impedance.

WARNING

The open collector is not an isolated interface; the common GND must be connected between the external device and the frame grabber.

The following figure is typical for each open collector output.

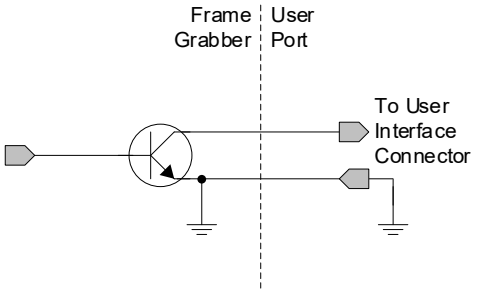


Figure 19: General Outputs Open Collector Electrical Diagram

Table 5: General Open Collector Output Details

Specification	Definition	Max	Notes
V_{OH}	High-level output voltage	26 V	
I_{OUT}	40 mA	40 mA	
F_O	Output Frequency	2 MHz	VI = 3.3V; RI = 300R; CI = 25pF
		5 MHz	VI = 3.3V; RI = 150R; CI = 25pF
		1 MHz	VI = 12V; RI = 1K; CI = 25pF
		2 MHz	VI = 12V; RI = 510R; CI = 25pF
		4 MHz	VI = 12V; RI = 300R; CI = 25pF
		0.6 MHz	VI = 24V; RI = 2K; CI = 25pF
		1 MHz	VI = 24V; RI = 1K; CI = 25pF
		2 MHz	VI = 24V; RI = 600R; CI = 25pF

WARNING

Without a resistor to limit the current for the output with a voltage higher than VCC, the IC is damaged/blown, and a voltage of about ~1V can be read at the output.

EMC Declarations of Conformity

Copies of the Declarations of Conformity documents are available on the product page on the Teledyne Vision Solutions websiteTeledyne DALSA website or by request.

FCC Statement of Conformance

This equipment complies with Part 15 of the FCC rules. Operation is subject to the following conditions:

- The product may not cause harmful interference; and
- The product must accept any interference received, including interference that may cause undesired operation.

FCC Class A Product

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their expense.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This equipment is intended to be a component of a larger industrial system.

EU and UKCA Declaration of Conformity

Teledyne Dalsa declares that this product complies with applicable standards and regulations.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This product is intended to be a component of a larger system and must be installed as per instructions to ensure compliance.

Acquiring Images

Basic Setup

NOTE

For the initial hardware and software installation see the Installing Xtium3-CLHS PX8 section.

Basic Acquisition Parameters

Basic acquisition parameters to verify:are:

- Camera Type
- Pixel Depth
- Data Lanes
- Horizontal Active
- Camera Sensor Geometry Setting

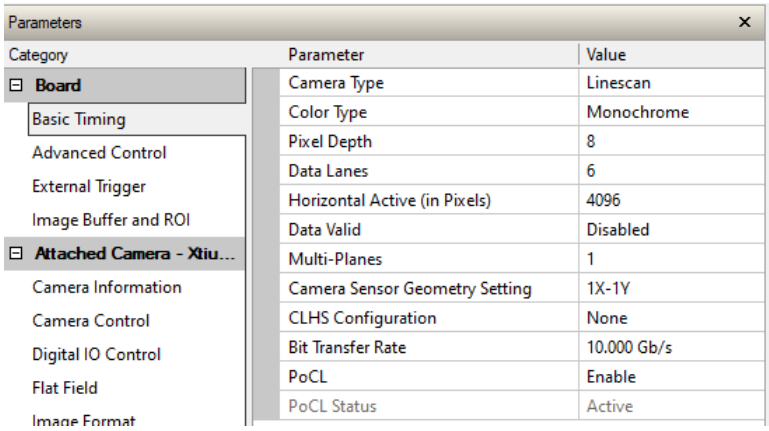
The Xtium3-PX8 CLHS acquires images generated from a connected CLHS camera; cameras are either areascan or linescan. This is set using the Camera Type parameter. Currently, only monochrome acquisiton is supported.

Areascan cameras output 2D image frames; linescan cameras output single 1D lines (certain TDI linescan cameras also use an areascan output for initial setup and alignment).

For the frame grabber, the number of horizontal active pixels should match the horizontal width of the lines output by the camera (this can be affected by the binning setting of the camera).

Verify that the Pixel Depth, Camera Sensor Geometry Setting and number of Data Lanes correspond to those output by the camera.

For example, in CamExpert, these parameters are available in the Basic Timing category:



When acquiring multi-plane images, set the Multi-Planes parameter to the number of planes; see Multi-Planes Image Processing for more information.

Related Sapera Parameters

CORACQ_PRM_SCAN, CORACQ_PRM_HACTIVE, CORACQ_PRM_MULTI_PLANES

TDI Multi-Plane Acquisition

When acquiring multiiple planes from a TDI (Time Delay Integration) CLHS camera, the Xtium3-CLHS PX8 automatically applies a vertical spatial correction when starting the acquisition grab. This ensures that only lines that have accumulated charge in all the arrays (each plane) are output. Spatial correction is necessary when using multiple array output, such as when using cameras with HDR or high full well modes.

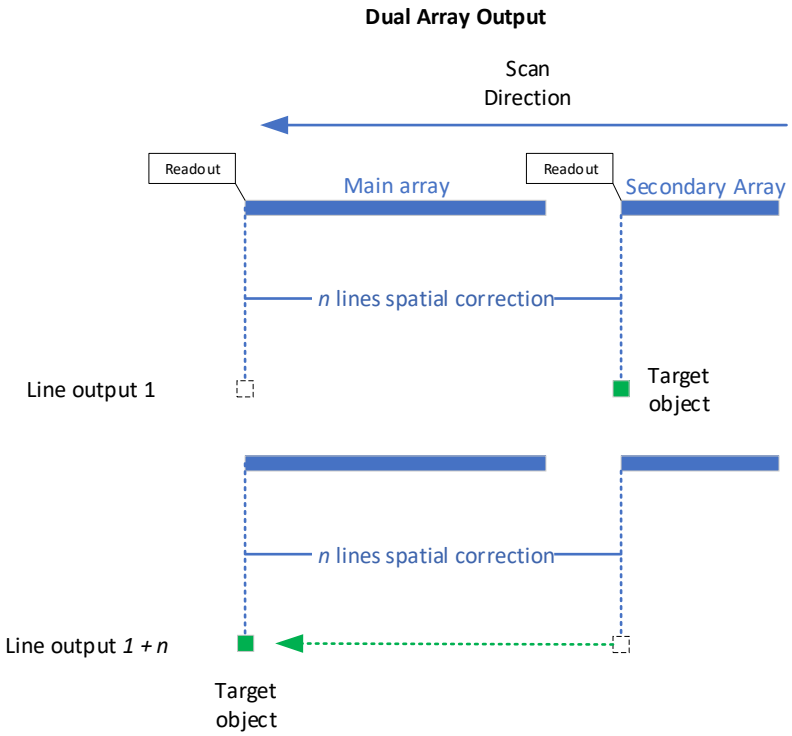


Figure 20: Spatial Correction

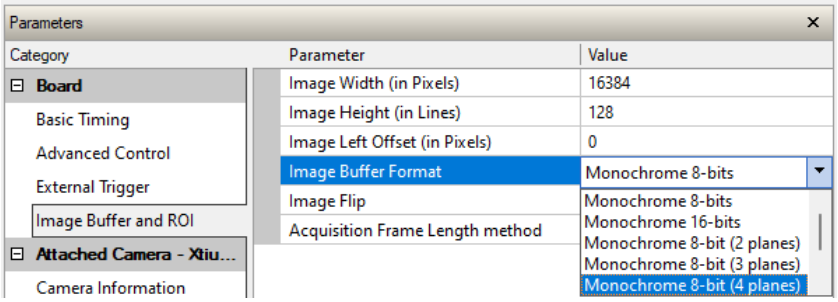
All data buffering is performed by the frame grabber as the camera does not have adequate memory resources for this function.

When a spatial correction is applied by the frame grabber, the distance in lines between the 1st and last plane (for example, distance “D”) must be taken into account when triggering camera lines at the start of the acquisition. For example, to get 1024 lines, the user needs to send an extra “D” lines (1024 + “D” triggers) to the camera in order to get 1024 lines in the host buffer. The actual number of extra lines is dependent on the scan direction.

The frame grabber then outputs a single line to a monochrome buffer (MONO8 or MONO16). This line can be a sum or average of the inputs, depending on the type of on-board processing selected; see the TDI Multi-Plane Image Processing section.

However, if required, each line can be preserved by setting the Image Buffer Format to the number of planes output by the camera.

For example, in CamExpert, the Image Buffer Format parameter is available in the Image Buffer and ROI category.



Refer to the camera documation for more details on the sensor array configuration and vertical spatial correction.

Each time the acquisition is stopped and restarted this must be taken into consideration.

NOTE

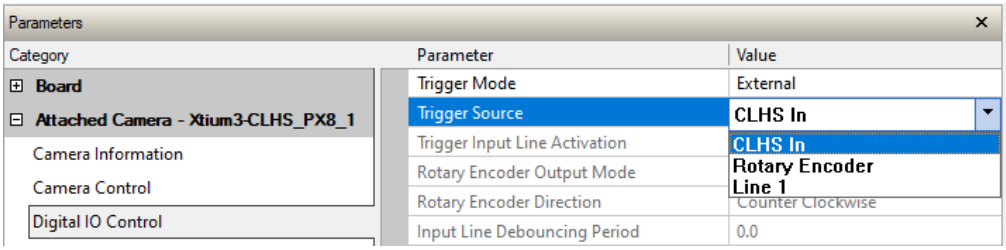
The frame grabber Multi-Planes parameter must be set to the number of planes to acquire to align data for the number of planes output by the camera (2, 3, or 4).

Related Sapera Parameters

CORACQ_PRM_OUTPUT_FORMAT, CORACQ_PRM_MULTI_PLANES

Using Triggers to Acquire Images

The camera outputs line (or frames) using an internal trigger or an external trigger. For example, in CamExpert, the attached camera's Trigger Mode and Trigger Source features determine the available options (refer to the camera documentation).



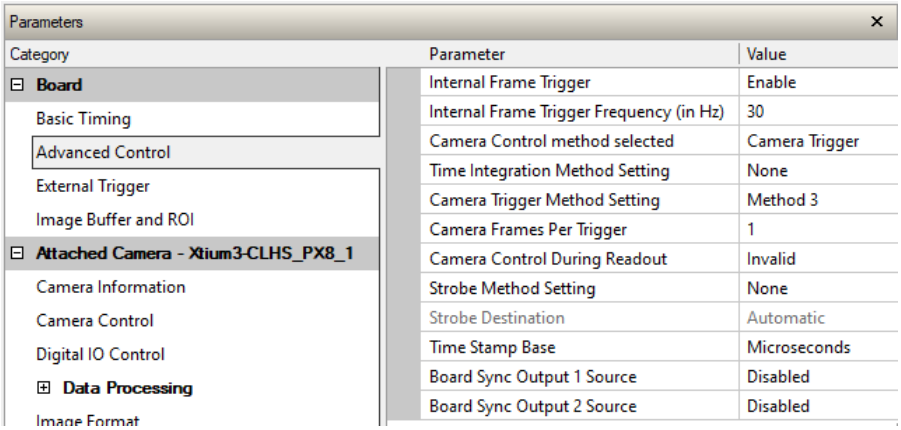
The external line trigger can be connected directly to camera inputs (if available) or to the frame grabber IO connector (J1 or J4) which sends the triggers to the camera through the CLHS cable. The frame grabber can also generate an internal trigger which can be sent to the camera through the CLHS channel.

For linescan cameras, a rotary encoder is often used to generate the line triggers. The encoder inputs can be connected to the camera inputs directly (if available) or to the frame grabber IO connector (J1 or J4) which sends the triggers to the camera through the CLHS cable.

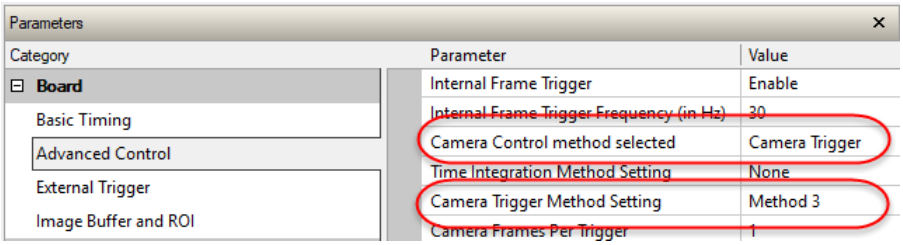
Area Scan Applications

With area scan cameras the frame grabber can generate internal frame triggers to send to the camera or use an external frame trigger.

In CamExpert, the internal frame trigger parameters are available in the Advanced Control category.

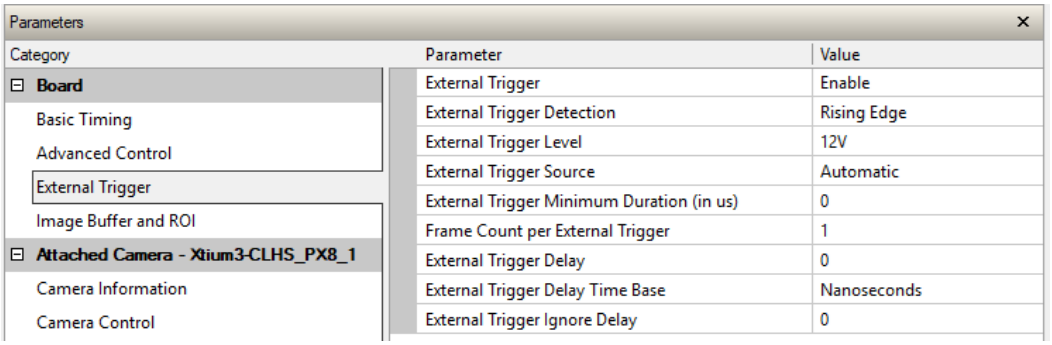


With the camera's Trigger Source is set to receive triggers from the framegrabber (CLHS In), to use the Internal Frame Trigger, enable it and set the Camera Control method to Camera Trigger with the Camera Trigger Method 3 .



If the camera supports Time Integration these methods can be used as well.

To use an external frame trigger connected to the frame grabber, use the parameters available in the External Trigger category.



Related Sapera Parameters

CORACQ_PRM_INT_FRAME_TRIGGER_ENABLE, CORACQ_PRM_INT_FRAME_TRIGGER_FREQ, CORACQ_PRM_CAM_TRIGGER_METHOD, CORACQ_PRM_EXT_TRIGGER_ENABLE

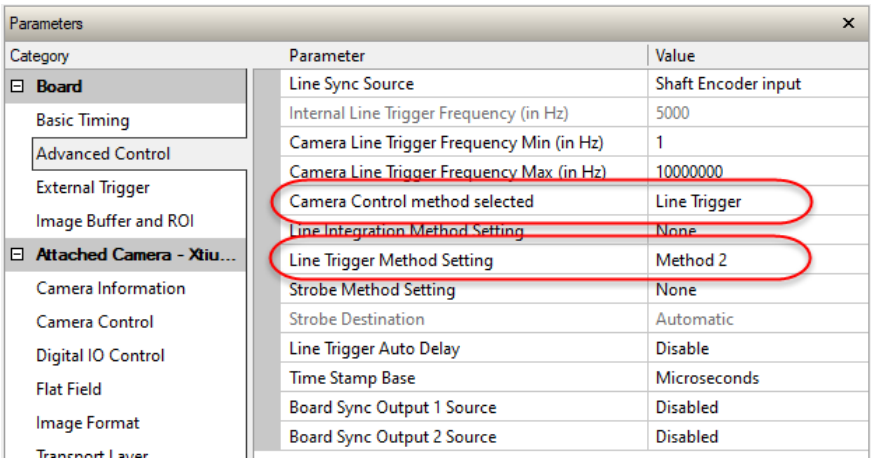
Line Scan Applications

Line Trigger Source Selection for Line Scan Applications

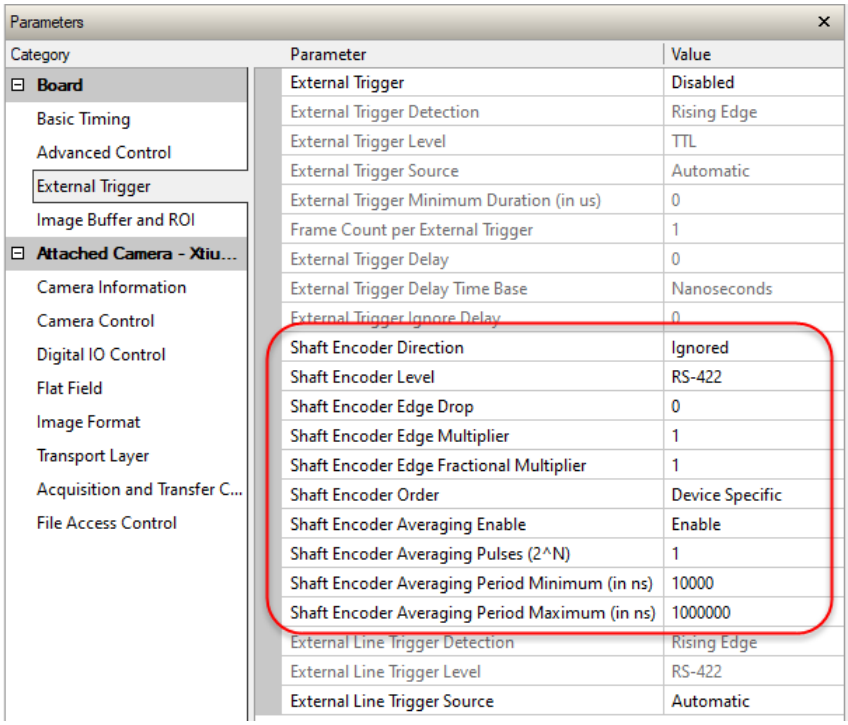
Line scan imaging applications require some form of external event trigger to synchronize line scan camera exposures to the moving object. This synchronization signal is either an external trigger source (one exposure per trigger event) or a rotary encoder source composed of a single or dual phase (also known as a quadrature) signal. Dual encoder signals are typically 90 degrees out of phase relative to each other and provide greater web motion resolution.

The External Line Trigger allows for exposure control using the pulse width.

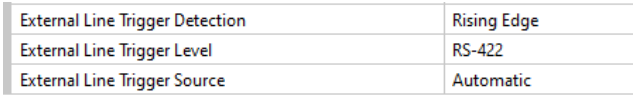
For any line sync source, also set the Camera Control method to Line Trigger and the Line Trigger Method to Method 2 (this also allows a trigger delay to be set).



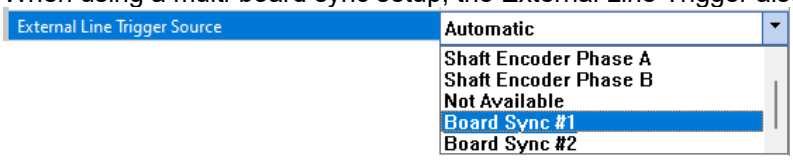
Encoder parameters are available in the External Trigger category:



For External Line Triggers, the following parameters are available:

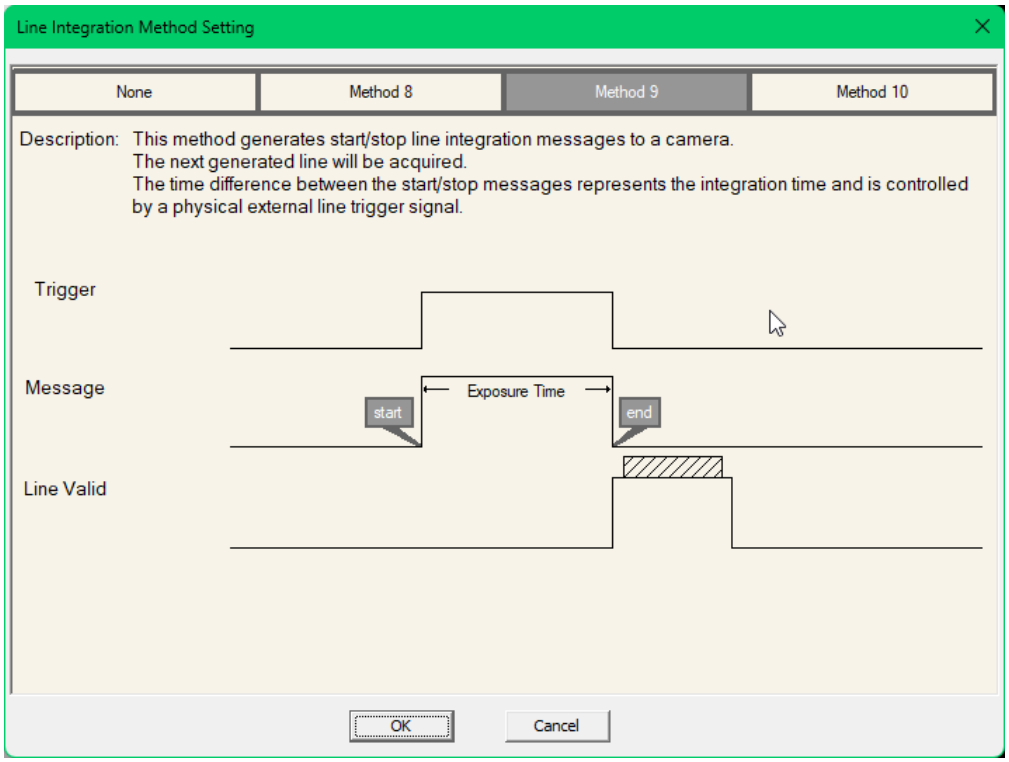


When using a multi-board sync setup, the External Line Trigger also allows you to use a Board Sync signal:



External line triggers can be used when the camera integration (exposure) time needs to be controlled by the pulse width (rotary encoders use a 50/50 duty cycle which does not allow for pulse width control).

In Sapera LT, use the Line Integration Method Setting, available in the Advanced Control category, to enable this method.



NOTE

TDI cameras typically do not support exposure control; the acquisition line rate controls integration time.

Related Sapera Parameters

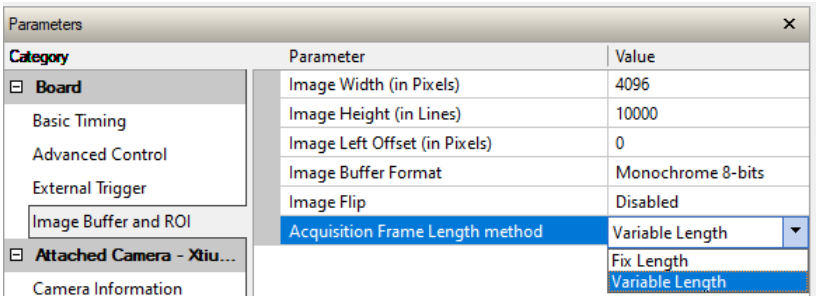
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE, CORACQ_PRM_LINE_TRIGGER_ENABLE, CORACQ_PRM_LINE_TRIGGER_METHOD, CORACQ_PRM_LINE_INTEGRATE_METHOD

Variable and Fixed Length Frames

When acquiring lines from a linescan camera, the frame grabber can use a fixed number of lines for each image frame or a variable number of lines.

The Image Height parameter sets the number of lines for fixed length; this setting also determines the maximum image height for variable length acquisition.

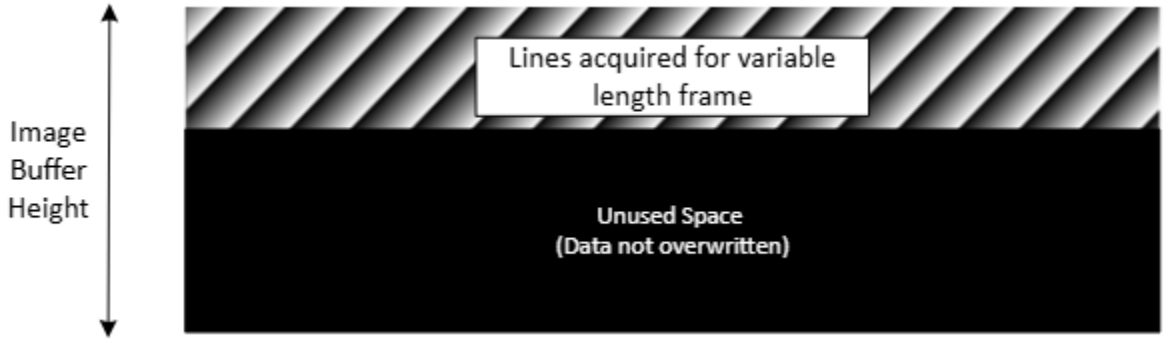
For example, in CamExpert, the Acquisition Frame Length method parameter is available in the Image Buffer and ROI category:



An external frame trigger can be connected to the frame grabber to signal when incoming lines should be acquired into an image frame; without an external frame trigger image frames are filled as lines are acquired.

For variable length frames, the External Trigger Detection parameter controls the number of lines sequentially grabbed up to the maximum of lines in the variable length frame buffer when set to Active High, Active Low or a Dual Input setting.

For a variable length frame, the image buffer will only write the number of lines determined by the external frame trigger; the remaining lines in the buffer are not overwritten (old data may be present).

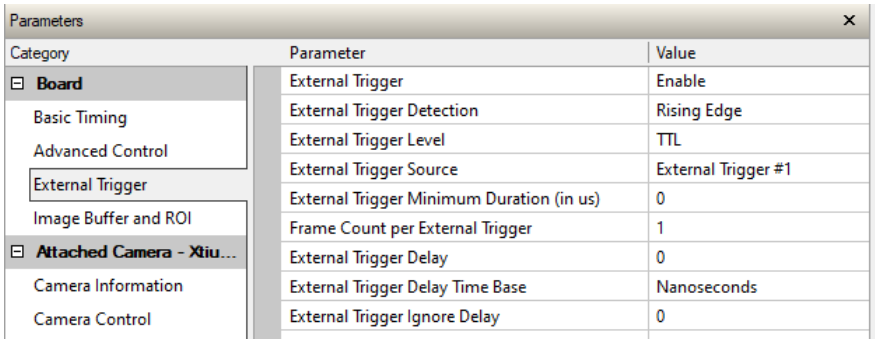


In Sapera LT, the SapBuffer::GetSpaceUsed function can be used to determine the actual size of the buffer and a child buffer created for the variable length frame (if the buffer is greater than 4GB the number of lines is returned).

For both fixed and variable length frames choosing an active low/high or dual input permits grabbing multiple consecutive images as long as the chosen signal is active. This action is also called “rolling over” to the next buffer. When choosing a single rising or falling edge, a single frame will be acquired; there is never any roll over.

External Frame Trigger Detection	Fixed Frame	Variable Frame
Active Low/High	Roll Over	Roll Over
Rising/Falling Edge	No Roll Over	No Roll Over
Dual Input Rising/Falling Edge	Roll Over	Roll Over

In Sapera LT, the external frame trigger parameters are available in the External Trigger category.



- Frame triggers can be differential (RS-422) or single ended (TTL, 12V, or 24V) industry standard, and be rising or falling edge active, active high or low, or double pulse rising or falling edge.
- Frame triggers: connect to the Xtium3-CLHS PX8 via the External Trigger Input 1 & 2 inputs (see J1 or J4).

Synchronization Signals for Fixed Frame Length Acquisition

A trigger event is only generated when a grab is active; when not grabbing no trigger events are generated. When a frame is complete, the frame grabber checks for the specified active trigger level and, if present, grabs the next frame; otherwise, it waits for the next detected active trigger level.

In the following diagrams:

- "*T*" indicates a valid external trigger event (in Sapera LT, SapAcquisition::EventExternalTrigger).
- "*Ignored*" is an ignored event (SapAcquisition::EventExternalTriggerIgnored).
- such that
  Ignored + *T* = total triggers received by frame grabber

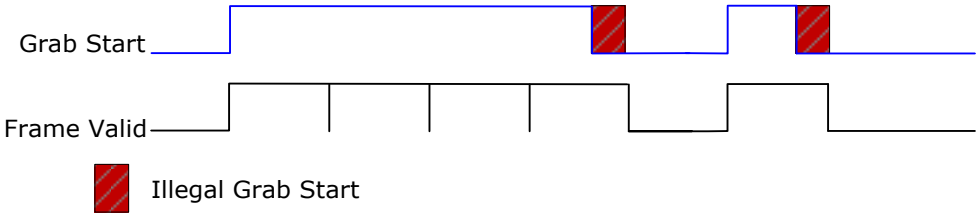


Figure 21: Line scan, Fixed Frame, No Trigger

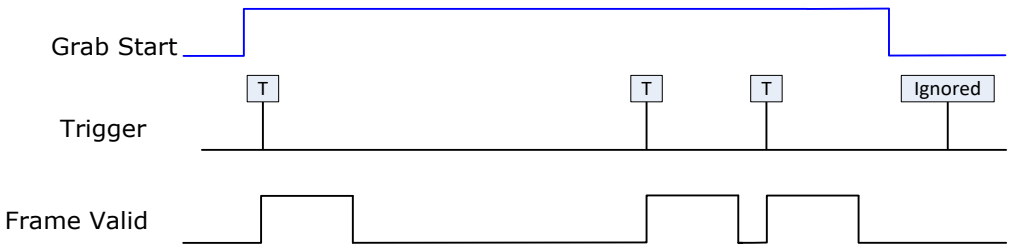


Figure 22: Line scan, Fixed Frame, Edge Trigger

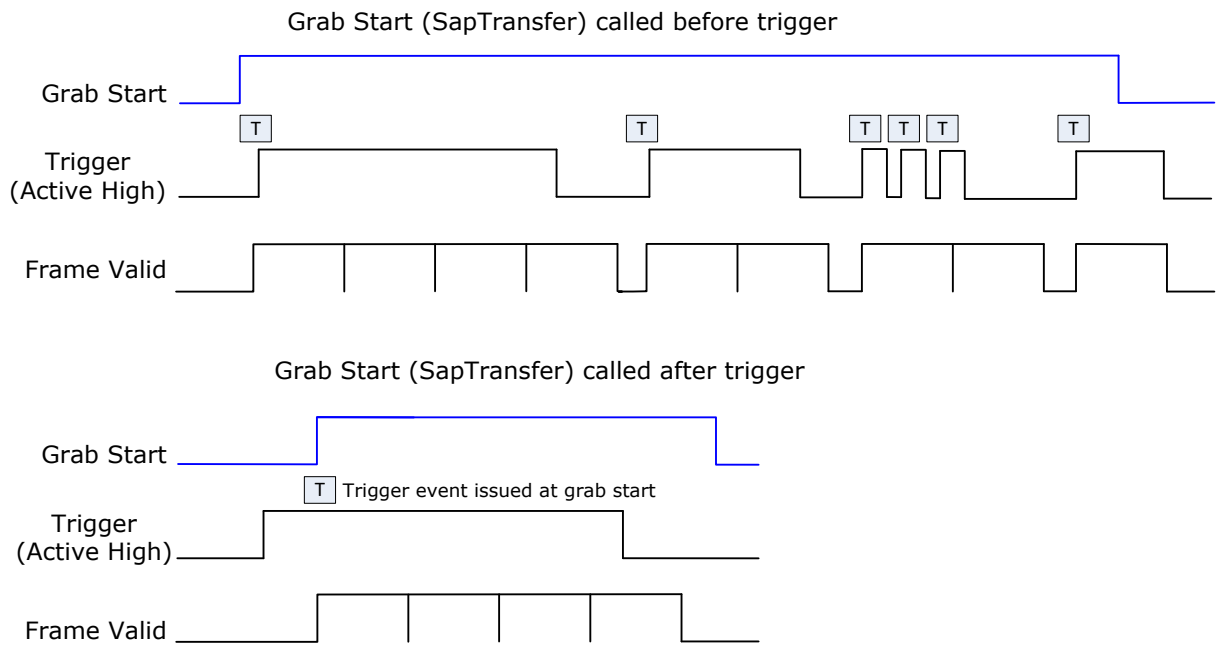


Figure 23: Line scan, Fixed Frame, Level Trigger (Roll-Over to Next Frame)

Synchronization Signals for Variable Frame Length Acquisition

For variable length frames, trigger ignored events (in Sapera LT, SapAcquisition::EventExternalTriggerIgnored) are not issued; a valid trigger event always initiates either a frame start or frame end.

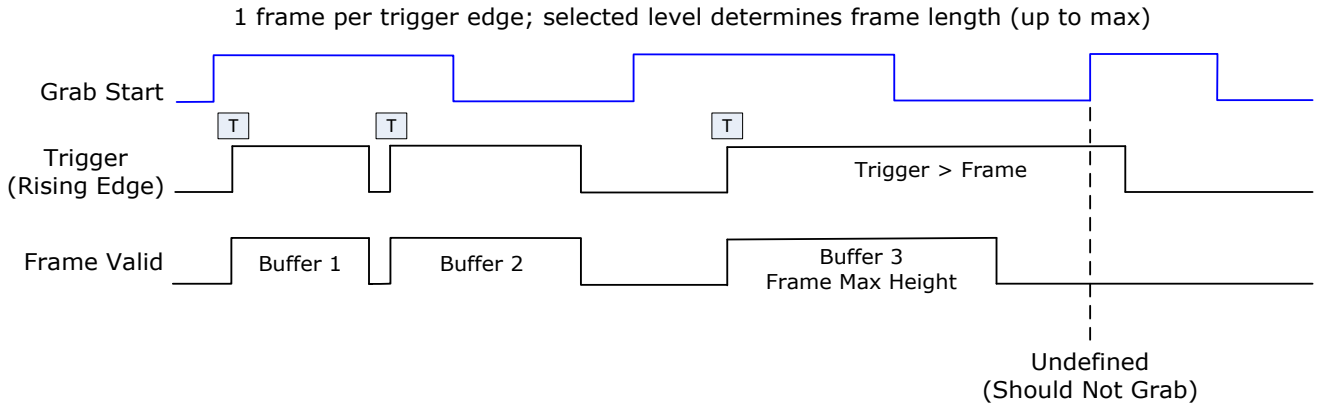


Figure 24: Line scan, Variable Frame, Edge Trigger (Active High determines Frame Length)

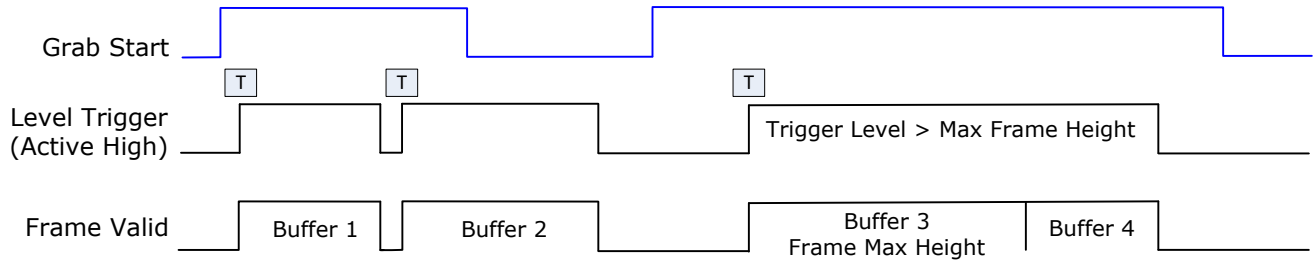


Figure 25: Line scan, Variable Frame, Level Trigger (Roll-Over)

Related Sapera Parameters

CORACQ_PRM_CROP_HEIGHT, CORACQ_PRM_FRAME_LENGTH, CORACQ_PRM_EXT_FRAME_TRIGGER_DETECTION

Encoder Interface Timing

The Xtium3-CLHS PX8 rotary (shaft) encoder inputs provide additional functionality with pulse drop, pulse multiply, and pulse direction support. There is also a fractional multiplier available. This allows the user to control the exact number of triggers generated for each encoder tick.

In CamExpert, these features are available in the External Trigger category.

Shaft Encoder Edge Drop	5
Shaft Encoder Edge Multiplier	1
Shaft Encoder Edge Fractional Multiplier	11.4

NOTE

The Shaft Encoder Edge Multiplier and Shaft Encoder Fractional Multiplier are mutually exclusive; only one mechanism can be used with the Shaft Encoder Edge Fractional Multiplier taking precedence when set to a value other than 1.

The encoder triggers the board every time a line needs to be acquired. The encoder consists of two inputs (A and B), offset by 90 degrees. Each transition corresponds to one trigger; the triggered camera acquires one scan line for each encoder pulse-edge.

NOTE

The distance between 2 ‘ticks’ must be >= 1/240Hz otherwise External Line Trigger Too Slow events may be generated. For example, with an encoder A or B use a minimum 120Hz signal or a 60Hz signal if using A and B.

When multiplying, do not exceed 125MHz.

Example using any Encoder Input with Pulse-drop Counter

The drop parameter can be used to skip transitions between valid acquisition triggers.

The figure below depicts a system where a valid camera trigger is any pulse edge from either encoder signal. For example, with a drop = 2, after a trigger the two following triggers are ignored.

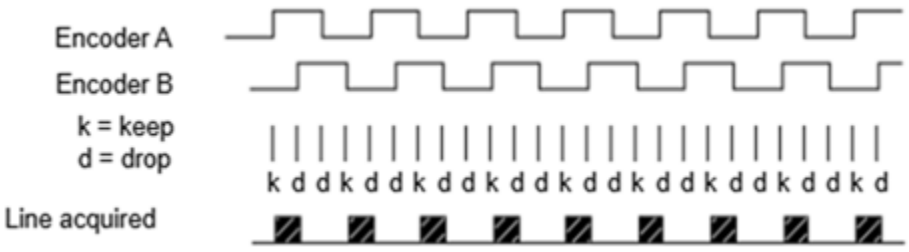


Figure 26: Encoder Input with Pulse-drop Counter

Encoder Averaging Engine

The rotary (shaft) encoder averaging engine allows the elimination of jitter from an encoder input signal. At high line rate frequency, small changes in camera line trigger rate can affect the integration time of the lines such that different intensity between captured lines can be seen. Jitter can also render the acquired image non linear as the time difference between the lines will vary. Removing the jitter from the encoder input eliminates these differences.

The encoder averaging engine can also be used to restore a non 50/50 duty cycle signal into a 50/50 one such that the acquisition of lines is linear (that is, time difference between consecutive acquired lines is equal). Engine has been tested with signals up to 80/20.

The pulse averaging engine works by calculating a rolling average of the period over 2**N pulses.

When enabled, the averaging engine uses the following controls:

- **Number of Pulses to average**. Range for N[1..8].
- **Minimum/Maximum period for valid pulses**. Maximum period is 2000 usec (minimum frequency of 500 Hz); minimum period is 0.05 usec (maximum frequency of 20 MHz)

In CamExpert, the averaging engine feaures are available , in the External Trigger category:

Shaft Encoder Averaging Enable	Enable
Shaft Encoder Averaging Pulses (2^N)	1
Shaft Encoder Averaging Period Minimum (in ns)	10000
Shaft Encoder Averaging Period Maximum (in ns)	1000000

NOTE

The averaging engine uses 100 MHz clock = 10 nsec resolution.

Theory of Operation

The encoder averaging engine allows the elimination of jitter from an encoder input signal. At high line rate frequency, small changes in camera line trigger rate can affect the integration time of the lines such that different intensity between captured lines can be seen. Jitter can also render the acquired image non linear as the time difference between the lines will vary. Removing the jitter from the encoder input eliminates these differences.

The encoder averaging engine can also be used to restore a non 50/50 duty cycle signal into a 50/50 one such that the acquisition of lines is linear (ie. time difference between consecutive acquired lines is equal).

Averaging is performed by calculating a rolling average of the period over 2**N pulses, where N has a range of 1...8 (2 to 256 pulses).

When the averaging engine is started, the rolling average is reset to 0 and only once 2**N consecutive valid pulses (period between each pulses are below user provided maximum period) generates a period that is between the minimum and maximum period provided will the engine start outputting averaged pulses. The 1st (2**N – 1) pulses will be passed to the next stage unmodified.

If the period between 2 consecutive pulses is above the maximum period:

- an External Line Trigger Too Slow event will be generated
- the rolling average count is reset to 0
- no more averaged pulses are output until a start condition is detected

If no pulses are received after a time that is greater than 2 times the current rolling average:

- an External Line Trigger Too Slow event will be generated
- the rolling average count is reset to 0
- no more averaged pulses are output until a start condition is detected

If rolling average goes below user provided minimum period:

- an External Line Trigger Too Fast event will be generated
- the rolling average is reset to 0
- no more averaged pulses are output until a start condition is detected.

NOTE

Maximum Period allowed is 2000 usec (Minimum Frequency of 500 Hz).

Minimum Period allowed is 0.05 usec (Maximum Frequency of 20 MHz)

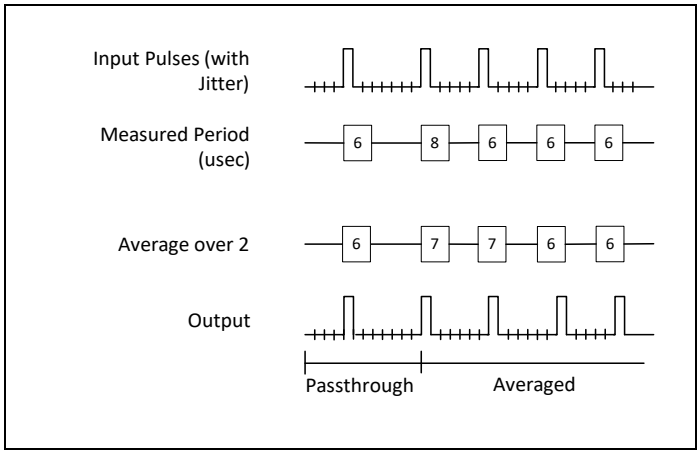
Exmples

The following examples use the following:

- N = 1 (that is, average of 2 pulses)
- Minimum Period = 6.0 usec (Max Frequency = 167 kHz
- Maximum Period = 20.0 usec (Min Frequency = 100 kHz

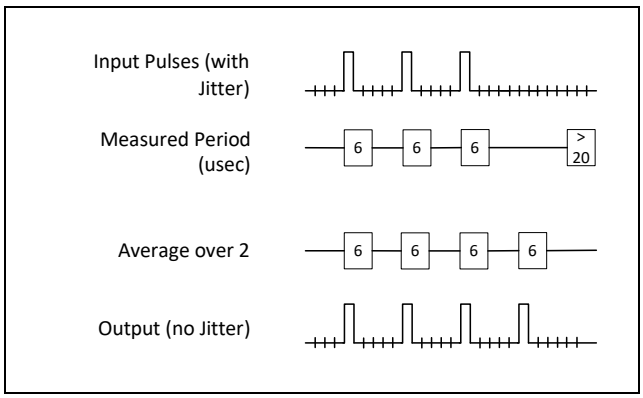
Start Condition

The start condition is detected once ‘N’ consecutive pulses generate a rolling average that is between the minimum and maximum period provided by user and that those ‘N’ consecutive pulses are all spaced with a period that is below the user provided maximum period.



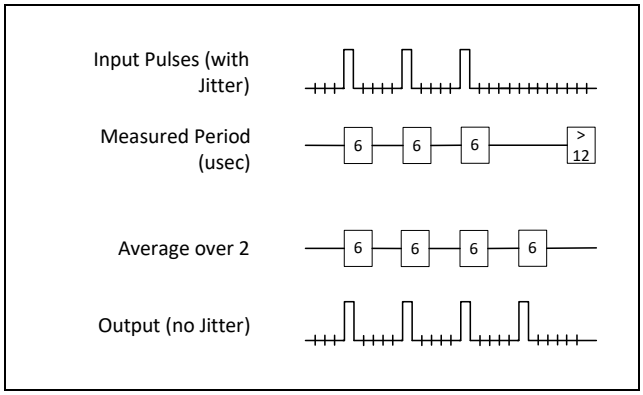
Stop Condition #1: Line Trigger Too Slow

This condition will occur if a period greater than the maximum period specified is detected. When this condition is detected a Line Trigger Too Slow event will be generated.



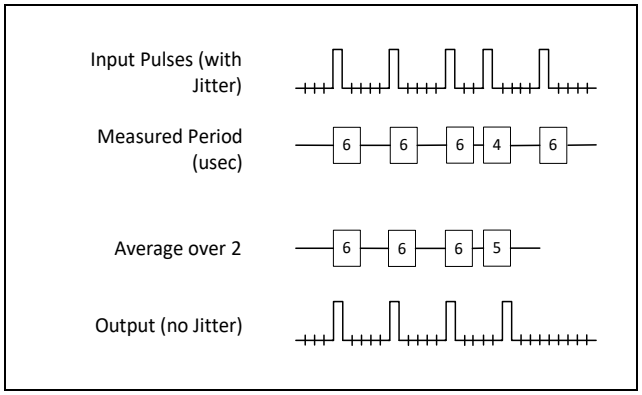
Stop Condition #2: Line Trigger Too Slow

This condition will occur if no pulses are received after a time that is more than 2 times the current rolling average. When this condition is detected a Line Trigger Too Slow event will be generated.



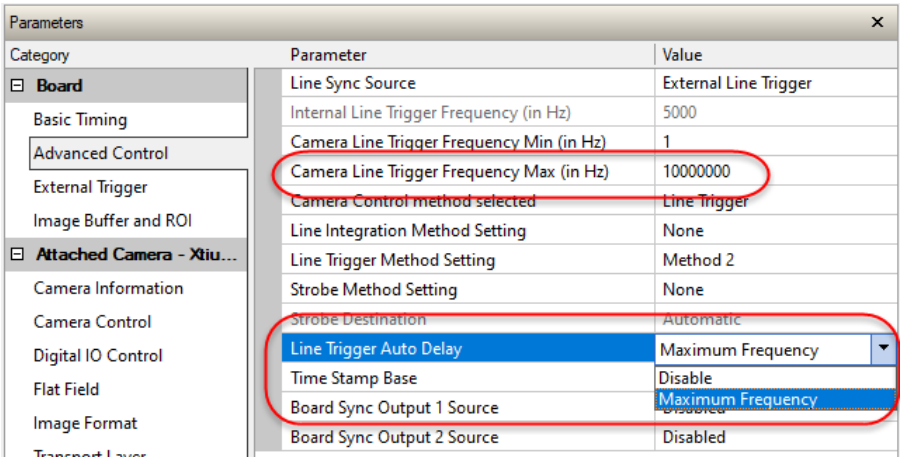
Stop Condition #3: Line Trigger Too Fast

This condition will occur if the **rolling average** goes below the minimum period specified. When this condition is detected a Line Trigger Too Fast event will be generated.



Rotary Encoder Jitter

When triggering at maximum line rates using a rotary (shaft) encoder with high jitter, the Line Trigger Auto Delay parameter, in conjunction with the Camera Line Trigger Max parameter, can be used to automatically delay line triggers to avoid over-triggering a camera, and thus not miss a line. The Line Trigger Too Fast event can be enabled when using the 'auto delay' feature.



Related Sapera Parameters

CORACQ_PRM_SHAFT_ENCODER_AVERAGING_ENABLE, CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PULSES, CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PERIOD_MIN, CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PERIOD_MAX, CORACQ_PRM_LINE_TRIGGER_AUTO_DELAY

Encoder Direction and Count

The rotary encoder engine, when fed with 2 phases A and B, can detect the direction forward or reverse of the encoder. Each encoder tick increments or decrements the encoder counter depending on the mode selected. Also, it is possible to select a specific direction (forward or reverse) such that the frame grabber will only send line triggers to the camera when the encoder direction detected matches the selected direction.

The example figure below shows encoder signals with high jitter. If the acquisition is triggered when phase B follows phase A, with jitter present phase B may precede phase A. Use of the Shaft Encoder Direction parameter will prevent false trigger conditions.

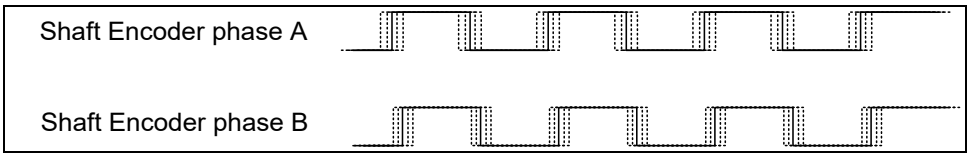


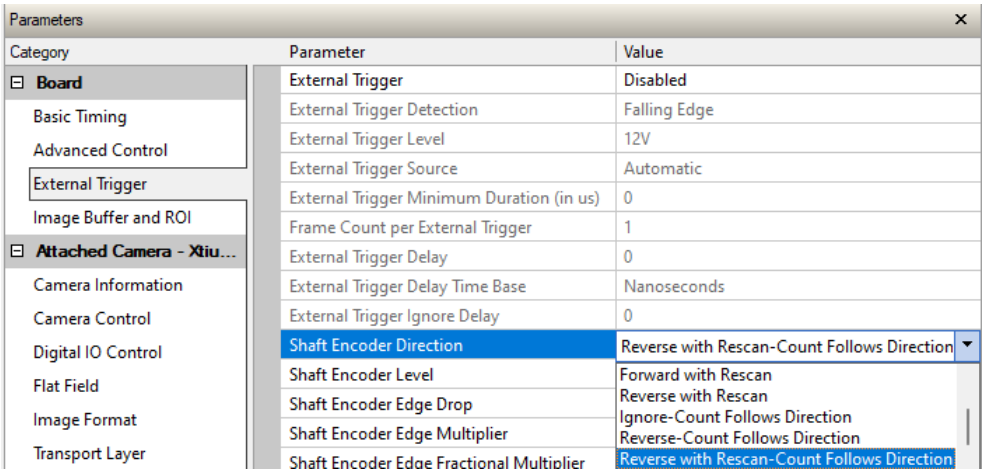
Figure 27: Using Shaft Encoder Direction Parameter

In the case the direction is opposite of the one selected, a reverse counter is used and will prevent re-scanning of lines located at the same encoder tick count by suppressing line triggers to the camera. For example, this can occur where system vibrations and/or conveyor backlash cause the encoder to briefly travel backwards.

In these cases, the acquisition device must count the reverse steps and subtract the forward steps such that only pulses after the reverse count reaches zero are considered. By using the event “Shaft Encoder Reverse Counter Overflow”, an application can monitor an overflow of this counter.

Alternatively, by enabling the rescan mode, the frame grabber will always send line triggers to the camera when the selected direction is detected.

In Cam Expert, the Shaft Encoder Direction is available in the External Trigger category.



Rotary (Shaft) Encoder Modes

Mode	Description
Ignored	Do not take into account the encoder direction. All encoder pulses are considered valid.
Forward	Increment the encoder counter when a forward motion is detected. A forward motion is detected when the order of the pulses are A/B. The encoder count will increment or decrement depending on the direction detected: forward = increment, reverse = decrement.
Reverse	Increment the encoder counter when a reverse motion is detected. A reverse motion is detected when the order of the pulses are B/A.
Forward with Rescan	The device will rescan when the encoder moves forward after reverse motion.
Reverse with Rescan	The device will rescan when the encoder moves in the reverse direction after forward motion.
Ignore Count Follows Direction	All encoder pulses are considered valid. The encoder count will increment or decrement depending on the direction detected: forward = increment, reverse = decrement.
Reverse Count Follows Direction	Reverse encoder pulses are considered valid. The encoder count will increment or decrement depending on the direction detected: forward = increment, reverse = decrement.
Reverse with Rescan -Count Follows Direction	Reverse encoder pulses are considered valid . The device will rescan when the encoder moves in the reverse direction after forward motion.The encoder count will increment or decrement depending on the direction detected: forward = increment, reverse = decrement.

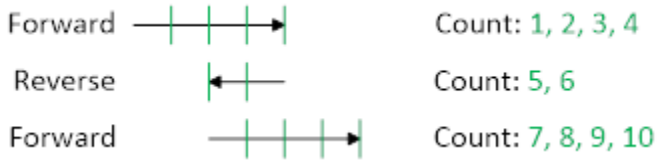
Mode	Use Reverse Count	Encoder Direction	Encoder Count	Trigger / Acquire Lines
Ignored	No	Forward	Increment	Yes
		Reverse	Increment	Yes
Ignore Count Follows Direction	No	Forward	Increment	Yes
		Reverse	Decrement	Yes
Forward	Yes	Forward	Increment	Yes
		Reverse	Decrement	No
Forward with Rescan	No	Forward	Increment	Yes
		Reverse	Decrement	No
Reverse	Yes	Forward	Decrement	No
		Reverse	Increment	Yes
Reverse Count Follows Direction	Yes	Forward	Increment	No
		Reverse	Decrement	Yes
Reverse with Rescan	No	Forward	Decrement	No
		Reverse	Increment	Yes
Reverse with Rescan -Count Follows Direction	No	Forward	Increment	No
		Reverse	Decrement	Yes

Encoder Modes Examples

Green: Trigger to Camera and Line Acquired.

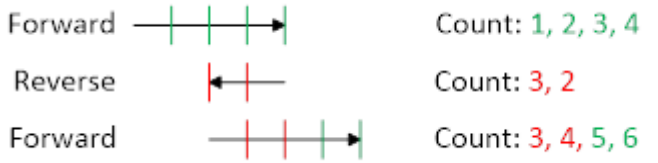
Red: No Trigger to Camera, no Line Acquired.

Ignored (Count Always Increment)

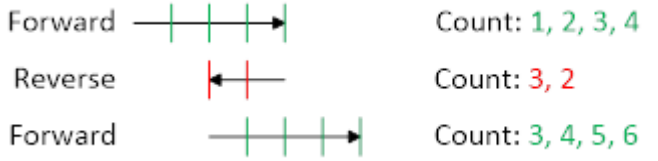


Ignore - Count Follows Direction

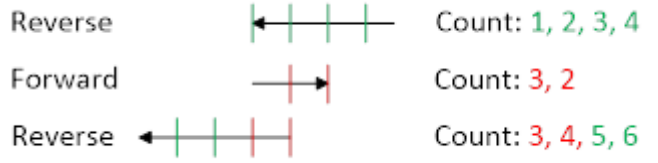
Forward (Count Increments)



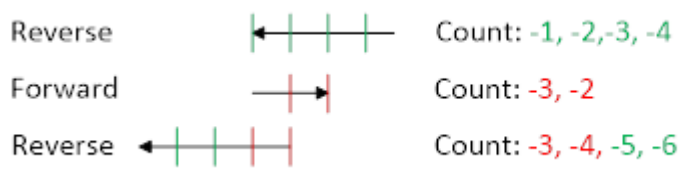
Forward with Rescan



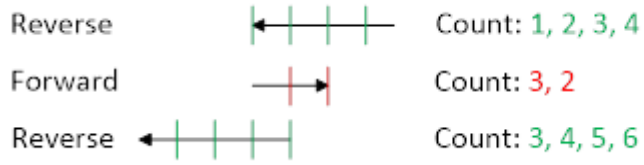
Reverse (Count Increments)



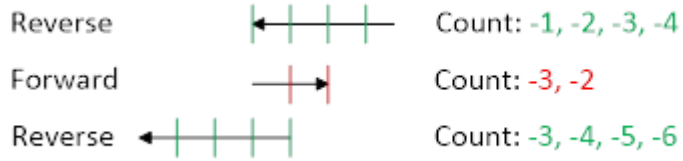
Reverse - Count Follows Direction



Reverse - Count Follows Direction



Reverse with Rescan - Count Follows Direction



Related Sapera Parameters

CORACQ_PRM_SHAFT_ENCODER_COUNT, CORACQ_PRM_SHAFT_ENCODER_DIRECTION

Xtium3-CLHS PX8 Sapera Parameters

The following sections detail the Xitum3-CLHS PX8 series Sapera LT parameters available through the Sapera LT CamExpert tool. For a complete list of supported parameters, see Appendix C: Sapera Parameter Reference.

CamExpert organizes frame grabber parameters into four categories, under the Board section:

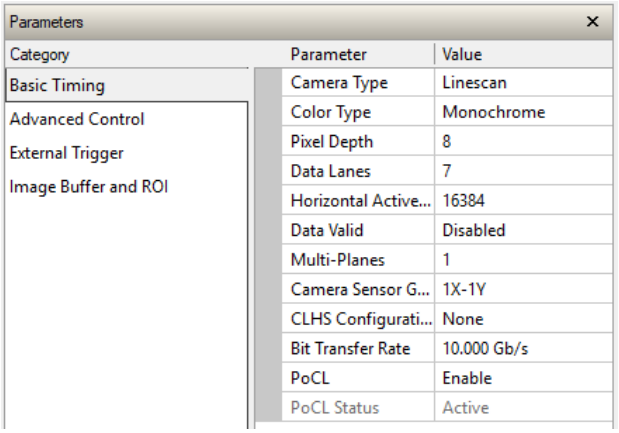
- Basic Timing
- Advanced Control
- External Trigger
- Image Buffer and ROI

CamExpert also provides access to connected cameras under the Attached Camera section.

For more information on using CamExpert, see the CamExpert Quick Start section.

Basic Timing Category

The Basic Timing category groups parameters such as camera type, the active image size, and other settings related to basic timing.



Parameter Descriptions

The following table describes the CamExpert Basic Timing category of Sapera LT parameters. Acquisition server notes, if applicable, indicate if parameter availability or supported values are dependent on the selected frame grabber acquisition server and acquisition device.

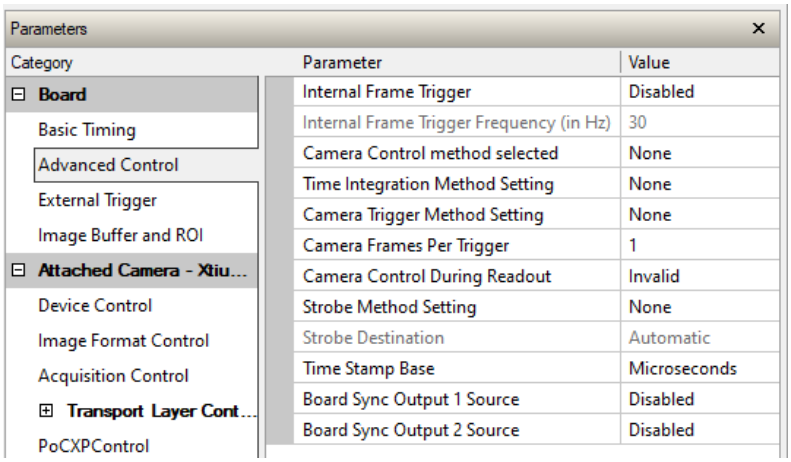
Display Name	Parameter	Description	Notes
Camera Type	CORACQ_PRM_SCAN	Source image type. Possible values are areascan or line scan.	
Color Type	CORACQ_PRM_VIDEO	Sets the color format of the input source.	Monochrome servers support: Monochrome Bayer mosaic
Pixel Depth	CORACQ_PRM_PIXEL_DEPTH	Pixel depth (bits per pixel) of the input source.	Monochrome servers support: 8, 10, or 12-bit
Data Lanes	CORACQ_PRM_DATA_LANES	Number of data lanes output by the camera.	
Horizontal Active (in Pixels)	CORACQ_PRM_HACTIVE	Sets the horizontal camera resolution in pixels. This corresponds to the visible part of the image from the camera. Valid range is: min = 32 pixel max = 65536 pixel step = 32 pixel Note: minimum is per lane	
Vertical Active (in Lines)	CORACQ_PRM_VACTIVE	Sets the vertical camera resolution in lines per frame. This corresponds to the visible part of the image from the camera. Valid range is 1-16777215.	Not shown for line scan cameras.

Data Valid	CORACQ_PRM_DATA_VALID_ENABLE	Specifies if the acquisition board uses the camera data valid signal. Boolean parameter (TRUE or FALSE).	
Multi-Planes	CORACQ_PRM_MULTI_PLANES	Specifies the number of planes output by the CLHS camera.	
Camera Sensor Geometry Setting	CORACQ_PRM_CHANNEL	Defines the number of channels output by the camera.	
CLHS Configuration	CORACQ_PRM_CLHS_CONFIGURATION	Sets the board's CLHS configuration. Possible values are: None Manual Acquisition Start/Stop	
PoCL	CORACQ_PRM_POCL_ENABLE	Enables/disables sending power to the Camera Link CLHS connector for Active Optical Cable (AOC). Boolean parameter (TRUE or FALSE).	This does not power the camera and must always be set to TRUE; setting to FALSE is the equivalent of disconnecting the AOC cable..
PoCL Status	CORACQ_PRM_SIGNAL_STATUS	Status of power signals connected to the acquisition device. Possible values are Active or Not Active.	

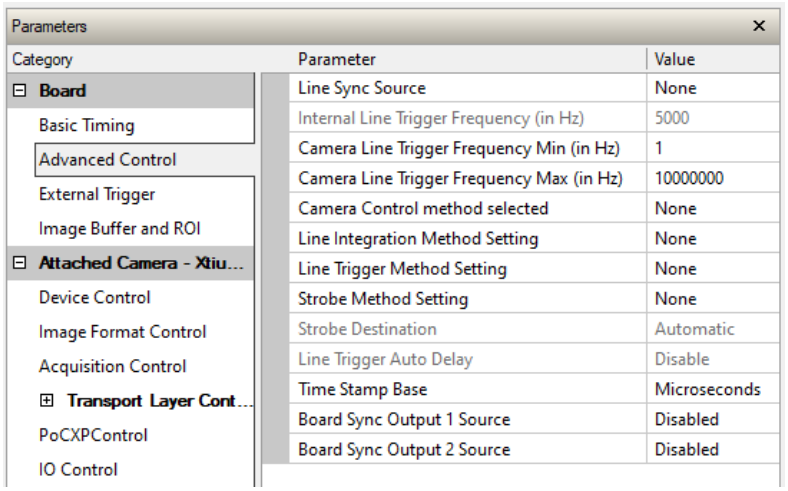
Advanced Control Category

The Advanced Control category groups parameters for configuring camera control signals, board sync outputs and other advanced settings.

Area Scan Parameters



Line Scan Parameters



Parameter Descriptions

The following table describes the CamExpert Advanced Control category of Sapera LT parameters. Acquisition server notes, if applicable, indicate if parameter availablility or supported values are dependent on the selected frame grabber acquisition server and acquisition device.

Display Name	Parameter	Description	Notes
Internal Frame Trigger	CORACQ_PRM_INT_FRAME_TRIGGER_ENABLE	Enables/disables the acquisition device's internal frame trigger. Boolean parameter (TRUE or FALSE).	Applies to areascan cameras only.
Internal Frame Trigger Frequency (in Hz)	CORACQ_PRM_INT_FRAME_TRIGGER_FREQ	Internal frame trigger frequency in Hz. Set to the required frame rate when using internal frame trigger to control camera acquisition. Valid range is 0.001-41000Hz.	Applies to areascan cameras only.
Line Sync Source	CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE CORACQ_PRM_INT_LINE_TRIGGER_ENABLE CORACQ_PRM_SHAFT_ENCODER_ENABLE	Selects the line trigger source for line scan cameras, unless free-running.	Applies to line scan cameras only.
Interal Line Trigger Frequency (in Hz)	CORACQ_PRM_INT_LINE_TRIGGER_FREQ	Sets the internal line trigger frequency, in Hz. Applies only when the Line Sync Source is set to Internal Line Trigger.	Applies to line scan cameras only.
Camera Line Trigger Frequency Min (in Hz)	CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MIN	Sets the camera's minimum line trigger frequency. Minimum value is 1Hz.	Applies to line scan cameras only.
Camera Line Trigger Frequency Max (in Hz)	CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MAX	Sets the camera's maximum line trigger frequency. Maximum value is 10000000 Hz.	Applies to line scan cameras only.
Camera Control method selected	CORACQ_PRM_TIME_INTEGRATE_ENABLE CORACQ_PRM_CAM_TRIGGER_ENABLE CORACQ_PRM_LINE_TRIGGER_ENABLE	Enables or disables an available camera control method. Each supported control method has one or more operating modes to choose from; refer to the parameters: Camera Trigger Method Setting Time Integration Method Setting.	
Time Integration Method Setting	CORACQ_PRM_TIME_INTEGRATE_METHOD CORACQ_PRM_TIME_INTEGRATE_DELAY	When the Camera Control method is Time Integration, select and configure the control method required. Click on the parameter field to open the configuration dialog.	
Camera Trigger Method Setting	CORACQ_PRM_CAM_TRIGGER_METHOD	When an asynchronous trigger pulse to a camera is required, select and configure the required method.	
Line Integration Method Setting	CORACQ_PRM_LINE_INTEGRATE_METHOD	Sets the method for controlling the camera's line integration.	Applies to line scan cameras only
Line Trigger Method Setting	CORACQ_PRM_LINE_TRIGGER_METHOD	Sets the method for line trigger pulse output.	Applies to line scan cameras only
Camera Frames Per Trigger	CORACQ_PRM_CAM_FRAMES_PER_TRIGGER	Specifies the number of frames output by the camera per trigger; currently not available.	Applies to area scan cameras only.
Camera Control During Readout	CORACQ_PRM_CAM_CONTROL_DURING_READOUT	Specifies if the camera control signals can be sent during the readout of a frame. Possible values are: • Valid • Invalid • Ignore	
Strobe Method Setting	CORACQ_PRM_STROBE_METHOD CORACQ_PRM_STROBE_ENABLE CORACQ_PRM_STROBE_DELAY CORACQ_PRM_STROBE_DURATION CORACQ_PRM_STROBE_LEVEL CORACQ_PRM_STROBE_POLARITY	When a strobe output signal from the acquisition board is required, select and configure the control method required. Note, method 1 is only available for areascan camera type; method 3 for line scan only.	
Line Trigger Auto Delay	CORACQ_PRM_LINE_TRIGGER_AUTO_DELAY	Enables delaying line triggers to a camera based on the selected method. Used to avoid over-triggering a camera.	Applies to line scan cameras only
Time Stamp Base	CORACQ_PRM_TIME_STAMP_BASE	Sets the counter stamp time base. Possible values are: • Microseconds • Line Counts • External line trigger or shaft encoder • Shaft Encoder • 100 Nanoseconds	

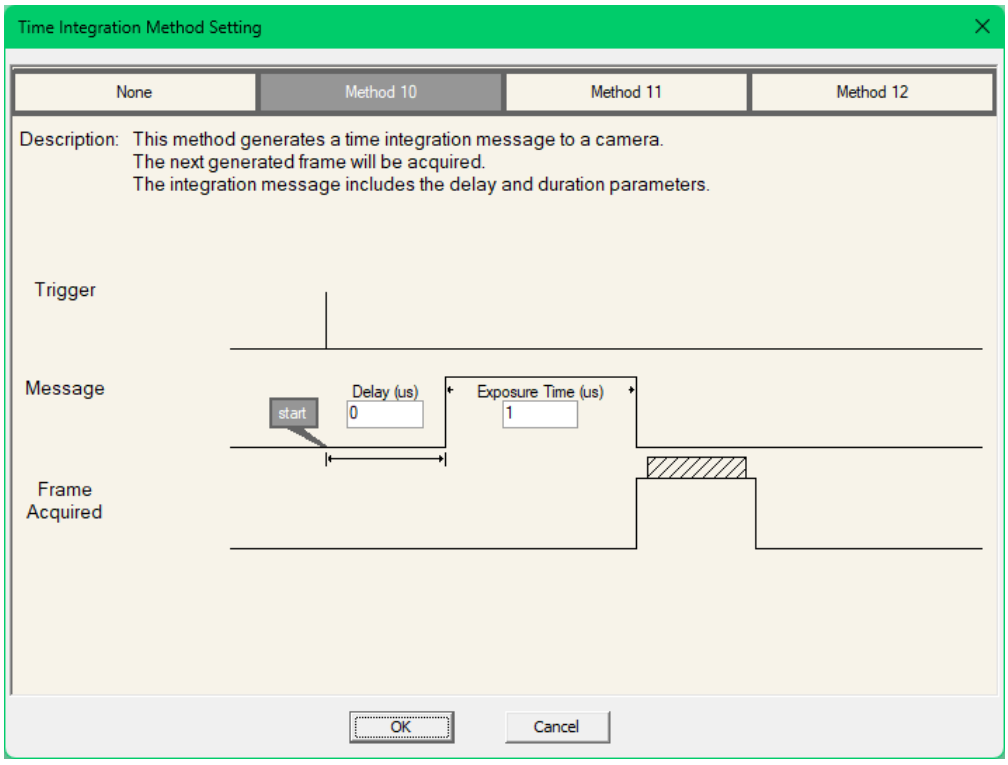
Board Sync Output 1 Source	CORACQ_PRM_BOARD_SYNC_OUTPUT1_SOURCE	Specifies the signal to output on board sync output 1. This parameter permits the synchronization of two acquisition devices using a signal from one acquisition device and synching the second acquisition device with it.	
Board Sync Output 2 Source	CORACQ_PRM_BOARD_SYNC_OUTPUT2_SOURCE	Specifies the signal to output on board sync output 2. This parameter permits the synchronization of two acquisition devices using a signal from one acquisition device and synching the second acquisition device with it.	

Sapera 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 3 supported)
- Line Trigger Methods (method 2)
- Line Integration Methods (methods 8, 9 and 10 supported)
- Time Integration Methods (method 10, 11 and 12)
- Strobe Methods (method 1, 3, 4 and 5 supported)

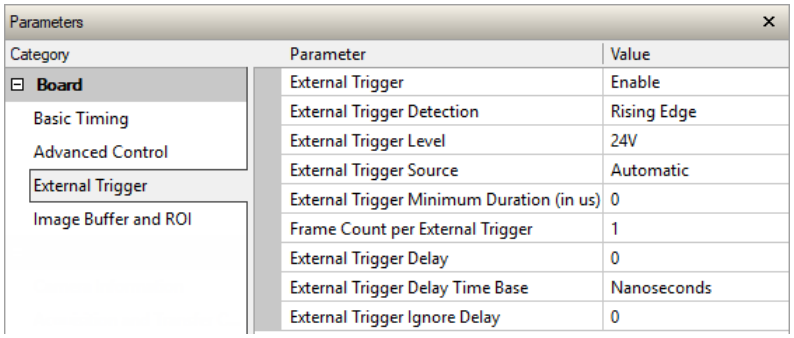
For each method the CamExpert dialog provides detailed graphics which illustrate how the method works, included parameter fields for setting timing values. For example, the Time Integration Method Setting dialog provides interactive settings for all available methods.



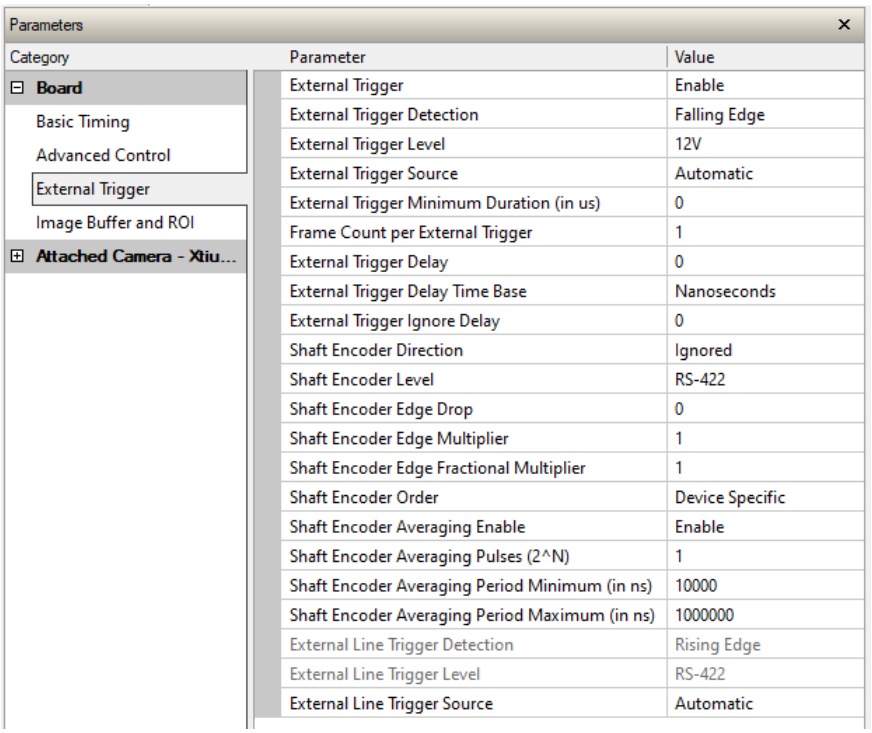
External Trigger Category

The External category groups parameters for configuring an external trigger for controlling image acquisition.

Area Scan Parameters



Line Scan Parameters



Parameter Descriptions

The following table describes the CamExpert External Trigger category of Sapera LT parameters.

Display Name	Parameter	Description
External Trigger	CORACQ_PRM_EXT_TRIGGER_ENABLE (area scan) CORACQ_PRM_EXT_FRAME_TRIGGER_ENABLE (line scan)	Enables/disables external trigger on the acquisition board. When enabled, the acquisition board acquires an image frame from the camera after receiving the trigger. Boolean parameter (TRUE or FALSE). Note: Applies to area scan cameras only.
External Trigger Detection	CORACQ_PRM_EXT_TRIGGER_DETECTION (area scan) CORACQ_PRM_EXT_FRAME_TRIGGER_DETECTION (line scan)	Defines the signal detected that generates an external trigger event to the acquisition device. Two types of trigger are available: *Level Trigger: Active Low / High* Logic level (Low/High) on the trigger input enables continuous image capture until the trigger input is set to opposite logic . *Edge Trigger: Rising / Falling edge* Edge transition of a trigger pulse captures one image frame. Line scan cameras also support: *Dual-Input Trigger Rising Edge / Dual-Input Trigger Falling Edge*
External Trigger Level	CORACQ_PRM_EXT_TRIGGER_LEVEL (area scan) CORACQ_PRM_EXT_FRAME_TRIGGER_LEVEL (line scan)	Specifies the electrical level of the external trigger connected to the acquisition board. Possible values: TTL single-ended logic signal RS-422 balanced logic signal 12V single-ended logic signal 24V single-ended logic signal
External Trigger Source	CORACQ_PRM_EXT_TRIGGER_SOURCE (area scan) CORACQ_PRM_EXT_FRAME_TRIGGER_SOURCE (line scan)	Specifies the physical input source the external frame trigger is connected to or which trigger input is used on the acquisition device. Note: to assign the external trigger source to a GPIO it must be reserved; By default, boards are shipped with User Interface General Inputs 1 & 2 reserved for External Triggers and User Interface General Outputs 1 & 2 reserved for Strobe Outputs. Refer to User Interface GIOs Reservation setting for more information on using the Teledyne DALSA Device Manager tool to reserve GPIOs.
External Trigger Minumum Duration (in µs)	CORACQ_PRM_EXT_TRIGGER_DURATION	Minimum external trigger pulse duration (in µs), needed for the pulse to be acknowledged by the acquisition device. If the duration of the pulse is shorter, the pulse is ignored. This feature is useful for trigger pulse debouncing. If the value is '0', no validation is performed
Frame Count per External Trigger	CORACQ_PRM_EXT_TRIGGER_FRAME_COUNT	Number of images to acquire upon receiving an external trigger. Valid range is 1-262142. Note, infinite frame count (-1) is not supported.
External Trigger Delay	CORACQ_PRM_EXT_TRIGGER_DELAY	Sets the delay between the reception of the trigger signal and the start of the image acquisition. Units are specified by the External Trigger Delay Time Base parameter.
External Trigger Delay Time Base	CORACQ_PRM_EXT_TRIGGER_DELAY_TIME_BASE	Sets the external trigger delay time base. Possible values: • Line Counts • Nanoseconds • External Line Trigger or Shaft Encoder • Shaft Encoder
External Trigger Ignore Delay	CORACQ_PRM_EXT_TRIGGER_IGNORE_DELAY	Sets the time delay, in µsec, where if another external trigger occurs, it is ignored. Valid range is 0-42949672. The start of the delay (time '0') is the end of the next vertical sync for analog cameras, or the beginning of the next frame valid for digital cameras, following a valid external trigger.

Shaft Encoder Direction	CORACQ_PRM_SHAFT_ENCODER_DIRECTION	Selects the direction of the encoder that increments/decrements the acquisition device encoder counter. Support of dual phase encoders might require that the direction of motion be considered. This is the case where system vibrations and/or conveyor backlash can cause the encoder to momentarily travel backwards. The acquisition device must in those cases count the reverse steps and subtract the forward steps such that only pulses after the reverse count reaches zero are considered valid.
Shaft Encoder Edge Drop	CORACQ_PRM_SHAFT_ENCODER_DROP	Number of encoder signal edges dropped between active encoder triggers. Use the pulse drop feature to reduce the acquisition rate without reducing the encoder trigger rate.
Shaft Encoder Edge Multiplier	CORACQ_PRM_SHAFT_ENCODER_MULTIPLY	Number of signal edges generated internally on the acquisition board for each external encoder signal edge. Use when image acquisitions are controlled by an external encoder trigger but multiple acquisitions are needed from each trigger.
Shaft Encoder Edge Fractional Multiplier	CORACQ_PRM_SHAFT_ENCODER_MULTIPLY	Number of signal edges generated internally on the acquisition board for each external encoder signal edge. Use when image acquisitions are controlled by an external encoder trigger but multiple acquisitions are needed from each trigger.
Shaft Encoder Order	CORACQ_PRM_SHAFT_ENCODER_ORDER	Specifies the order of the drop/multiply operation of the encoder.
Shaft Encoder Averaging Enable	CORACQ_PRM_SHAFT_ENCODER_AVERAGING_ENABLE	Enables/Disables averaging of the encoder signal edges received.
Shaft Encoder Averaging Pulses (2^N)	CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PULSES	Specifies the number of encoder signal edges used to make an average.
Shaft Encoder Averaging Period Minimum (in ns)	CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PERIOD_MIN	Minimum time between 2 encoder signal edges for the pulses to be averaged. If minimum time is not respected, the average engine will do a clean restart.
Shaft Encoder Averaging Period Maximum (in ns)	CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PERIOD_MAX	Maximum time between 2 encoder signal edges for the pulses to be averaged. If the maximum time is not respected, the average engine will do a clean restart.
External Line Trigger Detection	CORACQ_PRM_EXT_LINE_TRIGGER_DETECTION	Select the signal detected (rising edge/falling edge) that generates an external line trigger to the acquisition device.
External Line Trigger Source	CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE	Specifies the physical input source the external line trigger is connected to on the acquisition device, in the case where the acquisition device has more than one input. Line scan cameras typically use the encoder signals as the acquisition board line trigger. The ‘Automatic’ choice selects the trigger normally used with the acquisition module, in the case of multiple modules – multiple trigger inputs.

External Trigger Reservation

By default, General Input 1 and 2 are reserved as External Trigger Inputs; as such, these external trigger inputs generate individual interrupts and are read by the Sapera application. To use these inputs for purposes other than external trigger, the “reserved” status can be changed using the Xtium3 Device Manager; see the Information Field Description for the User Interface GIOs Reservation setting.

When external triggers are not reserved, the External Trigger Source can only be set to Software Trigger or Data Valid.

For example, in CamExpert:

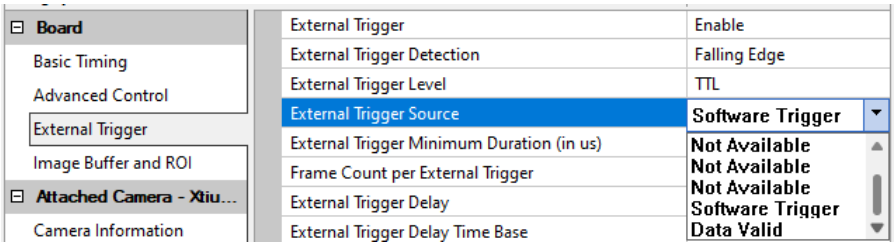
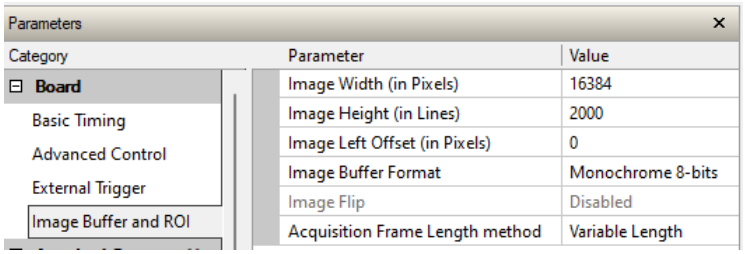


Image Buffer and ROI Category

The Image Buffer and ROI category groups parameters for the configuring the image buffer format, size and offset settings, as well as image flipping.



Parameter Descriptions

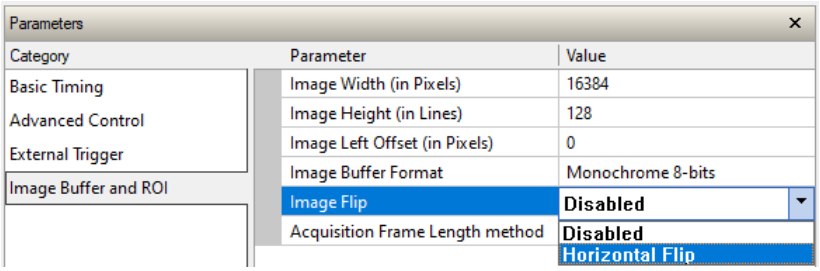
The following table describes the CamExpert Image Buffer and ROI category of Sapera LT parameters. Acquisition server notes, if applicable, indicate if parameter availability or supported values are dependent on the selected frame grabber acquisition server and acquisition device.

Display Name	Parameter	Description	Notes
Image Width (in Pixels)	CORACQ_PRM_CROP_WIDTH	Cropped width of the acquisition image, in pixels; this parameter defines the width of the image transferred to the frame buffer. The maximum width is the active horizontal of the image source (see the Horizontal Active parameter in the Basic Timing category). Cropping increments depend on the selected acquisition server; CamExpert automatically adjusts numerical entries to valid increments.	Note: image data is not scaled.
Image Height (in Lines)	CORACQ_PRM_CROP_HEIGHT	Cropped height of the acquisition image, in lines; this parameter defines the vertical dimension of the image transferred to the frame buffer. The maximum height is the active vertical width of the image source (see the Vertical Active parameter in the Basic Timing category). Cropping increments depend on the selected acquisition server; CamExpert automatically adjusts numerical entries to valid increments.	Note: image data is not scaled.
Image Left Offset (in Pixels)	CORACQ_PRM_CROP_LEFT	Number of pixels to crop from the left side of the acquisition image before transfer to the frame buffer. The maximum left offset is the active horizontal width of the image source less one increment step. Cropping increments depend on the selected acquisition server; CamExpert automatically adjusts numerical entries to valid increments.	Note: image data is not scaled.
Image Top Offset (in Lines)	CORACQ_PRM_CROP_TOP	Number of lines to crop from the top of the acquisition image before transfer to the frame buffer. The maximum top offset is the active vertical height of the image source less one increment step. Cropping increments are acquisition hardware dependent; CamExpert automatically adjusts numerical entries to valid increments.	Note: image data is not scaled.
Image Buffer Format	CORACQ_PRM_OUTPUT_FORMAT	Data format for the acquisition image transfer to the frame buffer.	The data buffer format is dependent on the selected acquisition server; for details refer to the CORACQ_PRM_OUTPUT_FORMAT parameter description

Image Flip	CORACQ_PRM_FLIP	Enables real-time on-board horizontal image flip function. The Xtium3-CLHS PX8 also supports a vertical flip operation using CORXFER_PRM_FLIP.	
Acquisition Frame Length method	CORACQ_PRM_FRAME_LENGTH	Specifies if the images output by the acquisition device have a fixed or variable frame length. Variable frame length is of interest with line scan applications where the external frame trigger timing changes. External frame trigger could be active high or low pulse width controlled, or external frame trigger could be from a double pulse control.	Applies to line scan cameras only.

Horizontal and Vertical Flip

The horizontal flip is enabled using the CORACQ_PRM_FLIP parameter and is performed during the acquisition. This parameter is available in CamExpert in the board's Image Buffer and ROI category:



The vertical flip is enabled using the CORXFER_PRM_FLIP parameter and is done when performing the DMA transfer.

Using the Xtium3-CLHS PX8 with Sapera

Sapera Servers and Resources

A Sapera Server is an abstract representation of a physical device like a frame-grabber or camera. When using the SapAcquisition or SapAcqDevice constructors, the location parameter specifies the server to use to create the object. Use the Sapera Configuration utility to find the names and indices of all Sapera servers in your system.

In Sapera LT all frame grabbers are configured using the SapAcquisition class. All CLHS cameras are GenCP compliant and are configured in Sapera LT using the SapAcqDevice class.

The GIO Module allows for IO events, such as rising or falling edges, to execute callback functions, as well as to get/set the pin state and are configured using the SapGio class. Additional acquisition events are available for inputs used as frame or line triggers, as well as the rotary (shaft) encoder; see Supported Events and Transfer Methods.

NOTE

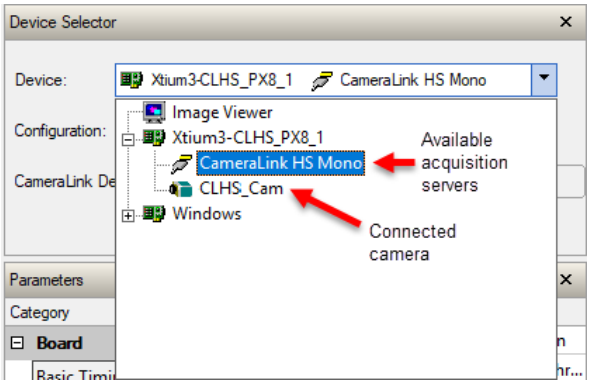
Currently, CLHS cameras do not have their own server, therefore it is available under the Xtium3_CLHS server. For example, in CamExpert the Xtium3 server displays both the frame grabber and camera resources.

Servers	**Resources**			
Name	**Type**	**Name**	**Index**	**Description**
Xtium3-CLHS_PX8_1 (Standard)	Acquisition Module	Camera Link HS Mono	0	CLHS Mono Camera
	Acquisition Device	*<Device Name>* * Name of Camera	0	CLHS Camera
All Configurations	GIO Module	General Outputs #1	0	4 General Outputs LVTTL
		General Inputs #1	1	6 General Inputs
		Bidirectional General I/Os #1	2	8 Bidirectional Board-to-Board I/Os
		General Outputs Open Collector #2	3	8 Open Collector Outputs

CamExpert Quick Start

The Sapera CamExpert tool is the interfacing tool for Xtium3-CLHS PX8 frame grabbers and connected cameras; it is supported by the Sapera library and hardware. CamExpert allows a user to test frame grabber and camera functions. CamExpert is the primary tool to configure, test and calibrate your camera and imaging setup. Display tools include, image pixel value readout, image zoom, and histogram.

When an acquisition server is selected, CamExpert only presents parameters supported by the selected device.



The Xtium3-CLHS PX8 firmware supports one Camera Link HS camera and provides the following acquisition servers:

Firmware	Acquisition Servers
Camera Link HS (default configuration)	Xtium3-CLHS PX8 1 CameraLink HS Mono

After CamExpert identifies the camera (as per the Camera Link device discovery protocol), timing parameters are displayed and the user can test image acquisition by clicking on Grab. Depending on the selected server, different parameters may be displayed.

For more information, see the Sapera Servers and Resources section.

CamExpert Interface

An important component of CamExpert is its live acquisition display window which allows immediate verification of timing or control parameters without the need to run a separate acquisition program.

Functional tools include hardware Flat Field calibration and operation support (see Flat Field Correction: Theory of Operation), plus support for either hardware based or software Bayer filter camera decoding with auto white balance calibration.

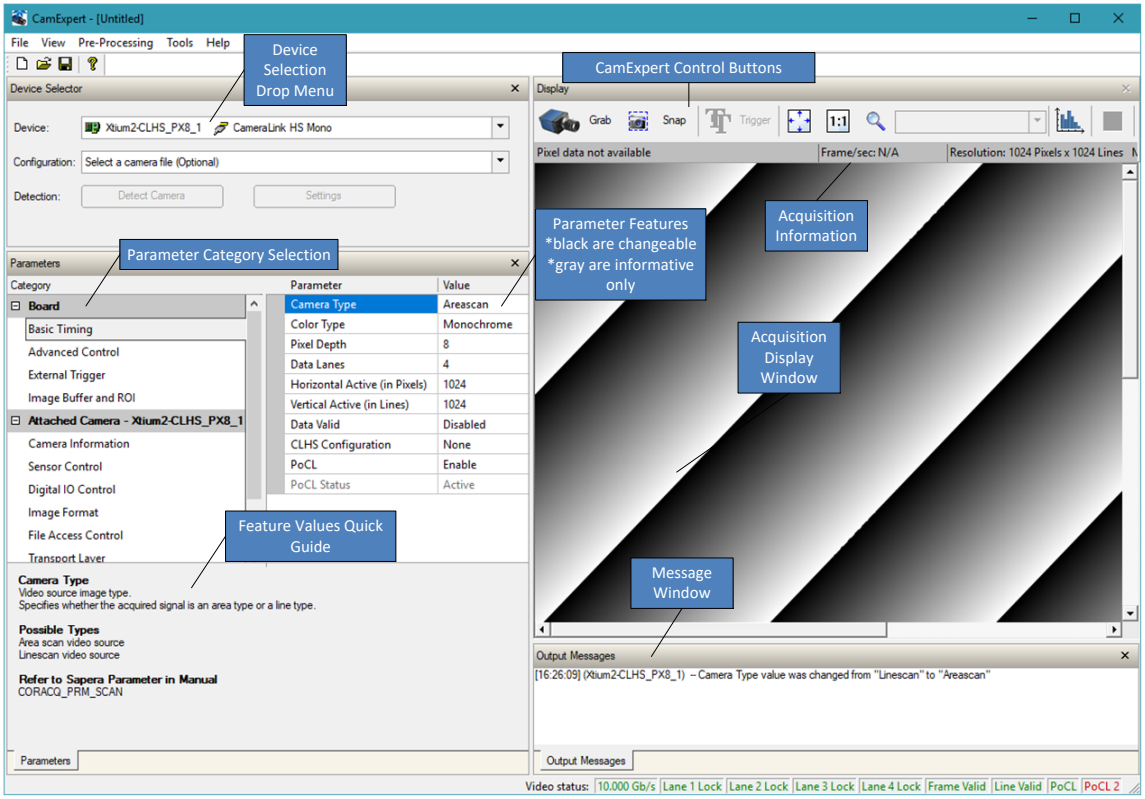
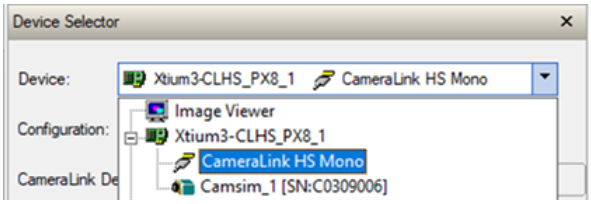


Figure 28: CamExpert Program

CamExpert groups camera features into functional categories. The features shown depend on the frame grabber used and what camera is connected. The values are either the camera defaults or the last stored value when the camera was used. The general descriptions below are not specific to a particular camera.

- **Device Selector:** Two drop menus allow selection of which device and which saved configuration to use.
  - **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.



  - **Configuration:** Select the timing for a specific camera model included with the Sapera installation or a standard image standard. The *User's* subsection is where user created camera files are stored.
- **Parameter Groups:** Select a function category and change parameter values as required. Descriptions for the camera parameters change dependent on the camera.
  - **Basic Timing**: Provides or change static camera parameters.
  - **Advanced Controls**: Advanced parameters used to select various integration methods, frame trigger type, Camera Link HS controls, etc.
  - **External Trigger**: Parameters to configure the external trigger characteristics.
  - **Image Buffer and ROI**: Allows control of the host buffer dimension and format.
- **Display:** An important component of CamExpert is its live acquisition display window, which allows immediate verification of timing or control parameters without the need to run a separate acquisition program. **Grab** starts continuous acquisition (button then toggles to **Freeze** to stop). **Snap** is a single frame grab. **Trigger** is a software trigger to emulate an external source.
- **Output Messages and Video Status Bar:** Events and errors are logged for review. Camera connection status is displayed where green indicates connected signals present.

The CamExpert tool is described more fully in the Sapera Getting Started and Sapera Introduction manuals.

Sapera Camera Configuration Files

CamExpert generates the Sapera camera configuration file (yourcamera.ccf) based on timing and control parameters entered. When using the Sapera LT API in your imaging application, the frame grabber parameter settings can be loaded from this file. 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 with a dialog window to select a camera configuration file (for details on the included demos, see the Sapera Demo Applications section). Even when using the Xtium3-CLHS PX8 with common image transfer protocols, a camera file is required. Therefore, CamExpert is typically the first Sapera application run after an installation. Existing .ccf files can be copied to any new board installations when similar cameras are used.

Camera Types & Files

The Xtium3-CLHS PX8 supports digital area scan or line scan cameras using the Camera Link HS interface standard. Browse our web site teledynevisionsolutions.com for the latest information on Teledyne CLHS cameras.

Using the Sapera CamExpert program, you can generate a camera configuration file (CCF) that describes the desired camera and frame grabber configuration.

Camera files are ASCII text, readable with Windows Notepad on any computer without having Sapera installed.

Overview of Sapera Acquisition Parameter Files (*.ccf)

Typically, a camera application will use a CCF file per camera operating mode. An application can also have multiple CCF files to support different image format modes supported by the camera or sensor (such as image binning or variable ROI).

CCF File Details

CCF files contain the parameters describing the camera image signal characteristics and operation modes (what the camera outputs). The Sapera parameter groups within the file are:

- Image format and pixel definition
- Image 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

CCF files also contain all operating parameters related to the frame grabber board - what the frame grabber can actually do with camera controls or incoming images. 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 image buffer size and buffer type (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.

Saving a Camera File

Use CamExpert to save a camera file (*.ccf) usable with any Sapera demo program or user application.

When CamExpert is setup as required, click on File•Save As to save the new .ccf file. The dialog that opens allows adding details such as camera information, mode of operation, and a file name for the .ccf file.

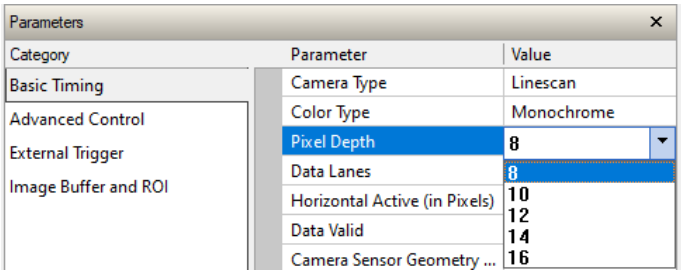
Lookup Table (LUT) Configuration

The following table defines the supported output LUT (look up tables) for the Xtium3-CLHS PX8.

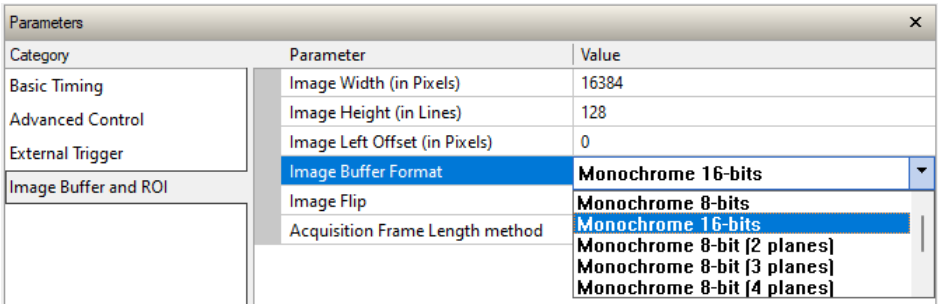
Input Pixel Depth (in bits)	Output Pixel Format	LUT Format	Notes*
8 8	MONO 8 MONO 16	8-in, 8-out 8-in, 16-out	
10	MONO 8	10-in, 8-out	8 MSBs
10	MONO 16	10-in, 16-out	
12	MONO 8	12-in, 8-out	8 MSBs
12	MONO 16	12-in, 16-out	

***** When no LUTs are available or LUTs are disabled, the data is packed in the LSBs of the target destination.

The input Pixel Depth parameter is availllable in Basic Timing category:



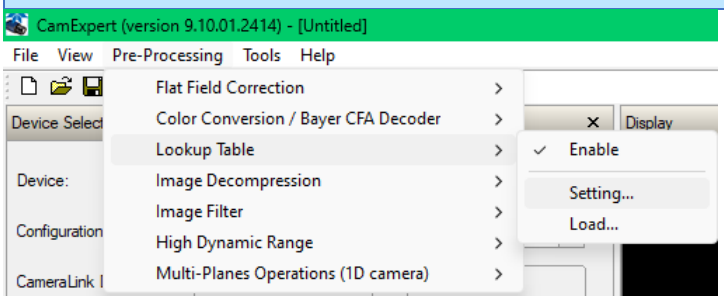
The output pixel format is set using the Image Buffer Format () in the Image Buffer and ROI category:



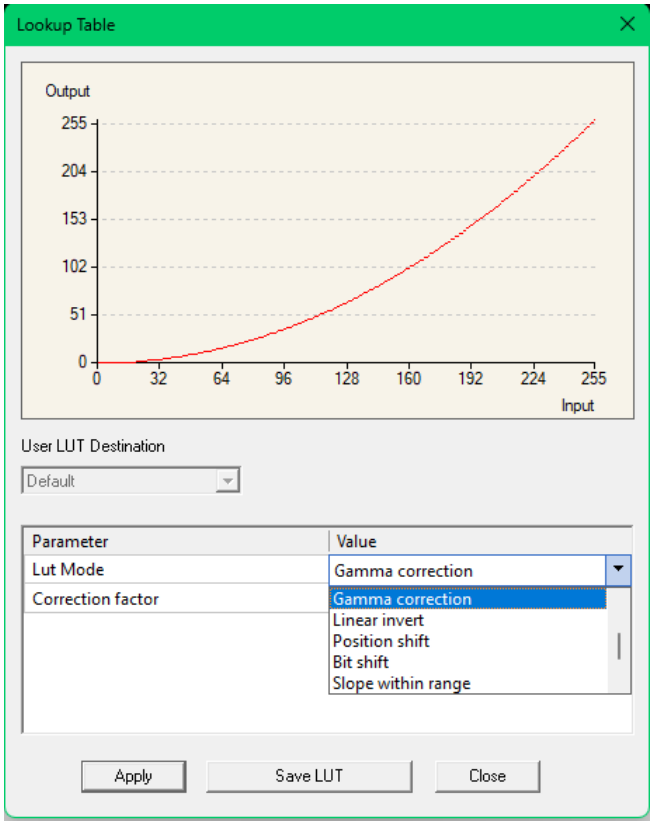
The Lookup Table dialog allows you to configure the LUT. To open this dialog, use the **Pre-Processing > Lookup Table > Setting** menu command.

NOTE

First enable the Lookup Table, otherwise the Setting command will be grayed-out.



To select the type of LUT to apply, use the LUT Modedrop-down list box and choose from the available modes. he Lookup Table dialog displays a graph of the input and output values for the selected LUT.



Click **Apply** to use the specified LUT.

Click **Save LUT** to save the current LUT settings to a Lookup Table File (*.lut).

1. Available LUT modes are:

- **Normal** (default): Modifies all LUT entries using a linear mapping with a positive slope
- **Arithmetic Operations**: modifies all LUT entries using an arithmetic operation
- **Binary pattern**: modifies some LUT entries based on a binary pattern
- **Boolean operation**: Modifies all LUT entries using a Boolean operation
- **Gamma correction**: Modifies all LUT entries using Gamma correction
- **Linear invert**: Modifies all LUT entries using a linear mapping with a negative slope
- **Position shift**: Shifts all pixels by a specific value (with wrap around)
- **Bit shift**: Performs a bit shift bit operation by a specified number of bits
- **Slope within range**: Modifies part of a LUT with a linear mapping
- **Single Threshold**: Modifies all LUT entries using a threshold operation
- **Double Threshold**: Modifies all LUT entries using a threshold operation

Related Sapera Parameters

CORACQ_PRM_PIXEL_DEPTH, CORACQ_PRM_OUTPUT_FORMAT, CORACQ_PRM_LUT_ENABLE

Multi-Planes Image Processing

The CamExpert Multi-Planes Image Processing dialog allows you to enable and select the type of on-board processing to perform on CLHS TDI cameras that output multiple plane images (2, 3, and 4 planes).

Related Sapera LT multi-plane image processing parameters are detailed here: Multi-Plane Input Processing Parameters.

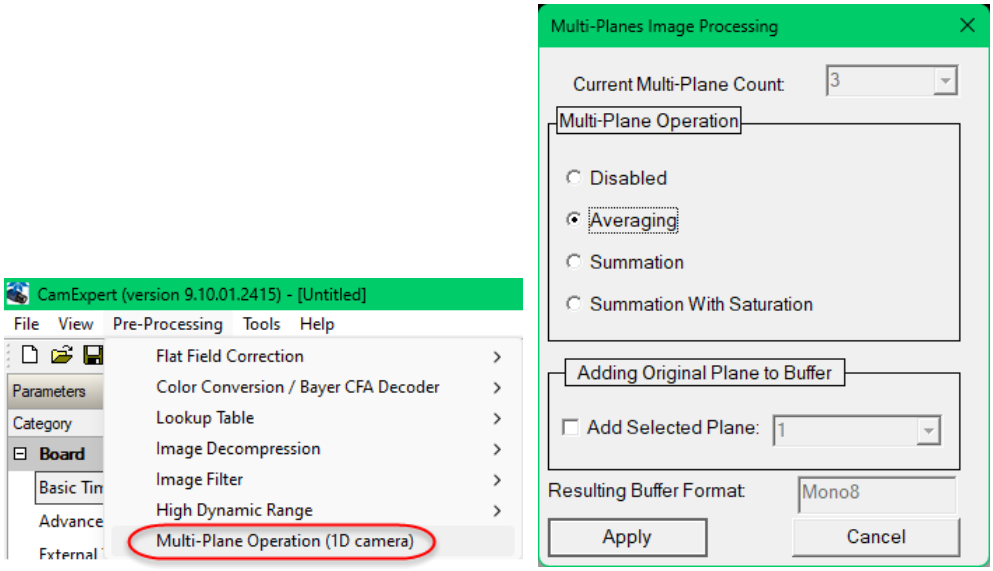
The following table summarizes the supported multi-plane processing for 2, 3 and 4-plane inputs:

Input Planes	**Pixel Depth** (in bits)	**Summation with Saturation**	**Averaging**	**Summation**	**Dual Output** Passthrough & Summation with Saturation	**Dual Output** Passthrough & Averaging
2	8	✔	✔			
	10	✔	✔			
3	8	✔	✔	✔	✔	✔
	10	✔	✔			
	12	✔	✔			
4	8	✔	✔		✔	✔

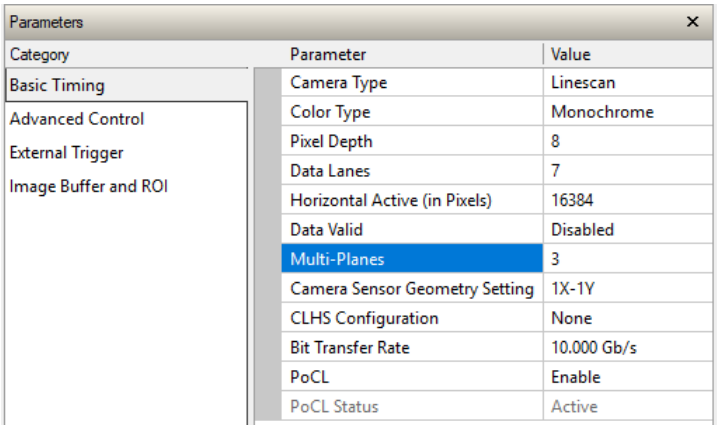
For an overview of the available processing options see:

- 2-Plane Processing Operations
- 3-Plane Processing Operations
- 4-Plane Processing Operations

The Multi-Planes Image Processing dialog is available from the Pre-Processing > Multi-Planes Operations (1D camera) menu.



To ensure that the Current Plane(s) Count is accurate, verify that the Multi-Planes parameter, available in the board's Basic Timing category, is set to the number of planes output by the camera.



2-Plane Processing Operations

2-Plane processing operations support 8 and 10-bit input images.

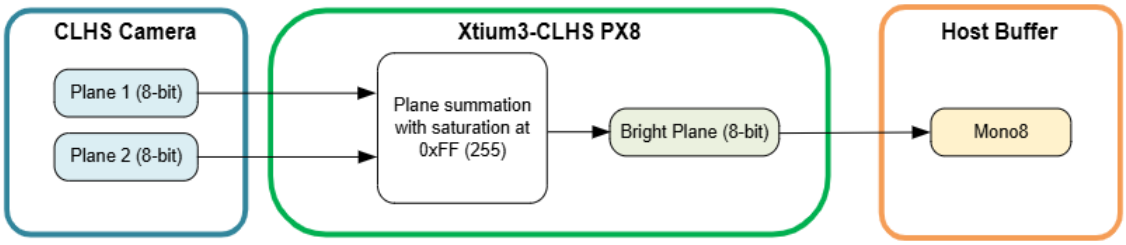
Plane Summation with Saturation: Bright Image Ouput

When acquiring planar images output by a CLHS camera, the Xtium3 PX8 can improve sensitivity in low-light or response situations by summing multiple image planes on-board in real-time to output a single image with extended dynamic range.

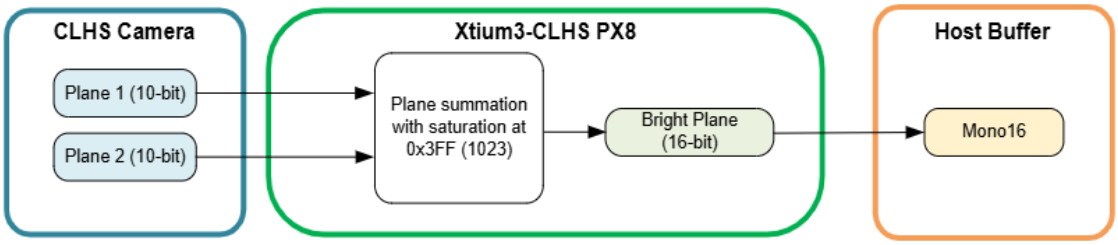
For example, when acquiring a 2-plane image, "Bright" image is created by summing both planes into a single image (sum of Plane 1 & 2).

The summation with saturation operation supports 8 or 10-bit input image planes.

For 8-bit input, the summed (bright) image output is a single Mono8 buffer.



For 10-bit input, the summed (bright) image output is a single Mono16 buffer.



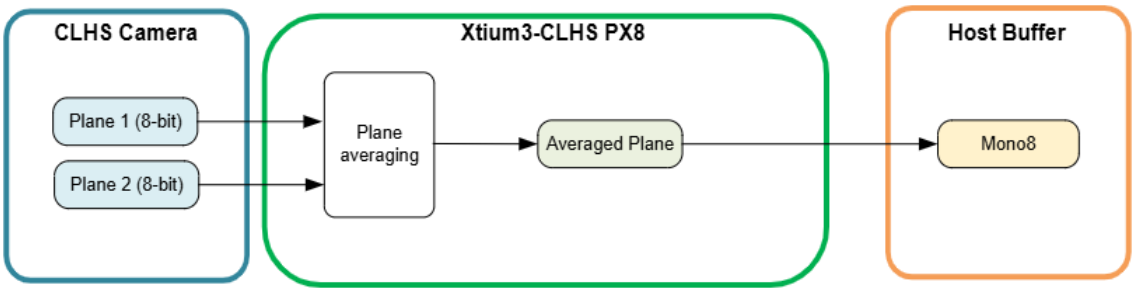
Plane Averaging

When acquiring planar images output by a CLHS camera, the Xtium3 PX8 can average the pixel values of these image planes on-board in real-time to output a single image. This averaging operation reduces the effect of random noise in the output image.

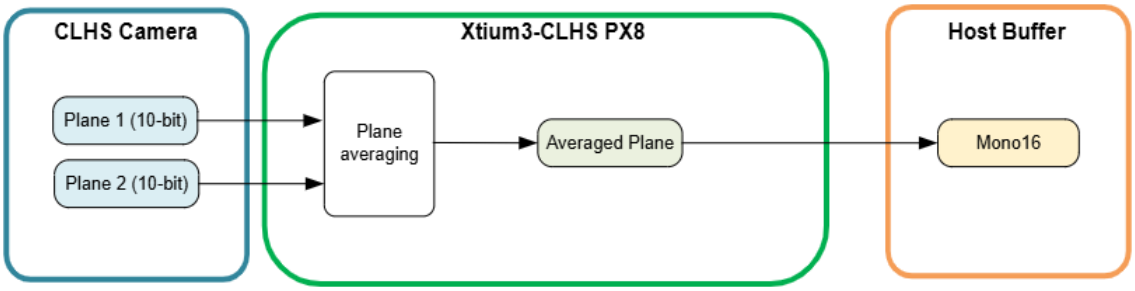
2. Averaged Image Output

The averaging operation supports 8 or 10-bit input image planes.

For 8-bit input, the averaged image output is a single Mono8 buffer.



For 10-bit input, the averaged image output is a single Mono16 buffer.



3-Plane Processing Operations

3-Plane processing summing with saturation and averaging operations support 8, 10 and 12-bit input images.

For summation only and dual output operations, only 8-bit input is supported.

Plane Summation: Bright Image Ouput

When acquiring planar images output by a CLHS camera, the Xtium3 PX8 can improve sensitivity in low-light or response situations by summing multiple image planes on-board in real-time to output a single image with extended dynamic range.

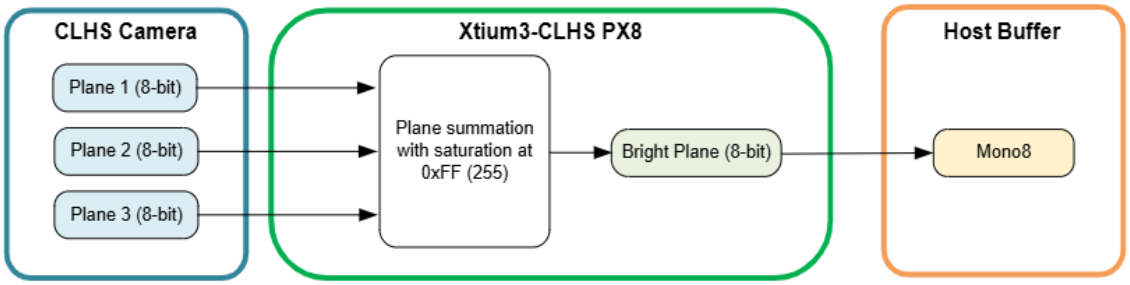
For example, when acquiring a 3-plane image, “Bright” image is created by summing both planes into a single image (sum of Plane 1, 2 and 3).

There are two types of summation available:

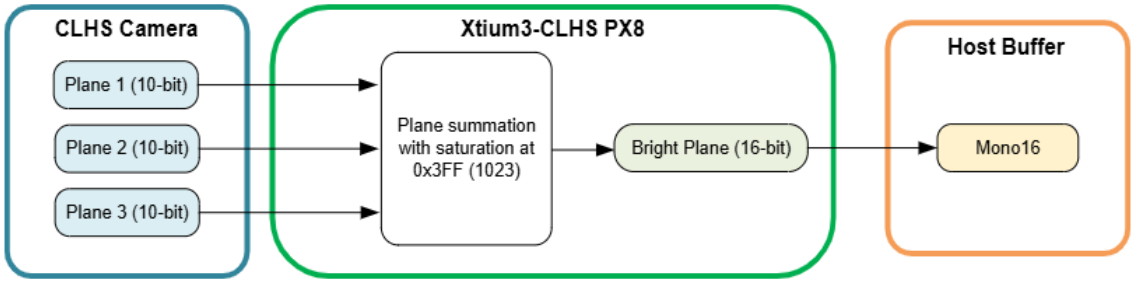
- **Summation with saturation**: 8-bit input with 8-bit output, 10 or 12-bit input with 16-bit output.
- **Summation without saturation**:8-bit input with 16-bit ouput

Summation with Saturation

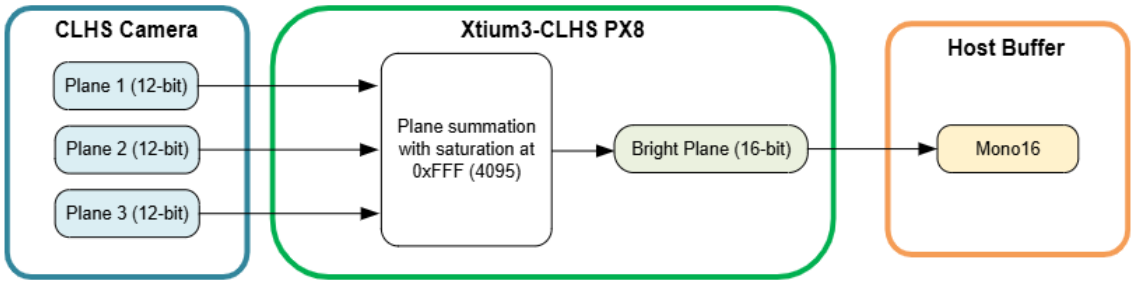
For 8-bit input, summation with saturation outputs the summed image in a Mono8 buffer.



For 10-bit input, summation with saturation outputs the summed image in a Mono16 buffer.

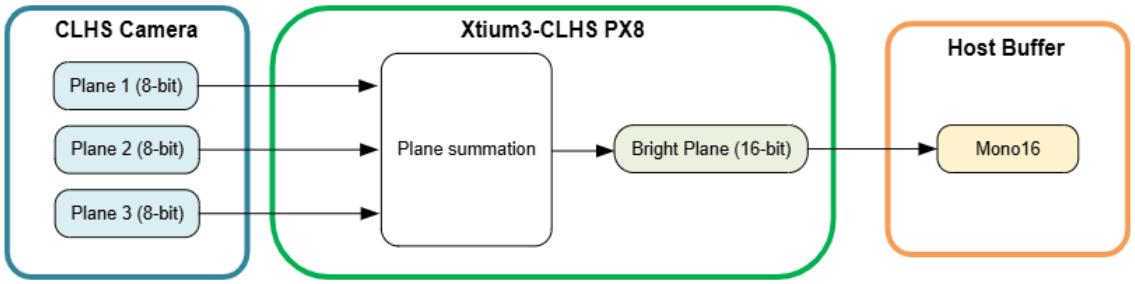


For 12-bit input, summation with saturation outputs the summed image in a Mono16 buffer.



Summation Only (without saturation)

For 8-bit input, summation only (without saturation) outputs the summed image in a Mono16 buffer.



Plane Summation with Saturation Dual Output Option: Dark & Bright Image Output

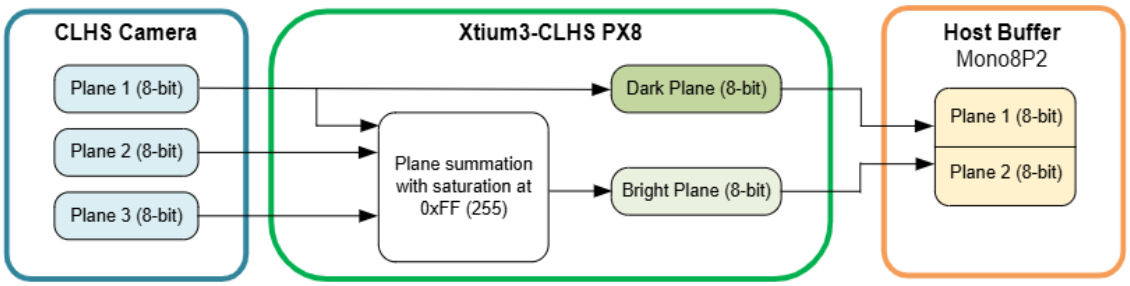
When acquiring 8-bit planar images output by a CLHS camera, the Xtium3 PX8 summation with saturation algorithm can process these image planes on-board in real-time to output a 2-plane image, with an unprocessed passthrough (dark) image and a summed (bright) image.

When acquiring a 3-plane image, one of the planar image is not modified (passthrough) and is considered the “dark” image plane; the “bright” image is created by summing both planes into a single image (bright plane is sum of Plane 1, 2 & 3). This passthrough image is selectable and can be any of the input planes.

The Xtium3 PX8 outputs these two image buffers, the dark and bright, as a Mono8P2 buffer (two planes).

3. Dual Output: Sum and Passthrough

The following example use the 1st plane as the dark plane.



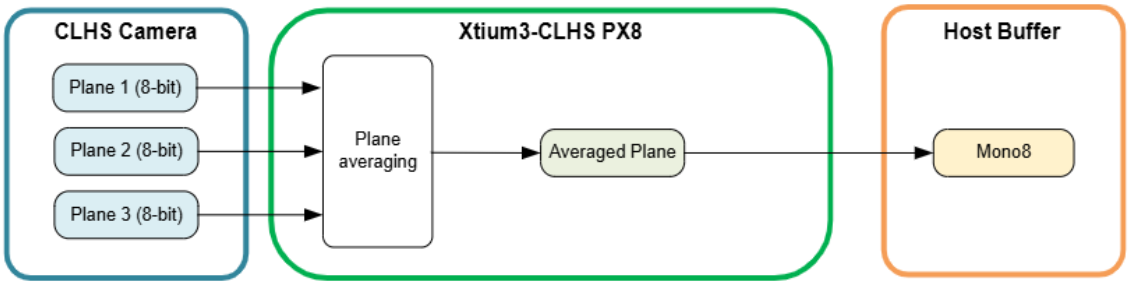
Plane Averaging

When acquiring planar images output by a CLHS camera, the Xtium3 PX8 can average the pixel values of these image planes on-board in real-time to output a single image. This averaging operation reduces the effect of random noise in the output image.

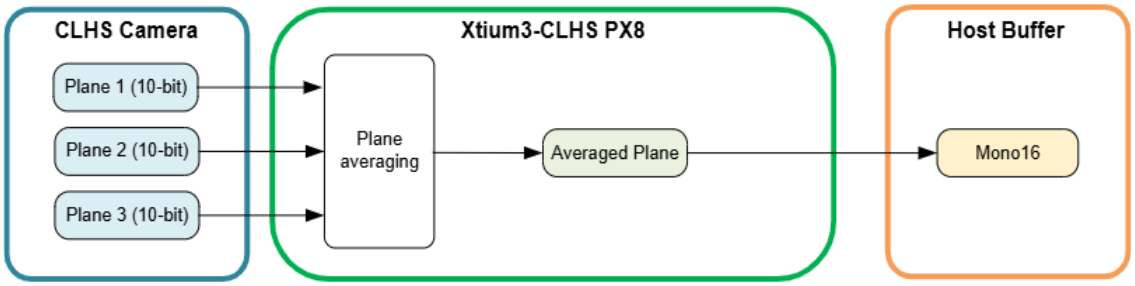
For 3-plane input the averaging operation supports 8-bit intput with 8-bit output, 10 or 12-bit input with 16-bit output.

4. Averaged Image Output

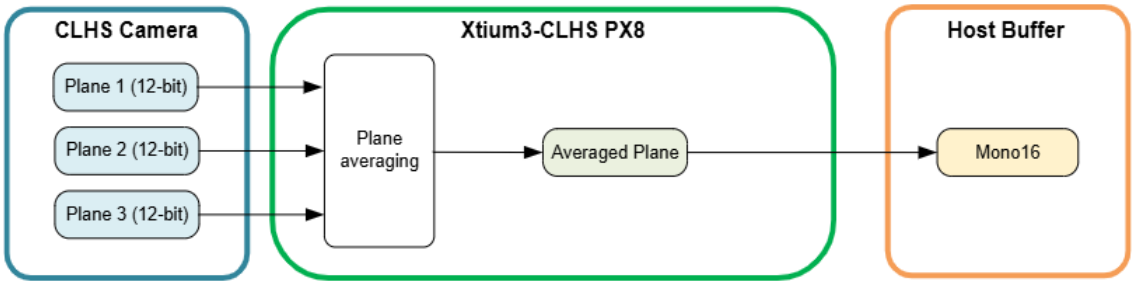
For 8-bit input, the averaging operation outputs the result in a Mono8 buffer.



For 10-bit input, the averaging operation outputs the result in a Mono16 buffer.



For 12-bit input, the averaging operation outputs the result in a Mono16 buffer.



Plane Averaging with Dual Output Option

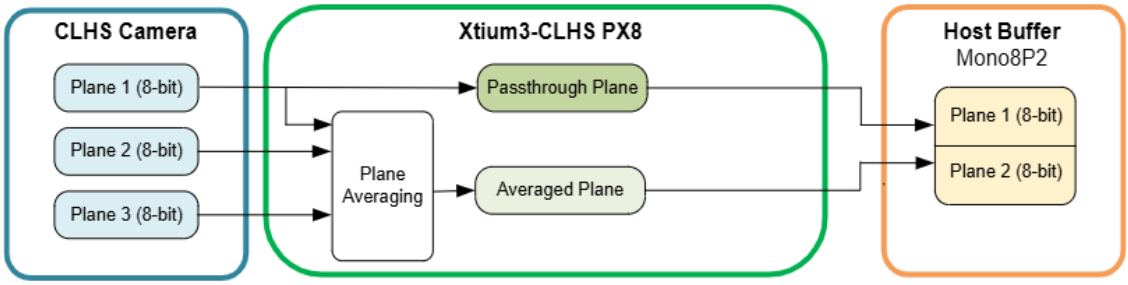
When acquiring 8-bit planar images output by a CLHS camera, the Xtium3 PX8 averaging algorithm can process these image planes on-board in real-time to output a 2-plane image, with an unprocessed passthrough image and an averaged image.

This passthrough image is selectable and can be any of the input planes.

The Xtium3 PX8 outputs these two image buffers, the passthrough and averaged, as a Mono8P2 buffer (two planes).

5. Dual Output: Averaged and Passthrough

The following example uses the 1st plane as the passthrough plane.



4-Plane Processing Operations

4-Plane processing operations support 8-bit input images.

- **NOTE**

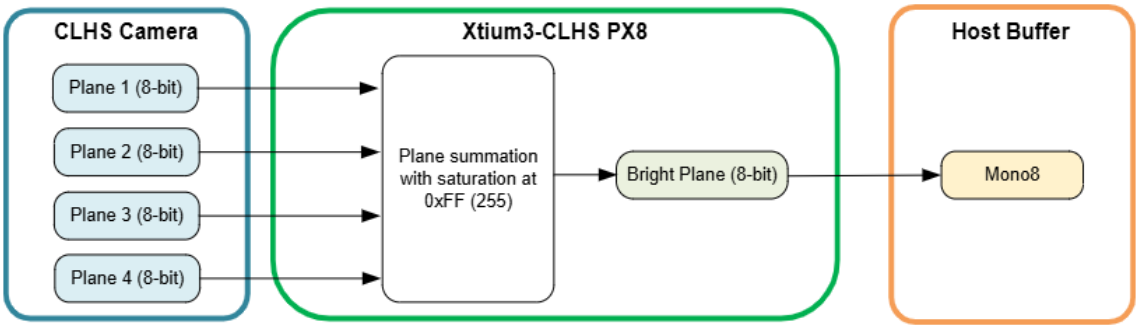
- 4-Plane processing operations require an Advanced Processing license.

Plane Summation: Bright Image Ouput

When acquiring 8-bit planar images output by a CLHS camera, the Xtium3 PX8 can improve sensitivity in low-light or response situations by summing multiple image planes on-board in real-time to output a single image with extended dynamic range.

For example, when acquiring a 4-plane image, “Bright” image is created by summing both planes into a single image (sum of Plane 1, 2, 3 & 4).

For 8-bit input, the summed (bright) image output is a single Mono8 buffer.



Plane Summation with Saturation Dual Output Option: Dark & Bright Image Output

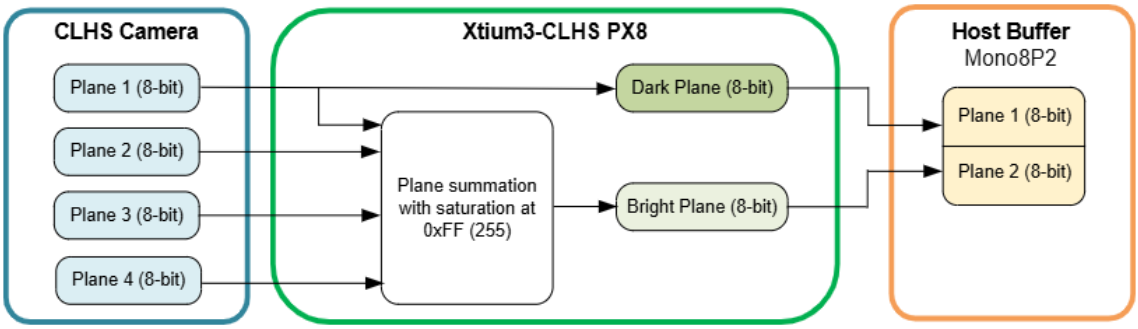
When acquiring planar images output by a CLHS camera, the Xtium3 PX8 summation with saturation algorithm can process these image planes on-board in real-time to output a 2-plane image, with an unprocessed passthrough (dark) image and a summed (bright) image.

When acquiring a 4-plane image, one of the planar image is not modified (passthrough) and is considered the “dark” image plane; the “bright” image is created by summing both planes into a single image (bright plane is sum of Plane 1, 2, 3 & 4). This passthrough image is selectable and can be any of the input planes.

The Xtium3 PX8 outputs these two image buffers, the dark and bright, as a Mono8P2 buffer (two planes).

6. Dual Output: Sum and Passthrough

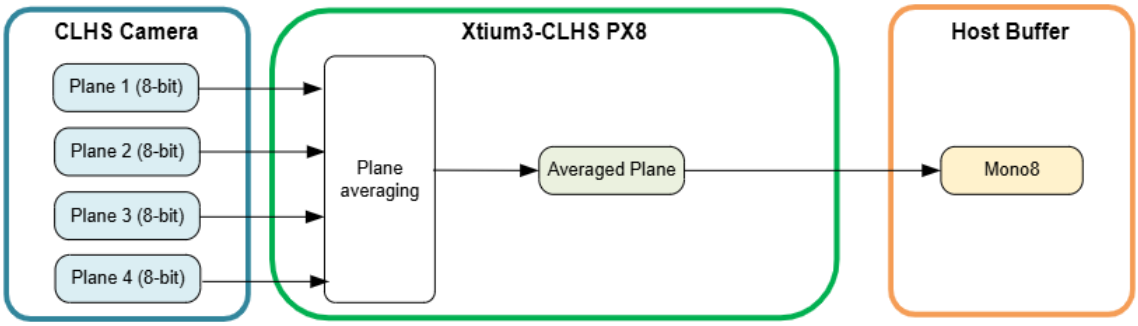
The following example use the 1st plane as the dark plane.



Plane Averaging

When acquiring 8-bit planar images output by a CLHS camera, the Xtium3 PX8 can average the pixel values of these image planes on-board in real-time to output a single image. This averaging operation reduces the effect of random noise in the output image.

7. Averaged Image Output



Plane Averaging with Dual Output Option

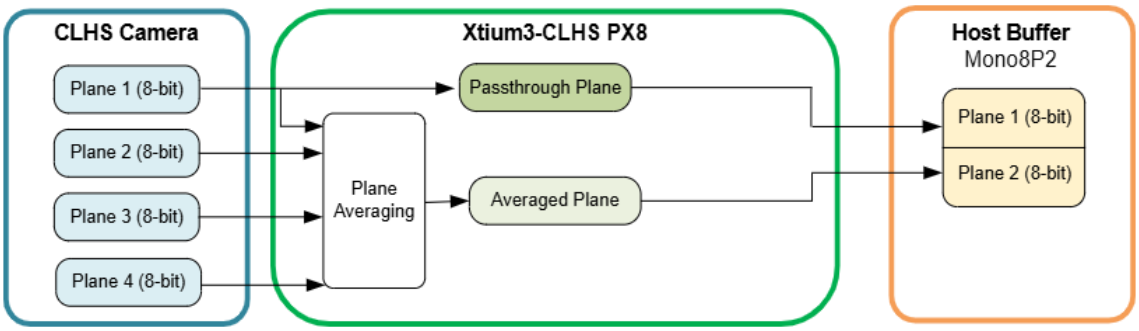
When acquiring 8-bit planar images output by a CLHS camera, the Xtium3 PX8 averaging algorithm can process these image planes on-board in real-time to output a 2-plane image, with an unprocessed passthrough image and an averaged image.

This passthrough image is selectable and can be any of the input planes.

The Xtium3 PX8 outputs these two image buffers, the passthrough and averaged, as a Mono8P2 buffer (two planes).

8. Dual Output: Averaged and Passthrough

The following example uses the 1st plane as the passthrough plane.

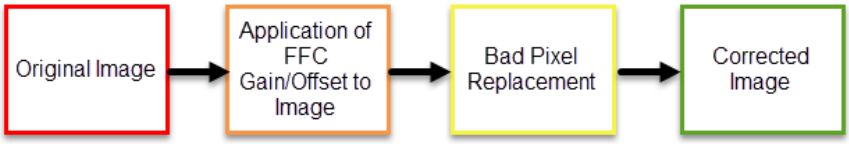


Using the Flat Field Correction Tool

Flat Field Correction is the process of eliminating small gain differences between pixels in a sensor array. A sensor when exposed to a uniformly lit field will have no gray level differences between pixels when calibrated flat field correction is applied to the image.

The flat field correction processing consists of 2 stages:

- The 1st stage applies the FFC Gain/Offset to all of the good pixels.
- The 2nd stage replaces the bad pixels using Neighborhood Replacement:
  Bad pixels are replaced with the average of its 2 neighbors on the same line.



The gain and offset is applied to each good pixel. The maximum gain and offset values are dependent on the pixel depth of the acquired image.

- A Gain of 0 identifies a bad pixel. In this case, no gain or offset will be applied to this pixel.
- Offsets range from 0 to Max Pixel Value.
- Gains range from 1 .. 2.
- Corrected Value = (Original Value – Offset) * ((Gain) /Gain Divisor)
- If the Corrected Value < 0 then Corrected Value = 0;
- If the Corrected Value > Max Pixel Value then Corrected Value = Max Pixel Value

Pixel Depth	Offset Range	Gain Range	Gain Divisor
8	0 .. 255	1 .. 255	128
10	0 .. 1023	1 .. 1023	512
12	0 .. 4095	1 .. 4095	2048

Performing Flat Field Correction

Sapera LT CamExpert includes a tool for performing flat field calibration and generating coefficient files. The CamExpert Flat Field tool functions with hardware supporting flat field processing. Generating flat field correction offset and gain coefficients requires acquiring dark and bright images of a uniform target.

The imaging setup, such as the lighting, lens and aperture settings, should be in the same state as in the intended application; if these characteristics change, the flat field calibration should be redone.

NOTE

When acquiring multiple planes a flat field correction is applied to each plane. If multi-plane processing is enabled, flat field correction is applied after the operation on the resulting output plane(s); in this case, flat field calibration should be done with required multi-plane processing operation enabled.

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.

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, a controlled diffused light source aimed directly at the lens should be used. 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 (8-bit acquisition).

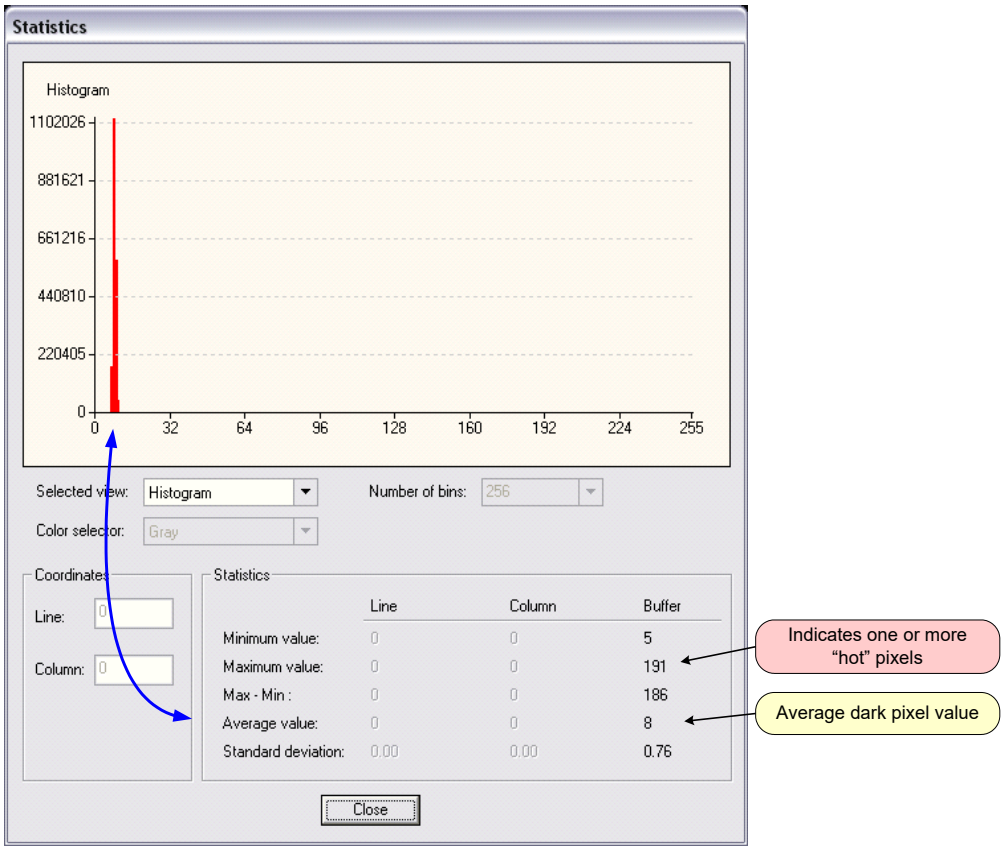


Figure 29: CamExpert Histogram of Dark Image

NOTE

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.

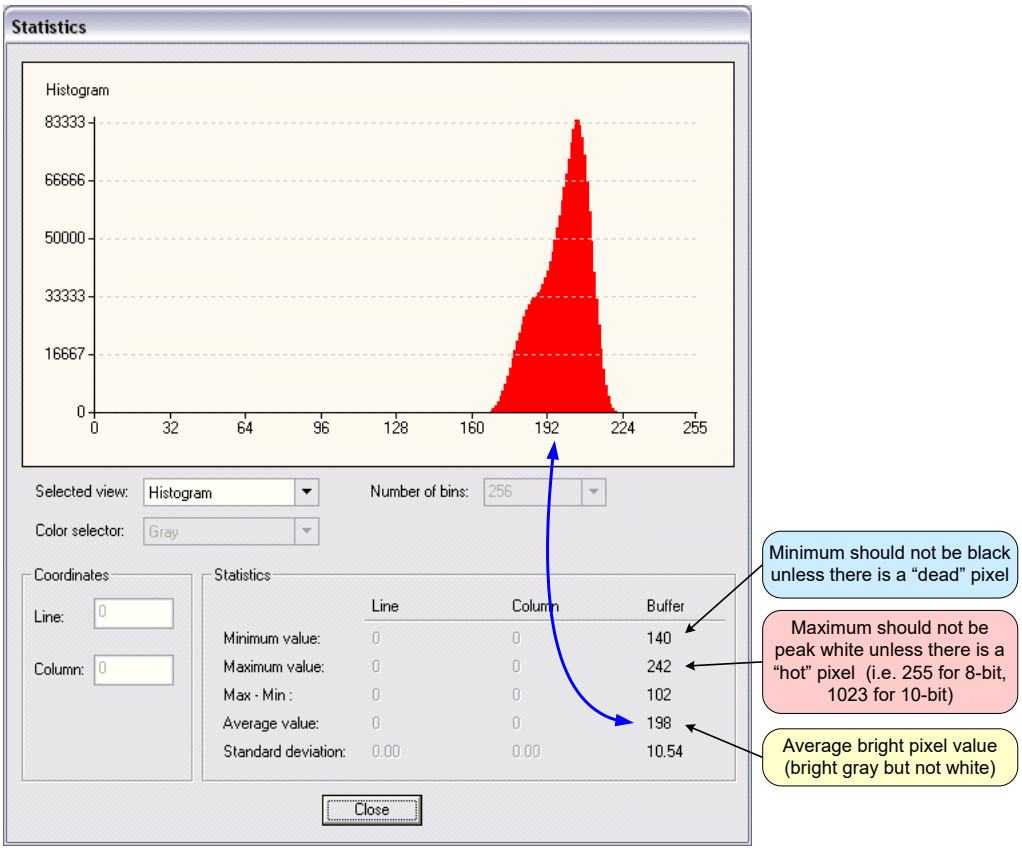


Figure 30: CamExpert Histogram of Bright Image

NOTE

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 so as to be able to repeat it during the calibration procedure.

Flat Field (Line) Correction Calibration Procedure

Calibration is the process of taking two reference images, one of a black field (line) – one of a light gray field (not saturated), to generate correction data for images captured by the sensor. Each sensor 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**.

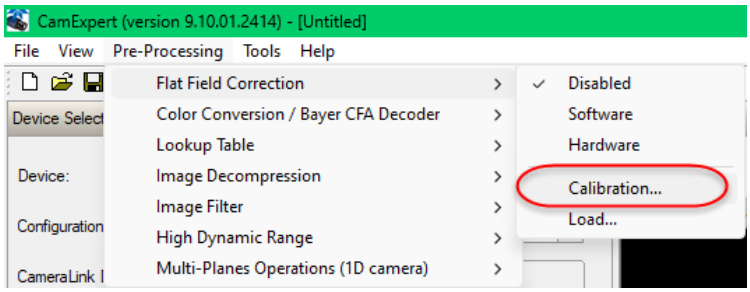


Figure 31: CamExpert Flat Field Correction Menu Command

Flat Field Correction Dialog

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

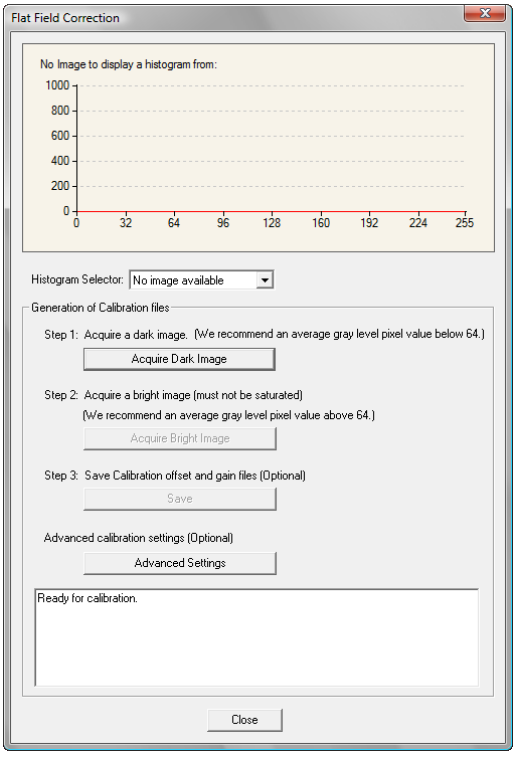
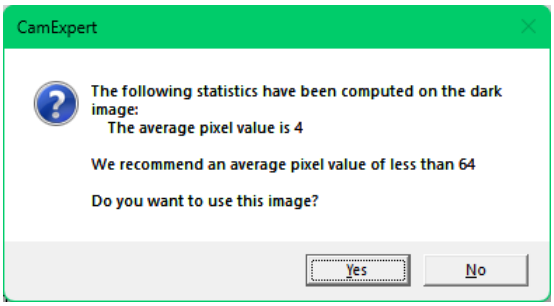


Figure 32: CamExpert Flat Field Correction Dialog

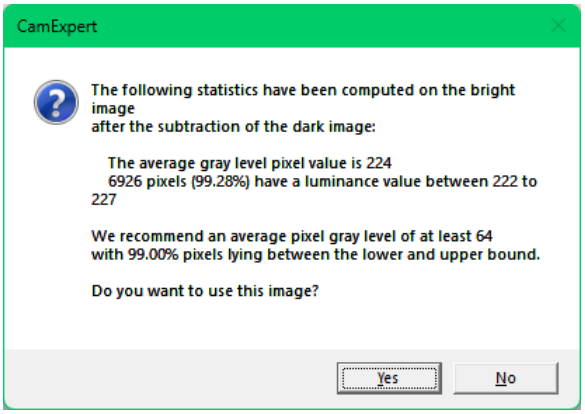
NOTE

Click **Advanced Settings** to modify the default calibration settings, such as the number of frames or lines to average and the target DN for the dark and bright images.

- 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 number of image frames to average (default = 10), analyze the pixel gray level spread, and present the statistics.



  The black reference image should have pixel values less than 20. If acceptable, accept the image as the black reference.
- Setup the camera to acquire a uniform white image (but not saturated white). Even illumination on white paper can be used, with a minimum gray level of 128. It is preferable to prepare for the white level calibration before the calibration procedure.
- Click on **Acquire White Image**. The flat field demo will grab an image frame, analyze the pixel gray level spread, and present the statistics. The captured gray level for all pixels should be greater than 128. If acceptable, accept the image as the white reference.



- Click on **Save**. The flat field correction data is saved as a TIF image with a file name of your choice (such as camera name and serial number).

Applying Flat Field Correction

From the CamExpert menu bar enable Flat Field correction (**Pre-Processing • 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 **Pre-Processing • Flat Field Correction • Load** to load in a flat field correction image from previous saved calibration data. CamExpert allows saving and loading calibration data for all cameras used with the imaging system.

Sapera Demo Applications

The Sapera Explorer application is the fastest way to access the demo programs, code samples, examples and other info provided with Sapera LT.

To open Sapera Explorer

- On the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Explorer**.

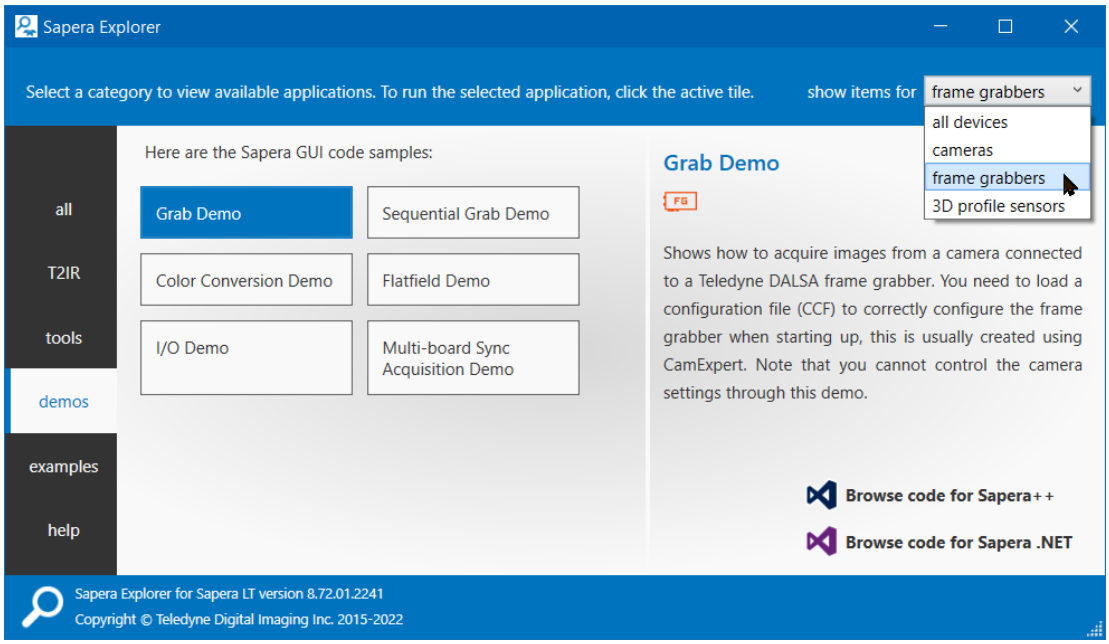


Figure 33: The Sapera Explorer application

All demo programs are available as a compiled binary; source code is provided for both C++ and .NET projects using Visual Studio 2013/2015/2017/2019.

Table 6: Demos Workspace Details

Program file	…\...\Sapera\Demos\Binaries\GrabDemo.exe
Visual C++ Solution	…\...\Sapera\Demos\Classes\Vc\SapDemos_2013.sln …\...\Sapera\Demos\Classes\Vc\SapDemos_2015.sln …\...\Sapera\Demos\Classes\Vc\SapDemos_2017.sln …\...\Sapera\Demos\Classes\Vc\SapDemos_2019.sln
Visual .NET Solution	…\...\Sapera\Demos\NET\SapNETCSDemos_2013.sln …\...\Sapera\Demos\NET\SapNETCSDemos_2015.sln …\...\Sapera\Demos\NET\SapNETCSDemos_2017.sln …\...\Sapera\Demos\NET\SapNETCSDemos_2019.sln …\...\Sapera\Demos\NET\SapNETCSDemos_2019.Core.sln
Remarks	This demo is based on Sapera LT classes. See the Sapera User's and Reference manuals for more information.

Grab Demo Overview

The Grab Demo program demonstrates the basic acquisition functions included in the Sapera library. The program either allows you to acquire images, in continuous or in one-time mode, while adjusting the acquisition parameters. The program code may be extracted for use within your own application.

Using the Grab Demo

To open the Grab Demo program

- In **Sapera Explorer**, on the left pane select **Demos**, then double-click **Grab Demo**.

Alternately, from the **Start** menu, select **Teledyne DALSA Sapera LT > Sapera++ Demos (Executables)**, which will open the folder where **GrabDemo.exe** is located.

Acquisition Configuration Window

When using frame grabbers, the demo program first displays the Acquisition Configuration window, where you can select from the available Sapera acquisition servers and input devices present on the selected server.

- From the **Acquisition Server** list, select and installed Sapera acquisition server (installed Teledyne acquisition hardware using Sapera drivers).
- From the **Acquisition Device** list, select an available input device on the selected driver.
- Click **Browse** to select the required camera configuration file for the connected camera. Sapera camera files contain timing parameters and image 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 CamExpert program to generate the camera configuration file (*.ccf) based on timing and control parameters entered. The CamExpert live acquisition window allows immediate verification of those parameters.

Grab Demo Main Window

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

Refer to the Sapera LT Getting Started Manual for Frame Grabbers, chapter **Demos and Examples**, for more information on the Grab Demo and other programs provided with Sapera LT.

General I/O Demo Overview

This program demonstrates the usage of General I/O functionality included in Sapera LT. It allows you to monitor the current state of the input and output pins and change their state between low and high polarity. It also allows you to monitor interrupts on the input pins and to count the number of incoming interrupts.

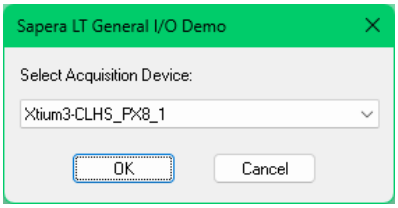
Using the General I/O Demo

To open the General I/O Demo program

- In **Sapera Explorer**, on the left pane select **Demos**, then double-click **IO Demo**.

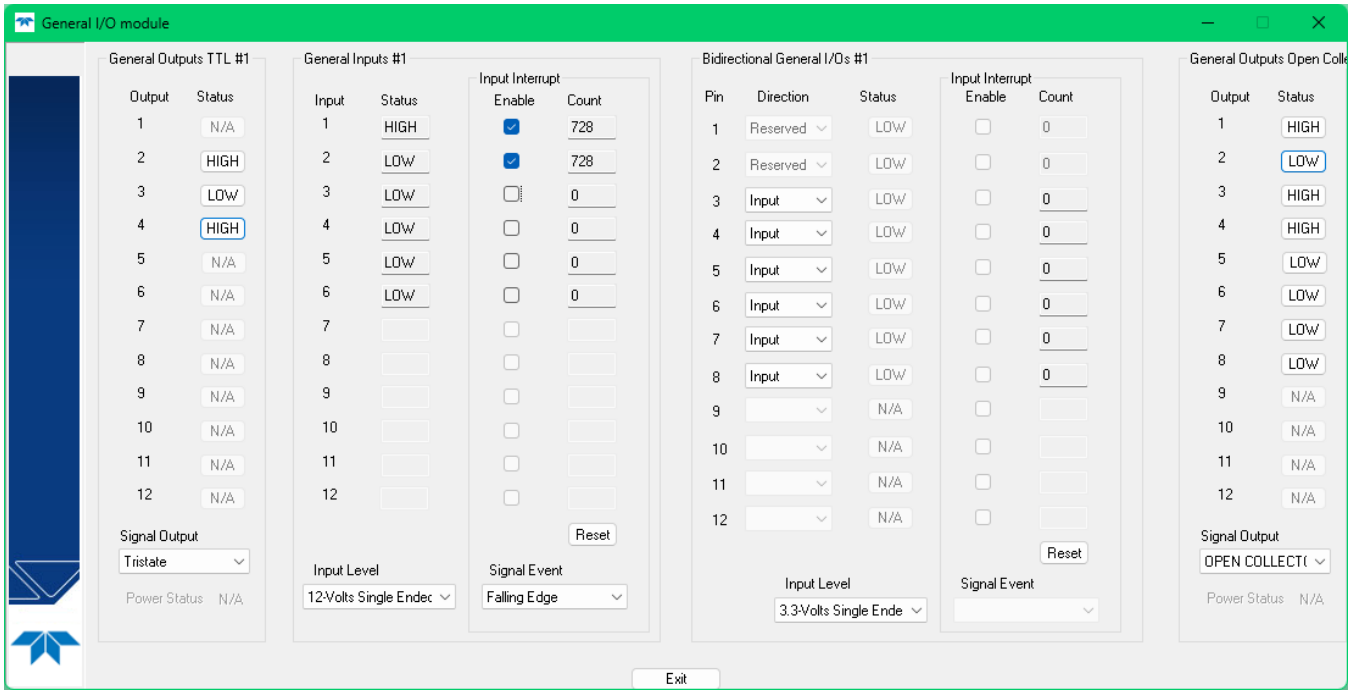
Alternately, from the **Start** menu, select **Teledyne DALSA Sapera LT > Sapera++ Demos (Executables)**, which will open the folder where**IODemo.exe** is located.

The demo program first displays a dialog where you can select from the available Sapera acquisition devices.



General I/O Demo Main Window

The General I/O Demo program displays the available IO modules for the frame grabber. Use the HIGH/LOW buttons to change the pin state for outputs, as well as monitoring events for inputs. In addition, bi-directional multi-sync board-to-board IOs can be set as input/ouput.



Trigger to Image Reliability

Trigger-to-image reliability incorporates all stages of image acquisition inside an integrated controller to increase reliability and simplify error recovery. The trigger-to-image reliability model brings together all the requirements for image acquisition to a central management unit. These include signals to control camera timing, on-board frame buffer memory to compensate for PCI bus latency, and comprehensive error notification.

If the Xtium3-CLHS PX8 detects a problem, the application can take appropriate action to return to normal operation. With the Xtium3-CLHS PX8, applications ensure trigger-to-image reliability by monitoring events and controlling transfer methods as described below:

Trigger-to-Image reliability manages:

- High bandwidth acquisition
- High frequency acquisition
- High lines or images rate per second
- High number of cameras
- Image buffer memory
- Statistics
- In-camera event status flags

Teledyne's T2IR framework includes powerful GUI based tools for continuous monitoring and rapid pinpoint of errors that are hard to trace back. This continuous system monitoring and deep debugging tools help reduce downtime. This is done with the help of the following tools:

- Sapera Monitor (available with installation of Sapera LT)
- External LEDs
- Teledyne LogViewer
- PCI Diagnostic Tool
- Sapera Configuration (available with installation of Sapera LT)
- Xtium3 Diagnostic Tool

Refer to the Xtium3-CLHS PX8 Utilities section for more information on these tools.

In addition, the Xtium3 supports frame grabber specific functions (not available in the standard Sapera LT API) that provide information on the PCIe connection details, such as the number of data lanes and bandwidth, directly in the user application (this information is also available in theXtium3 Diagnostic Tool).

Supported Events and Transfer Methods

Listed below are the supported acquisition and transfer events. Event monitoring is a major component to the Trigger-to-Image Reliability framework. Most of these events are available through the Sapera Monitor for easy debugging.

Acquisition Events

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

Table 7: Acquisition Events

Event	Sapera Parameter	Description
External Trigger (Used/Ignored/End)	CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_END	Generated when the external trigger pin is asserted, which indicates the start of the acquisition process. There are three types of external trigger events: 'Used', 'Ignored' and 'End'. Following an external trigger, if the event generates a captured image, an External Trigger Used event will be generated. If there is no captured image, an External Trigger Ignored event will be generated. An external trigger event is ignored if the event rate is higher than the possible frame rate of the camera. When using an active high/low or double pulse rising/falling edge trigger detection, the 2nd pulse will generate the External Trigger End event.
Start of Frame	CORACQ_VAL_EVENT_TYPE_START_OF_FRAME	Event generated during acquisition, with the detection of the start of an image frame by the board acquisition hardware.
End of Frame	CORACQ_VAL_EVENT_TYPE_END_OF_FRAME	Event generated during acquisition, with the detection of the end of an image frame by the board acquisition hardware.
Data Overflow	CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW	The Data Overflow event indicates that there is not enough bandwidth for the acquired data transfer without loss. Data Overflow would occur with limitations of the acquisition module and should never occur.
Frame Valid	CORACQ_VAL_EVENT_TYPE_VERTICAL_SYNC	Event generated on detection of the start of an image frame by the board acquisition hardware. Acquisition does not need to be active; therefore, this event can verify a valid signal is connected.
Link Lock/Unlock	CORACQ_VAL_EVENT_TYPE_LINK_LOCK CORACQ_VAL_EVENT_TYPE_LINK_UNLOCK.	Event generated on the transition from locking or not locking, of the camera required lanes.
Slave Link Lock/Unlock	CORACQ_VAL_EVENT_TYPE_SLAVE_LINK_LOCK CORACQ_VAL_EVENT_TYPE_SLAVE_LINK_UNLOCK.	Event generated on the transition from locking or not locking of a slave frame grabber connected to the data forwarding connector.
Frame Lost	CORACQ_VAL_EVENT_TYPE_FRAME_LOST	The Frame Lost event indicates that an acquired image failed to transfer to on-board memory. An example is 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 or no host buffers are available to receive an image.
External Line Trigger Too Slow	CORACQ_VAL_EVENT_TYPE_EXT_LINE_TRIGGER_TOO_SLOW	Event which indicates that the detected rotary encoder input tick rate is too slow for the device to take into account the specified encoder multiplier value.
Line Trigger Too Fast	CORACQ_VAL_EVENT_TYPE_LINE_TRIGGER_TOO_FAST	Event which indicates a previous line-trigger did not generate a complete image line from the camera. Note that due to jitter associated with using rotary encoders, the acquisition device can delay a line trigger if a previous line has not yet completed. This

		event is generated if a second line trigger comes in while the previous one is still pending.
Shaft Encoder Reverse Count Overflow	CORACQ_VAL_EVENT_TYPE_SHAFT_ENCODER_REVERSE_ COUNT_OVERFLOW	Event which indicates that the rotary encoder has travelled in the opposite direction expected and that the number of pulses encountered during that travel has exceeded the acquisition device counter. The acquisition device will thus not be able to skip the appropriate number of pulses when the expected direction is detected.
Camera Missed Trigger	CORACQ_VAL_EVENT_TYPE_CAMERA_MISSED_TRIGGER	Event which indicates that the camera could not respond to a trigger request as it was busy servicing a previous trigger request.
Camera Overrun	CORACQ_VAL_EVENT_TYPE_CAMERA_OVERRUN	Event which indicates that the camera data and/or image has been corrupted due to insufficient buffer space in the camera.
Link Error		Event which indicates that an error has occurred on one or more of the lanes. Information about the source of the link error and the number of occurances of this error can be retrieved using the SapAcqCallbackInfo class. GetGenericParam0: returns the source of the error: 1: CRC Error 2: 64/66B Error 3: Packet Buffer Overflow 4: Packet Resend 5: 64B/66B Corrected Error GetGenericParam1: returns a bitfield indicating which lane(s) generated the error Bit 0 = Lane 1, Bit 1= Lane 2, … GetCustomSize: returns 7 * UINT32 GetCustomData: returns the number of errors per lane. There are 7 entries, each entry being a UINT32.
Slave Link Lock/Unlock	CORACQ_VAL_EVENT_TYPE_SLAVE_LINK_LOCK CORACQ_VAL_EVENT_TYPE_SLAVE_LINK_UNLOCK	Event generated on the transition from locking or not locking, of the slave communication lane.

IO Events

IO events are related to the GIO module input pins.

Table 8: IO Events

Event	Sapera Parameter	Description
Rising Edge	CORGIO_VAL_EVENT_TYPE_RISING_EDGE	Rising edge of I/O pin state transition (low to high).
Falling Edge	CORGIO_VAL_EVENT_TYPE_FALLING_EDGE	Falling edge of I/O pin state transition (high to low)

Transfer Events

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

Table 9: Transfer Events

Event	Sapera Parameter	Description
Start of Frame	CORXFER_VAL_EVENT_TYPE_START_OF_FRAME	Start of Frame event generated when the first image pixel is transferred from on-board memory into PC memory.
End of Frame	CORXFER_VAL_EVENT_TYPE_END_OF_FRAME	End of Frame event generated when the last image pixel is transferred from on-board memory into PC memory.
End of Line	CORXFER_VAL_EVENT_TYPE_END_OF_LINE	The End of Line event is generated after an image line is transferred to a PC buffer.
End of N Lines	CORXFER_VAL_EVENT_TYPE_END_OF_NLINES	The End of N Lines event is generated after a set number of image lines are transferred to a PC buffer.
End of Transfer	CORXFER_VAL_EVENT_TYPE_END_OF_TRANSFER	End of Transfer event generated at the completion of the last image transfer from on-board memory into PC memory. Issue a stop command to the transfer module to complete a transfer (if transfers are already in progress). If a frame transfer of a fixed number of images is requested, the transfer module will stop transfer automatically.

Trigger Signal Validity

The Xtium3 ignores external trigger signal noise with its programmable debounce control. Program the debounce parameter for the minimum pulse duration considered as a valid external trigger pulse. For more information see External Trigger Debouncing.

Supported Transfer Cycling Methods

Images are accumulated in on-board memory in a FIFO type manner. On-board memory can get filled up if the rate at which the images are acquired is greater than the rate at which the DMA engine can write them to host buffer memory. On-board memory can also get filled-up if there are no more empty buffers available to transfer the on-board images.

When no memory is available for a new image to be stored in on-board memory, the image is discarded and a CORACQ_VAL_EVENT_TYPE_FRAME_LOST or trash buffer callback is generated. If a CORACQ_VAL_EVENT_TYPE_FRAME_LOST occurs when host buffers are available, it can indicate a problem with the PX8 bus bandwidth.

If image buffers are constructed using a trash buffer (SapBufferWithTrash using a transfer cycle mode with trash), when no host buffers are available and no memory is available for a new image to be stored in on-board memory, the SapXferCallBackInfo::IsTrash (C++) function or SapXferNotifyEventsArgs.Trash (.NET) property returns true. If a trash callback function has been registered during construction of the SapTransfer object, it will be executed when a trash event occurs.

When stopping the image acquisition, the event CORXFER_VAL_EVENT_TYPE_END_OF_TRANSFER will occur after all images currently in the on-board memory are transferred to host buffer memory.

NOTE

If the application does not provide enough empty buffers, the Xtium3 event will not occur and an acquisition abort will be required.

The Xtium3-CLHS PX8 supports the following synchronous or asynchronous transfer modes:

Table 10: Transfer Cycle Modes

Mode	Sapera Parameter	Description
Synchronous with Trash	CORXzFER_VAL_CYCLE_MODE_SYNCHRONOUS_WITH_TRASH	Before cycling to the next buffer in the list, the transfer device will check the next buffer's state. If its state is full, the transfer will keep the image in on-board memory until the next buffer's state changes to empty. If the on-board memory gets filled, trash callbacks will be generated.
Synchronous Next Empty with Trash	CORXFER_VAL_CYCLE_MODE_SYNCHRONOUS_NEXT_EMPTY_WITH_TRASH	When starting an acquisition, the buffer list is put in an empty buffer queue list in the exact order they were added to the transfer. Whenever a user sets a buffer to empty, it is added to the empty buffer queue list, so that after cycling once through the original buffer list, the buffers acquired into will follow the order in which they are put empty by the user. So in this mode, the on-board images will be transferred to host buffer memory as long as there are buffers in the empty buffer queue list. If no buffers are available on the host and the on-board memory gets filled, trash callbacks will be generated.
Asynchronous	CORXFER_VAL_CYCLE_MODE_ASYNCHRONOUS	The transfer device cycles through all buffers in the list without concern about the buffer state.

The following table describes the possible buffer states and resulting behavior:

Table 11: Buffer States and Behaviors

Trash Buffer (cycling mode with trash)	**Xtium3 On-Board Memory State**	**Host Sapera Buffer State**	**Resulting Event**
NO	Empty buffer available (at least 1)	Empty buffer available (at least 1)	Normal acquisition events
NO	Empty buffer available (at least 1)	Full	Acquire into Xtium3 on-board memory
NO	Full	Empty buffer available (at least 1)	Frame Lost Event
NO	Full	Full	Frame Lost Event
YES	Empty buffer available (at least 1)	Empty buffer available (at least 1)	Normal acquisition events
YES	Empty buffer available (at least 1)	Full	Acquire into Xtium3 on-board memory
YES	Full	Empty buffer available (at least 1)	Frame Lost Event
YES	Full	Full	Trash Callback

By default, the buffer state (empty or full) is automatically managed by Sapera LT; it can be managed manually by the user if necessary.

Xtium3-CLHS PX8 Utilities

The Windows Xtium3-CLHS PX8 driver installation includes the following utility programs:

- Device Manager
- Firmware Loader
- Teledyne LogViewer
- Xtium3 Diagnostic Tool
  PCI Diagnostic Tool

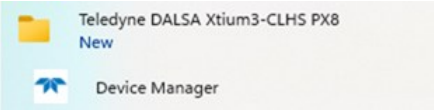
The Windows Sapera LT installation includes these additional utility programs:

- Sapera Monitor
- Sapera Configuration

Device Manager

The Device Manager utility allows users to perform firmware updates, reset the board, and configure various board settings, such as GPIOs. The Device Manager program also displays information about the Xtium3-CLHS PX8 boards installed in the system.

It is available from the Windows Start menu under **Teledyne DALSA Xtium3-CLHS PX8 • 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 Teledyne boards in the system and their individual device components. The right window displays the information stored in the selected board device.

Generate the Xtium3-CLHS PX8 device manager report file (BoardInfo.txt) by clicking **File • Save Device Info**. Teledyne DALSA Technical Support may request this report to aid in troubleshooting installation or operational problems.

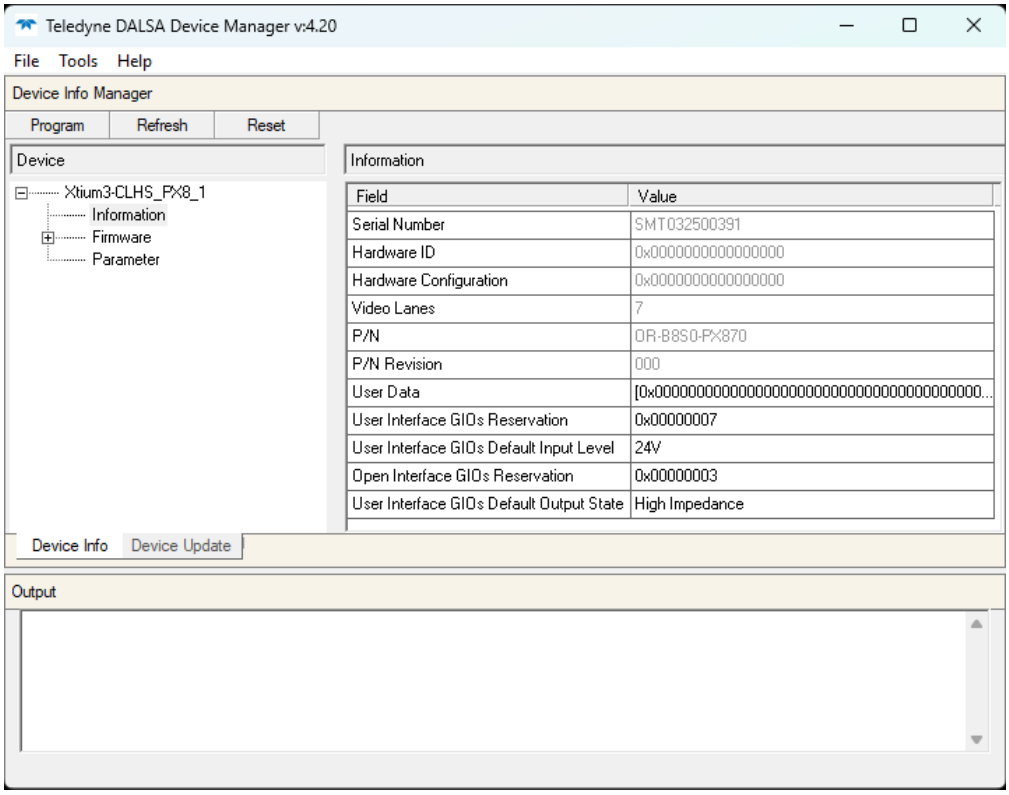
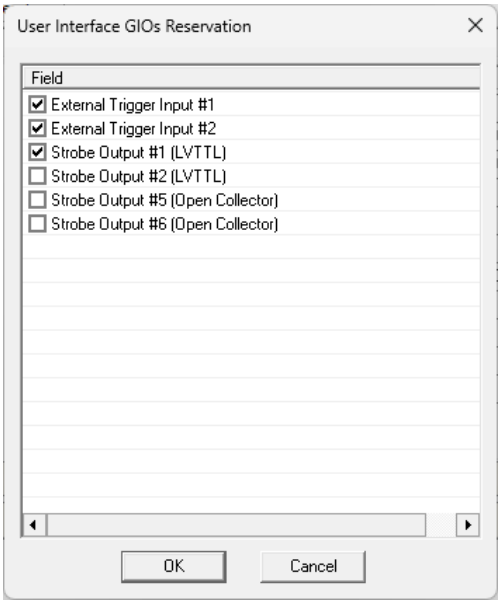
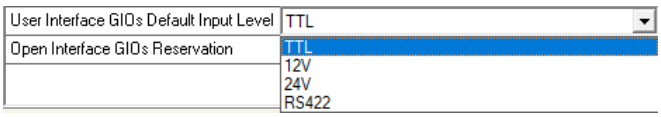
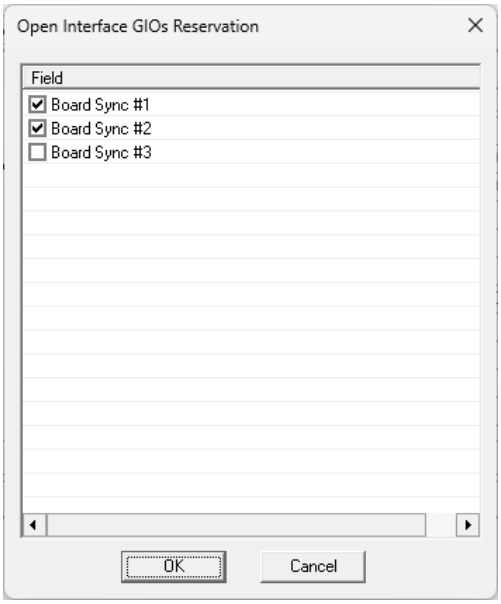
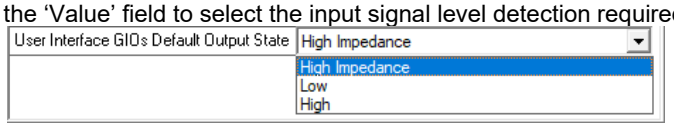


Figure 34: Board Information via Device Manager

Information Field Description

Field	Description
Serial Number	[Read-Only]: Serial Number of the board.
Hardware ID	[Read-Only]: Identifies hardware changes that affect the operation of the board.
Hardware Configuration	[Read-Only]: States the presence or absence of optional components. Possible values are: • **0x0000000000000000**
Video Lanes	[Read-Only]: Indicates the maximum number of video (image or data) lanes supported by the board. Xtium3-CLHS PX8: The value is 7.
User Interface Outputs	[Read-Only]: Number of available user interface outputs on the board. For this board, the value is 12.
P/N	[Read-Only]: Indicates the part number of the board. OR-B8S0-PX870
P/N Revision	[Read-Only]: Indicates the revision of the part number.
User Data	[Read/Write]: This is a 64 byte general purpose user storage area. For information on how to read/write this field at the application level, contact Teledyne Technical Support.
User Interface GIOs Reservation	[Read/Write]: Use this field to reserve User Interface GIOs for use by the acquisition module. By default, boards are shipped with User Interface General Inputs 1 & 2 reserved for External Triggers and User Interface General Output 1 reserved for Strobe Output. Click on the 'Value' field to open the dialog box show below. Disable any GIO reservations that are not required. Click the OK button to update the value field. 
User Interface GIOs Default Input Level	[Read/Write]: Use this field to select the default input level of the User Interface GIOs. Click on the 'Value' field to select the input signal level detection required.  By default, boards are shipped with User Interface General Inputs set to 24V. Note that the input level can also be modified at the application level.

Open Interface GIOs Reservation	[Read/Write]: Use this field to reserve Open Interface GIOs for use by the acquisition module. To specify the open interface GIO reservations, click on the 'Value' field to open the Open Interface GIOs Reservation dialog box. Disable any GIO reservations that are not required. Click OK to update the value field.  By default, boards are shipped with Open Interface GIOs 1 & 2 reserved for Board Sync 1 & 2.
User Interface GIOs Default Output State	[Read/Write]: Use this field to select the default Output State of the User Interface GIOs. Click on the 'Value' field to select the input signal level detection required.  By default, boards are shipped with User Interface General Outputs set to High Impedance. Note that the output state can also be modified at the application level.

Firmware Information

The Firmware Information panel displays details about the currently installed firmware, such as the Version number and its creation date. The Modified field indicates when the firmware was installed on the host machine.

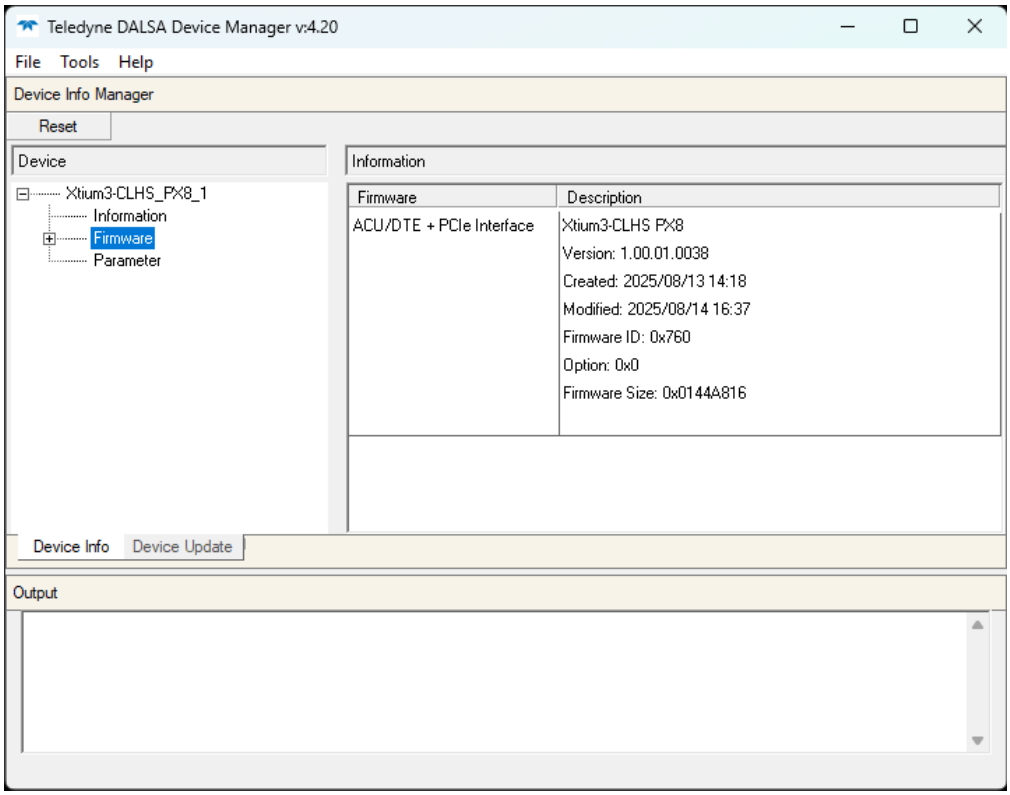


Figure 35: Firmware Information via Device Manager

The firmware itself is composed of what is referred to as the ACU (Acquisition Control Unit), DTE (Data Transfer Engine) and PCIe Interface.

Clicking on the ACU/DTE + PCIe Interface node allows you to manually load a firmware file (.cbf).

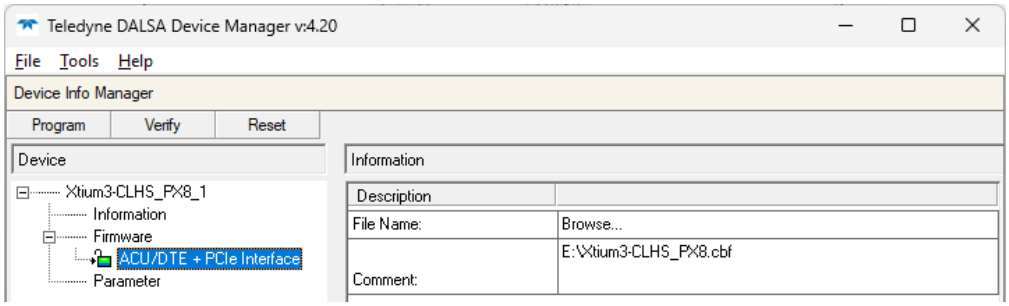


Figure 36: Manually Loading Firmware via Device Manager

Parameter Information

The Parameter panel displays the current value of various settings. In general, these parameters are for trouble shooting and it is not recommended that these be modified.

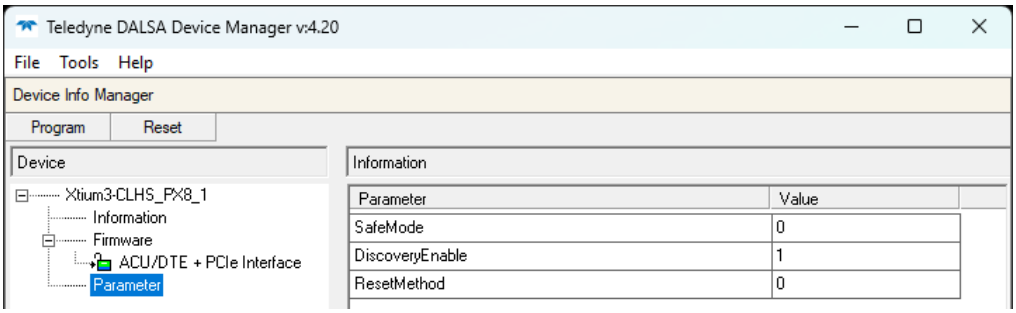


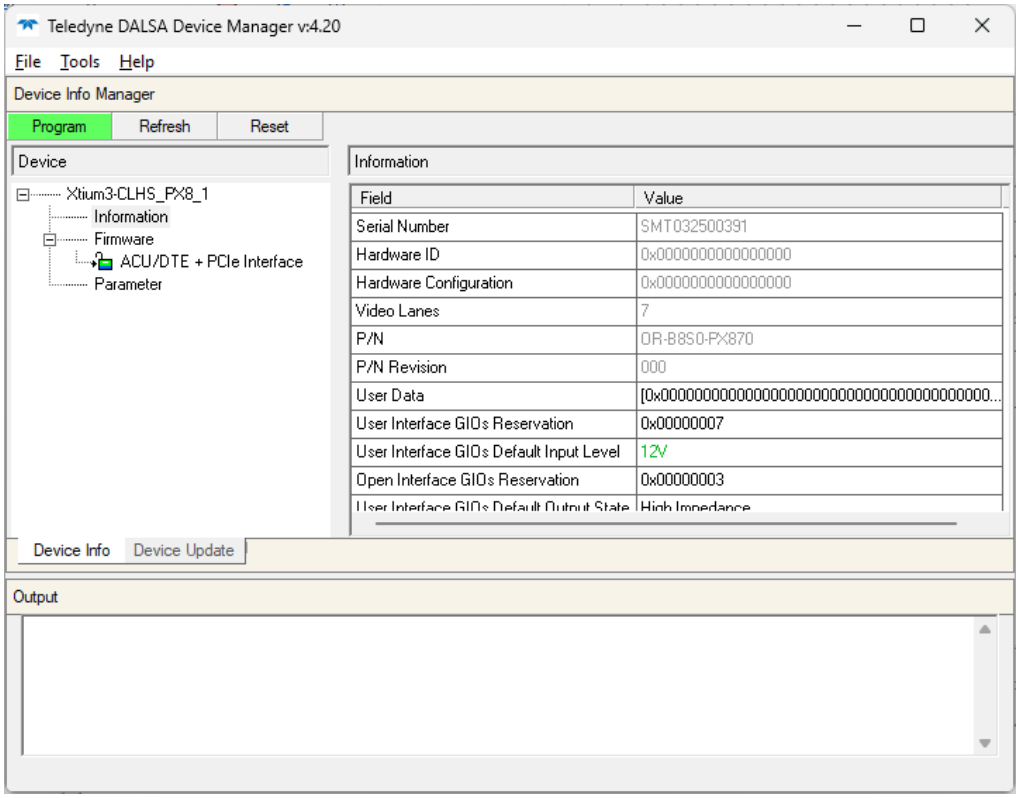
Figure 37: Device Manager Parameter Panel

Parameter	Description
SafeMode	If not zero, the board will not use the normal FPGA and stay in the backup FPGA; only needed if the normal FPGA is corrupted or needs to be upgraded manually. If the board is in SafeMode, the status LED will be blue and cannot be used for grabbing images.
DiscoveryEnable	In general, this parameter should not be modified; it is only for use with specialized custom cameras.
ResetMethod	If zero, the board will be reset by the PCIe bridge, else it will be reset by FPGA command. Needed only on some PCs that cannot use the bridge to reset the board.

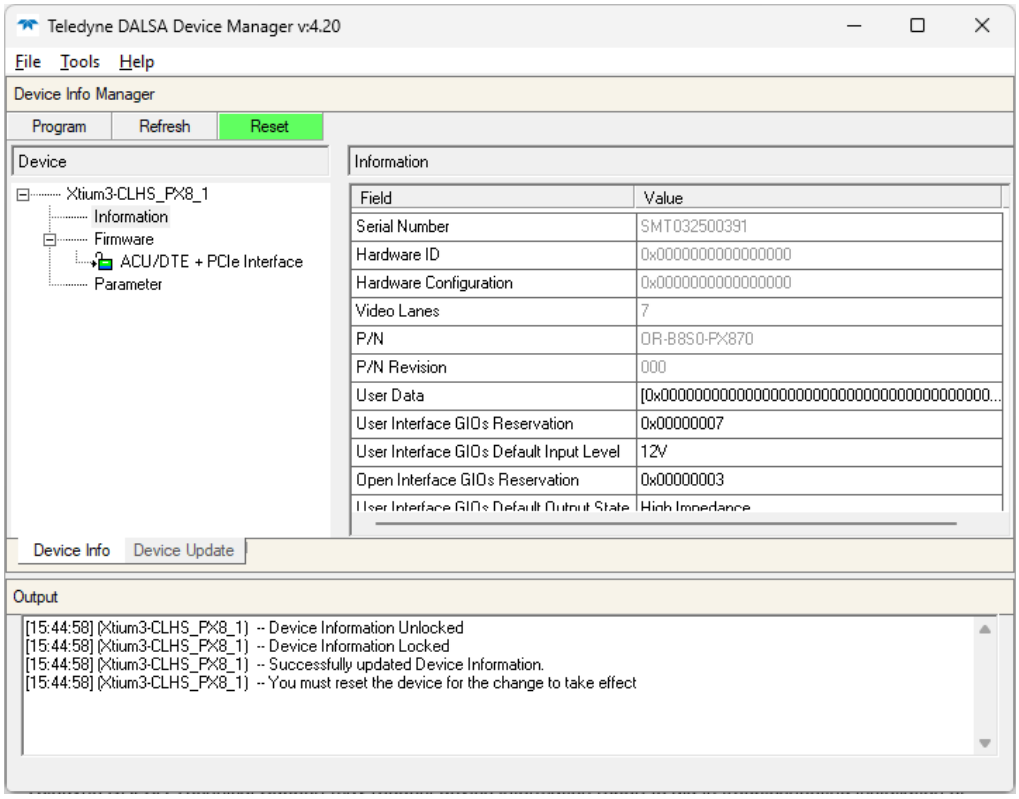
Changing Device Settings

When device parameter settings are changed (shown in **green**), the board must be programmed and reset to effect the changes.

If board programming is required, the Program button is displayed in green; click the button to start programming.

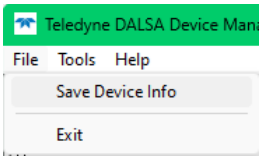


When programming is complete, the Reset button is displayed in green, click the button to reset the board.



Device Information Report

Teledyne Technical Support may request device information report to aid in troubleshooting installation or operational problems. Generate the Xtium3-CLHS PX8 device manager report file (BoardInfo.txt) by clicking **File • Save Device Info**.



Firmware Loader

The Firmware Loader utility provides a simple method to launch the Device Manager to perform a firmware update, in automatic or manual mode. It automatically launches after a driver installation; the Status field indicates if a firmware update is required.

Two firmware update options are availabe:

- Automatic: loads the default firmware available with the current driver.
- Manual: allows the user to load alternative firmware.

Administrator rights are required to update firmware.

Firmware Update: Automatic Mode

Select **Automatic** to update the Xtium3-CLHS PX8 firmware. The Xtium3-CLHS PX8 supports various firmware configurations.

With multiple Xtium3-CLHS PX8 boards in the system, all boards update with new firmware. If any installed Xtium3-CLHS PX8 board installed in a system already has the correct firmware version, an update is not required. In the following screen shot, a single Xtium3-CLHS PX8 board is installed and ready for a firmware upgrade.

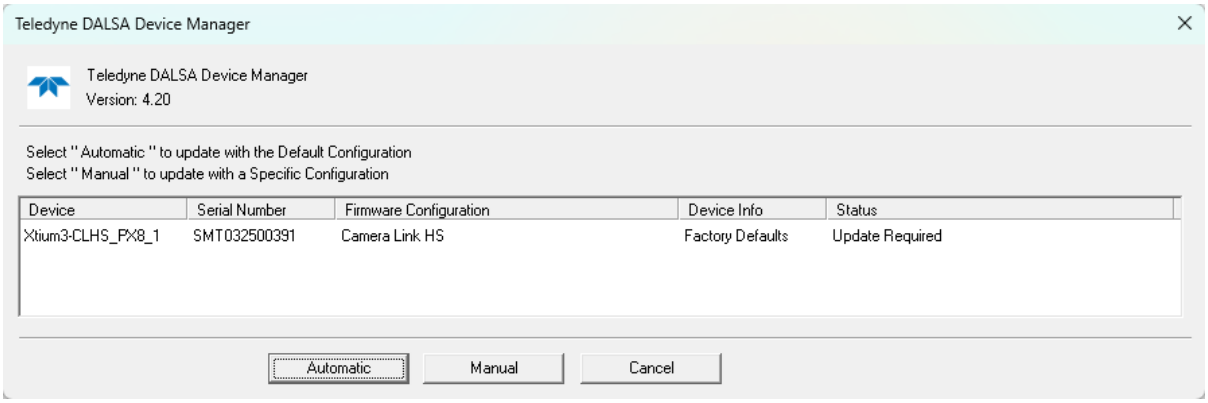


Figure 38: Automatic Firmware Update

Firmware Update: Manual Mode

Select **Manual** mode to load firmware other than the default version or when, in the case of multiple Xtium3-CLHS PX8 boards in the same system, if each requires different firmware.

The following figure shows the Device Manager manual firmware screen. Displayed is information on all installed Xtium3-CLHS PX8 boards, their serial numbers and their firmware components.

Performing a Manual Firmware Update

- Select the Xtium3-CLHS PX8 to update in the **Device** column (if there are multiple boards in the system).
- From the **Configuration** list, select the firmware version required (currently only one firmware is available).
- Click on the **Start Update** button.

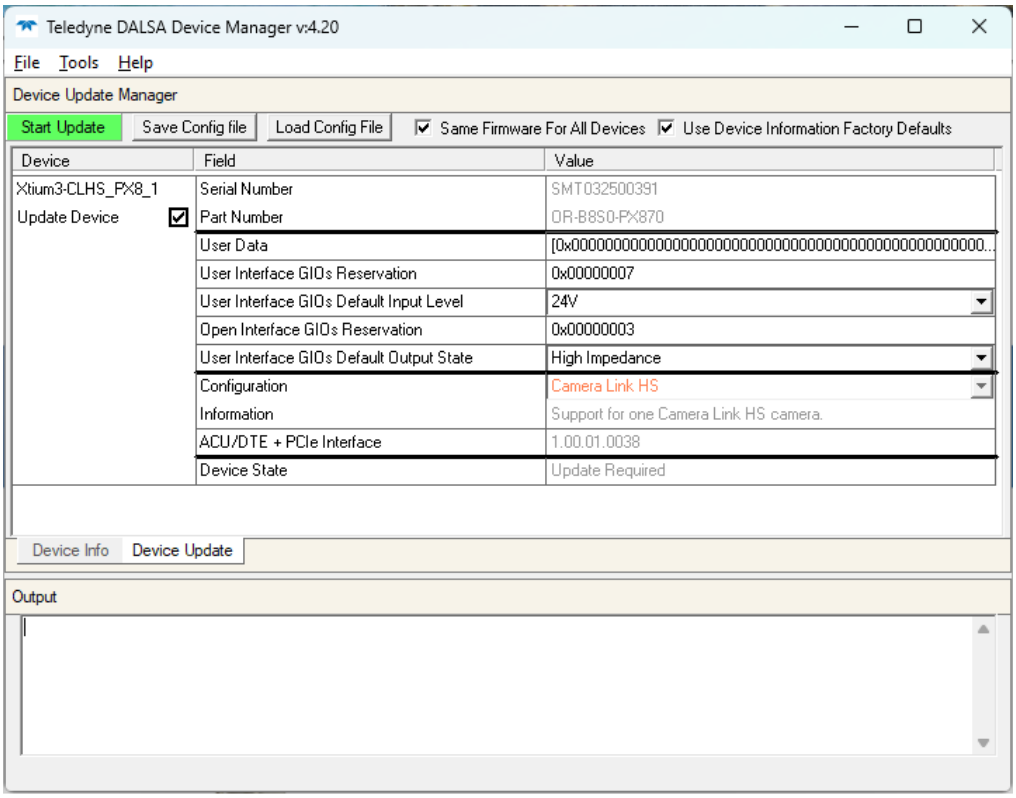


Figure 39: Manual Firmware Update

- The **Output** pane will display the firmware update progress.
- Close the Device manager program when the device reset complete message shows.

Executing the Firmware Loader from the Start Menu

If required, the Xtium3-CLHS PX8 Firmware Loader program is executed via the Windows Start Menu shortcut **Teledyne DALSA • Xtium3-CLHS PX8 Driver • Firmware Update**.

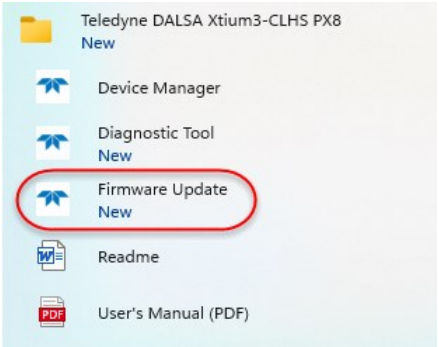


Figure 40: Start Menu Firmware Update Shortcut

A firmware change after installation would be required to select a different configuration mode.

In the rare case of firmware loader errors please see Recovering from a Firmware Update Error.

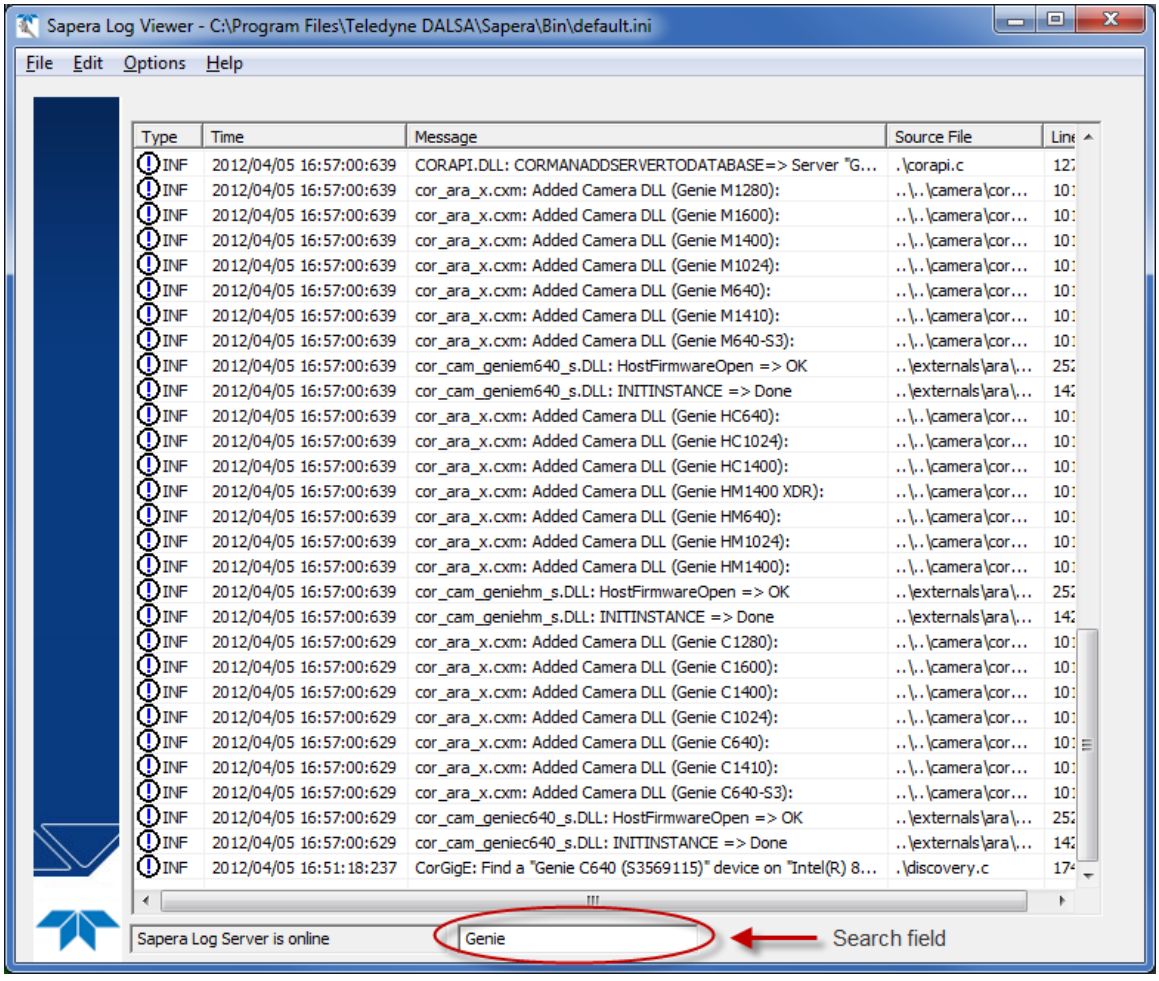
Teledyne Log Viewer

The Teledyne Log Viewer utility program included with the Xtium3-CLHS PX8 installation provides an easy way to view errors and other types of messages generated by Sapera LT applications and Teledyne DALSA hardware, such as camera and frame grabbers. A shortcut to this tool is in the Windows Start menu within the Teledyne DALSA Sapera LT folder.

The Log Viewer provides critical insight into interactions between the host application and Sapera modules. Its detailed message listing offers crucial system-wide information, thus making it an indispensable tool to pinpoint hard to isolate, infrequent errors.

Teledyne Log Viewer runs transparently in the background without impacting the application performance, and stores entire message communications and results. This allows analysis of the log even after the error has occurred. Configuration options allow users to set the type of results to log, such as ignoring info messages and logging only warning or error messages. Messages in the viewer can be dynamically filtered and/or searched for key terms.

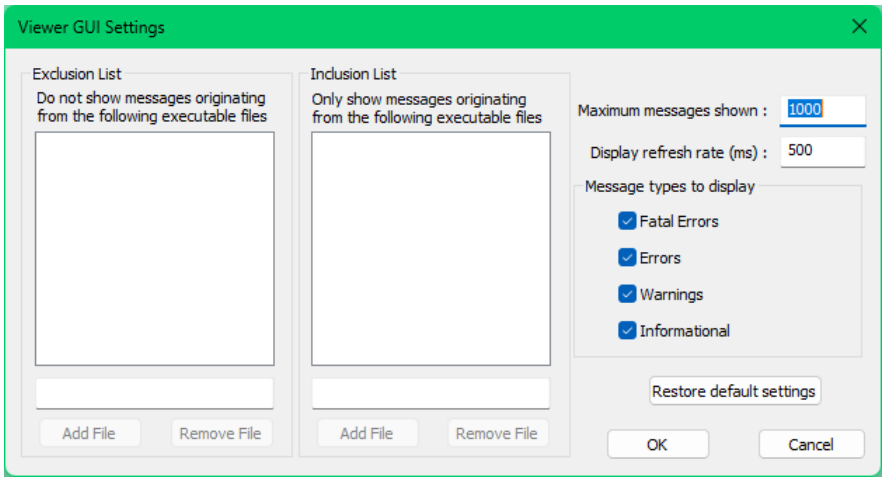
Refer to the utility's online help for more information on using the Log Viewer.



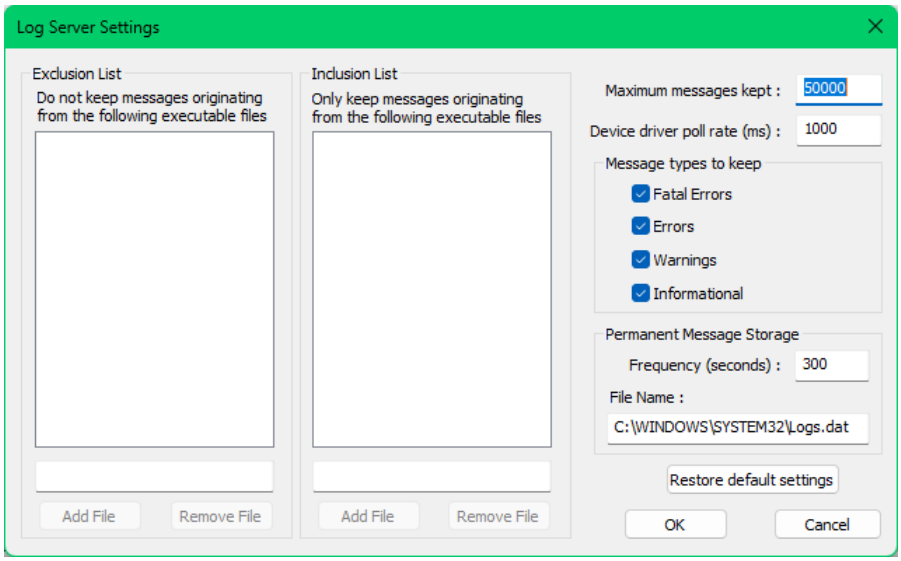
Furthermore, it is possible to run and customize multiple instances of the Log Viewer at the same time; therefore users, when dealing with multiple Teledyne acquisition devices, only view the messages of interest in each instance.

Log Viewer Options

Viewer GUI Options



Log Server Settings



Show Source File Information

The Show Source File Information option enables/disables the display of the Source File column in the Log Viewer window.

Xtium3 Frame Grabber Diagnostic Tool

The Xtium3 Board Diagnostic Tool provides a quick method to see board status and health of Xtium3 family frame grabbers. It also provides live monitoring of FPGA temperature and voltages, which may help in identifying problems.

Diagnostic Tool Main Window

The main window provides a comprehensive view of the installed Xtium3 board. Toolbar buttons execute the board self-test function and open a FPGA live status window.

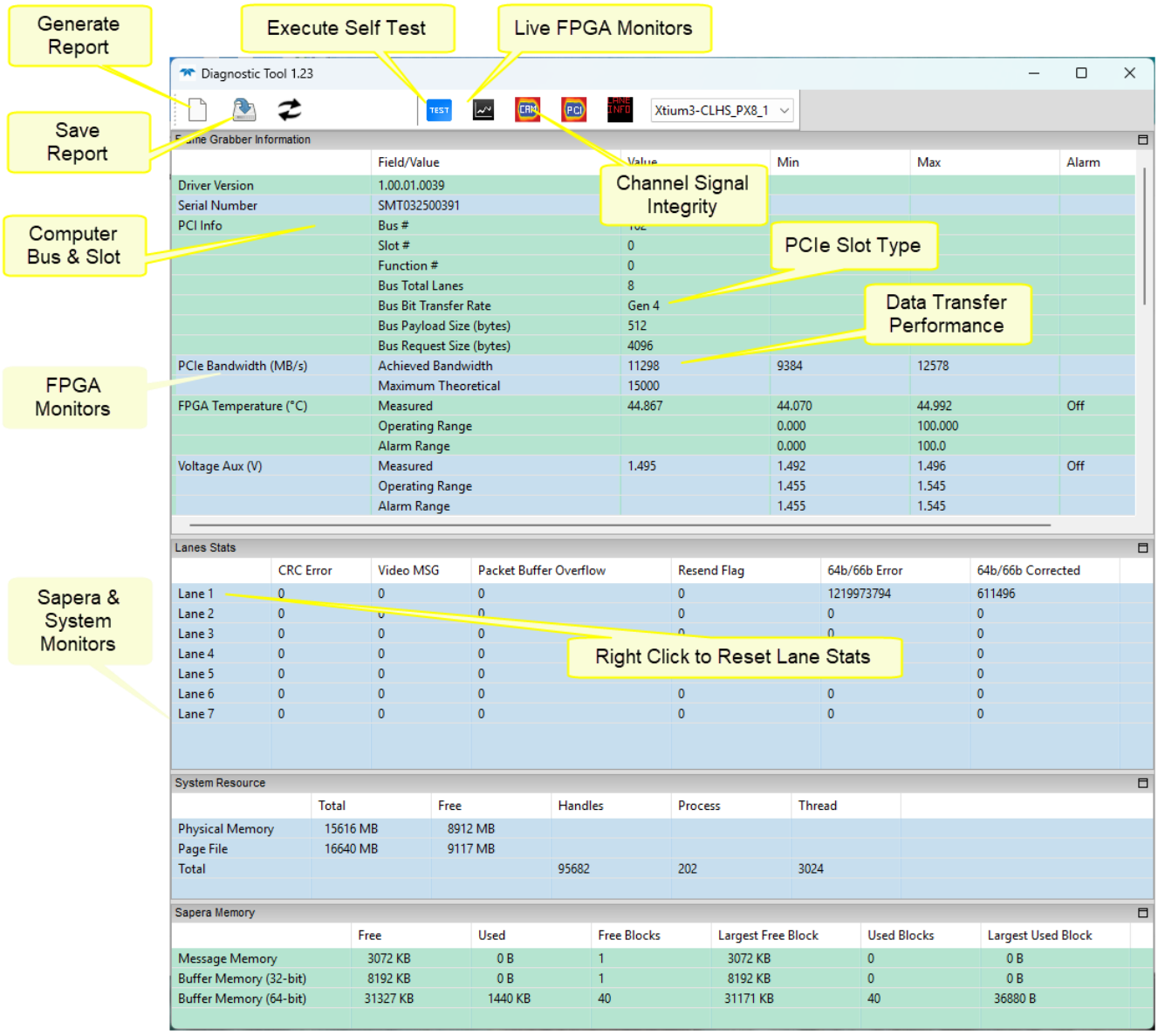


Figure 41: Diagnostic Tool User Interface

One important parameter is the PCI Express bus bit transfer rate supported by the host computer, which defines the maximum data rate possible in the computer.

Another important parameter is the internal Xtium FPGA temperature, which, if excessive, may explain erratic acquisitions due to poor computer ventilation.

Diagnostic Tool

The Xtium3-CLHS PX8 Board Diagnostic Tool provides a quick method to see board status and health. It additionally provides live monitoring of FPGA temperature and voltages, which may help in identifying problems. A shortcut to this tool is in the Windows Start menu within the Teledyne DALSA Xtium3-CLHS PX8 folder.

Do not have any other application running that connects to the Xtium3 such as CamExpert, else the diagnostic window will indicate an error for the PCIe Bandwidth, as shown in the screen capture below.

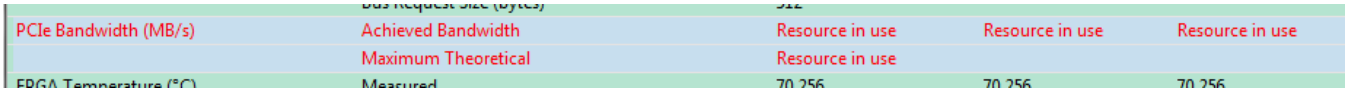


Figure 42: Diagnostic Tool "Resource in use"

The following section describes the information available from the Xtium3 board via the diagnostic program.

Diagnostic Tool Self-Test Window

Click the Start button to initiate the board memory self-test sequence. A healthy board will pass all memory test patterns.

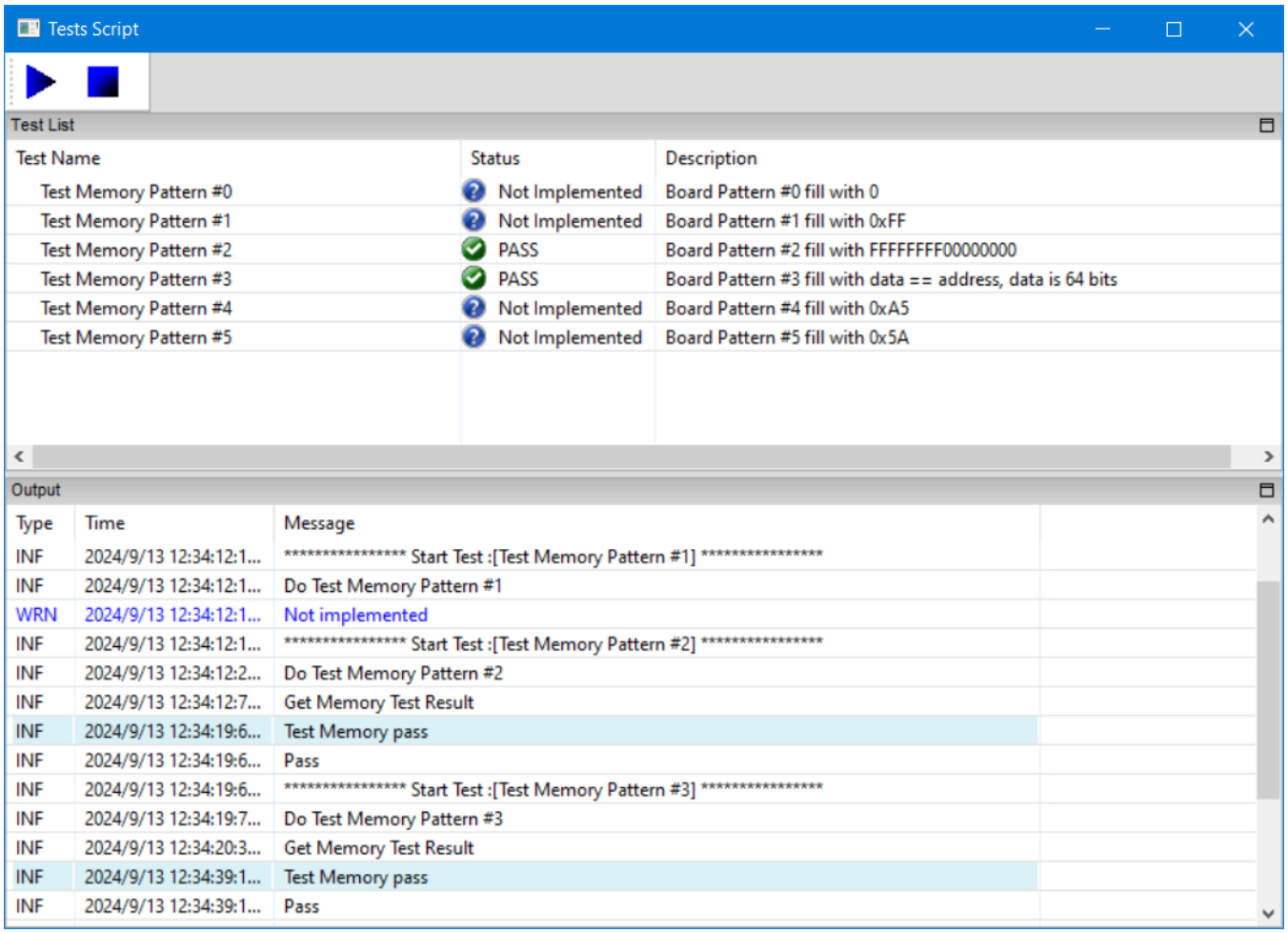


Figure 43: Diagnostic Tool Self-Test Window

Diagnostic Tool Live Monitoring Window

The three FPGA parameters can be monitored in real time. Choosing a parameter puts that graph at the top where the user can select the time unit and time range. Clicking the Output button will open a window displaying any error messages associated with that parameter.

Camera Eye Diagram Monitor

Camera eye diagrams are not available since the Xtium3 does not use SFP+ transceiver modules..

PCI Eye Diagram Monitor

PCI eye diagrams are not available since the Xtium3 does not use SFP+ transceiver modules..

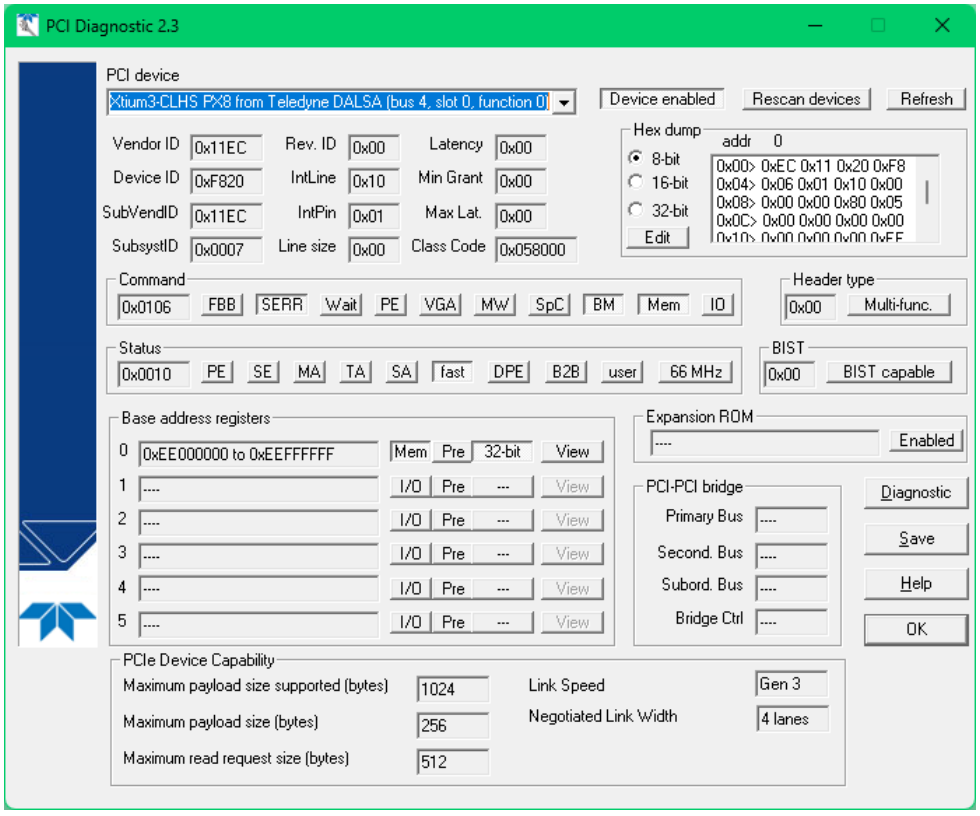
Diagnostic Tool Lane Information

Lane Information is not available since the Xtium3 does not use SFP+ transceiver modules.

.

PCI Diagnostic Tool

The PCI Diagnostic tool is used for debugging frame grabber hardware issues. PCI Diagnostic reads the content of the PCI configuration space and detects memory and I/O conflicts between PCI devices. Use it to verify the integrity of your system before and after installing a new PCI device. Refer to the utility's online help for more information.

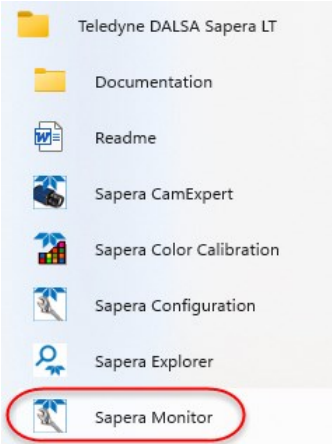


Sapera Monitor

As part of the Trigger-to-Image-Reliability (T2IR) framework, the Sapera Monitor Tool allows users to view the acquisition and transfer events generated by an acquisition device in real time. Sapera Monitor is a standalone application that runs concurrently with CamExpert or with a user application, and can therefore be useful for debugging applications and identifying problems without having to code event handlers.

To open Sapera Monitor

- From the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Monitor**.



Using Sapera Monitor

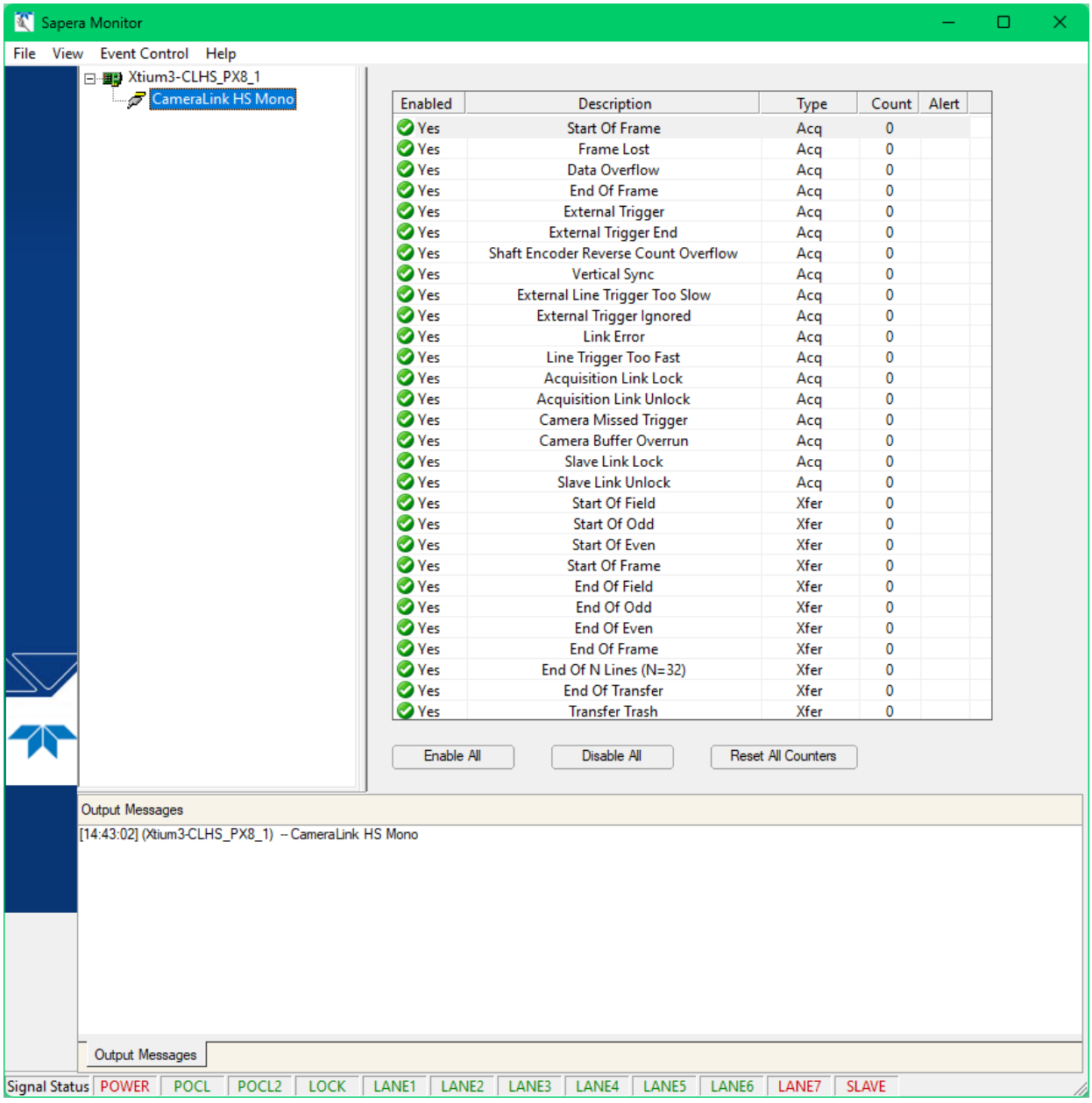
To use the Sapera Monitor tool to monitor a device:

- Open a Sapera application, such as CamExpert, that uses the device.
- Open the Sapera Monitor application.
- In the Sapera Monitor Device directory tree, select a device to monitor.
- In Event table, select the events to monitor.
  - Double-click on an event to toggle between the *Enabled* and *Disabled* state. Alternatively, you can use the **Enable All** and **Disable All** buttons to quickly enable or disable all events at once. Note that these commands do not change the alert and message settings of the events.
  - Use the **Event Control** menu or shortcut menu to modify the settings for an event.
- Start acquisition with the device.

Sapera Monitor Window

The Sapera Monitor Window is divided into three panes:

- **Device directory tree**: displays the available acquisition devices to monitor.
- **Event table**: displays the available events to monitor for the selected device.
- **Output Messages**: displays the messages generated by the selected monitored events.

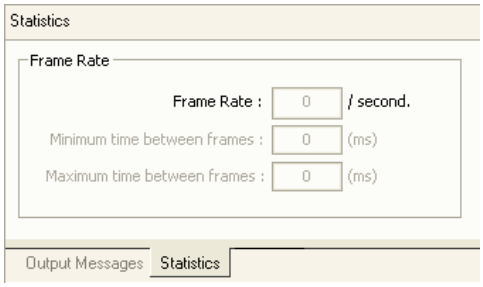


Statistics Tab

Clicking on the **Statistics** tab displays various real-time acquisition statistics, such as the Frame Rate.

NOTE

Different devices can support different statistics and not all devices support all statistics. In addition, these real-time acquisition statistics are not included in generated reports. Thus, **depending on the selected device, the Statistics tab may not be available**.



Sapera Monitor Menu Commands

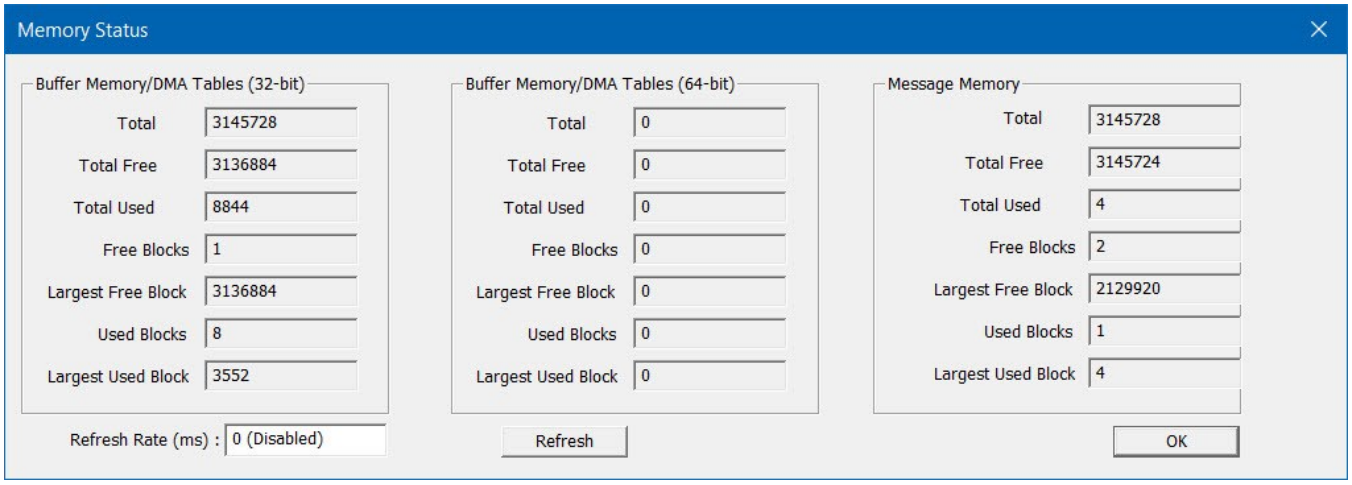
The Sapera Monitor menu provides access to **File**, **View**, and **Event Control** commands.

File Menu Commands

- **Generate Report.** Generates a text file report that includes all event settings and messages included in the current **Output Messages** pane.
- **Clear Log Information.** Clears the current **Output Messages** pane.

View Menu Commands

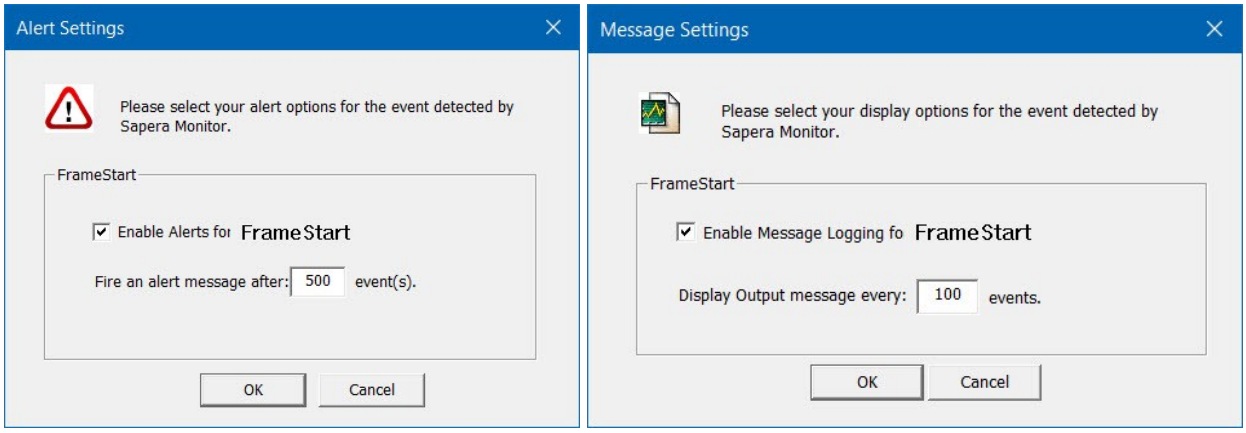
- **Always on Top.** Displays the Sapera Monitor on top of any other windows that may be visible on the desktop.
- **Select Events.** Opens the Sapera Monitor Events Display Settings dialog, which allows you to specify the events to display in the Event table.
- **View Memory Status**. Opens a Memory Status dialog with memory usage information.



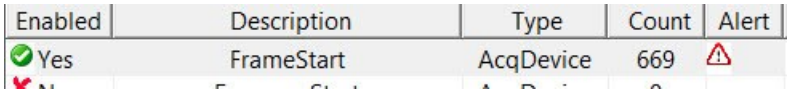
Event Control Menu Commands

Note that the commands are also available by right-clicking on an event.

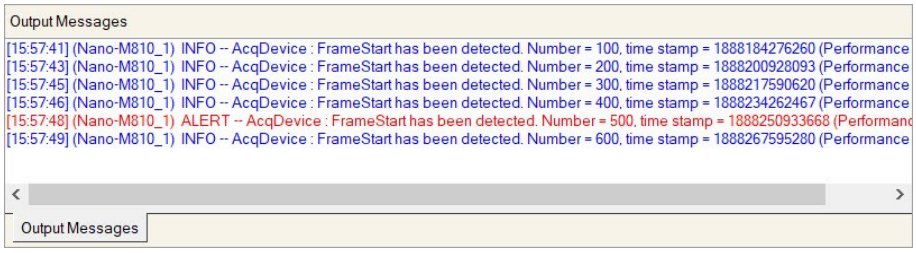
- **Enable** and **Disable.** Starts or stops the currently selected events.
- **Reset event counter.** Returns the event counter to zero for the currently selected.
- **Alert Settings.** Opens the Alert Settings dialog, where you can enable or disable alerts for the selected event and set the number of events required to generate an alert.
- **Message Settings.** Opens the Message Settings dialog, where you can enable message logging and set the number of events required to generate a log message. The log messages appear in the **Output Messages** pane.



When an alert is generated, the Alert icon appears in the Alert column of the event.



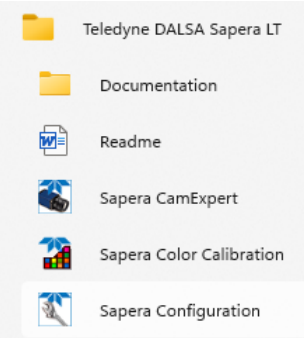
In the **Output Messages** pane, messages appear in blue, while alert messages appear in red.



Sapera Configuration

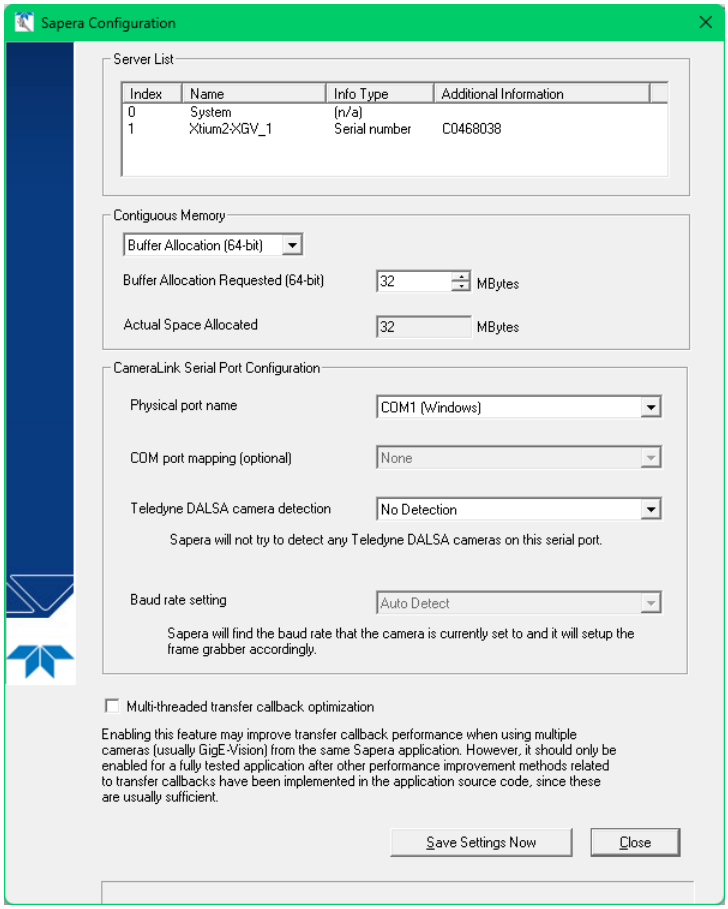
The Sapera Configuration utility allows you to configure Sapera LT resource such as contiguous memory.

It is available from the Windows Start menu under **Teledyne DALSA • Sapera LT • Sapera Configuration**:



Viewing Installed Sapera Servers

The Sapera configuration program 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 **Buffer Allocation Requested** value shows the driver default memory setting while the **Actual Space Allocated** value displays the amount of contiguous memory allocated successfully. A minimum of 32MB is required.

The **Sapera buffers** value determines the total amount of contiguous memory reserved at boot time for the allocation of dynamic resources used for frame buffer management such as scatter-gather list, 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 or when connecting the buffers via a transfer object.

You can approximate the worst-case scenario amount of contiguous memory required as follows:

- Calculate the total amount of host memory used for one frame buffer

  `[number of pixels per line * number of lines * (2 if buffer is 10/12/14 or 16 bits)]`

- Provide 200 bytes per frame buffer for Sapera buffer resources.
- Provide 64 bytes per frame buffer for metadata. Memory for this data is reserved in chunks of 64kB blocks.
- Provide 48 bytes per frame buffer for buffer management. Memory for this data is reserved in chunks of 64kB blocks.
- For each frame buffer DMA table, allocate 24 bytes + 8 bytes for each 4kB of buffer. For example, for a 120x50x8 image:

  `120  * 50 = 6000 = 1.46 4kB blocks -> roundup to 2 4kB blocks`

  Therefore 24 bytes + (2 * 8 bytes) = 40 bytes for DMA tables per frame buffer.
  Memory for this data is reserved in chunks of 64kB blocks.
  If vertical flipping is enabled, one must add 16 bytes per line per buffer. For example, for an image 4080x3072 image:

  `16 bytes * 3072 = 49152 bytes`

NOTE

Sapera LT reserves the 1st 5MB for its own resources, which includes the 200 bytes per frame buffer mentioned above.

Test for any memory error when allocating host buffers.

The following calculation is an example of the amount of contiguous memory to reserve beyond 5MB with 80,000 buffers of 2048x1024x8:
a) (80000 * 64 bytes)
b) (80000 * 48 bytes)
c) (80000 * (24 + (((2048*1024)/4kB) * 8))) = 323MB
d) Total = a (rounded up to nearest 64kB) + b (rounded up to nearest 64kB) + c (rounded up to nearest 64kB).

Host Computer Frame Buffer Memory Limitations

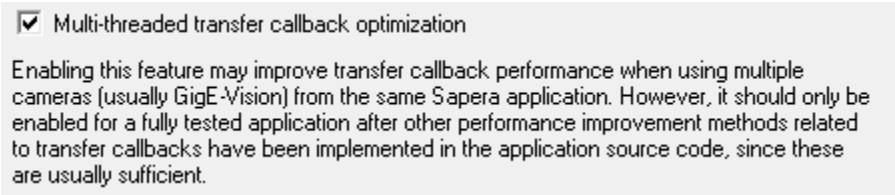
When planning a Sapera application and its host frame buffers used, plus other Sapera memory resources, do not forget the Windows operating system memory needs.

A Sapera application using the preferred *scatter gather buffers* could consume most of the remaining system memory, with a large allocation of frame buffers. If using frame buffers allocated as a *single contiguous memory block*, Windows will limit the allocation dependent on the installed system memory. Use the Buffer menu of the Sapera Grab demo program to allocate host buffer memory until an error message signals the limit allowed by the operating system used.

Contiguous Memory for Sapera Messaging

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

Multi-Threaded Transfer Callback Optimization



Usage Notes When Writing Sapera Applications

- Always disable the **Multi-threaded transfer callback optimization** option (the default) while developing and thoroughly testing the application, especially making sure that appropriate robustness standards are met.
- If the application does not meet performance requirements, all the known performance improvements that can be implemented in application code must be tried (for example, limiting operations as much as possible in the transfer callback function).
- If performance requirements are still not met, and there is only one camera per running instance of the application, then still leave disabled since it provides no performance benefit.
- Only consider enabling if performance requirements are not met with multiple cameras in the same running instance of the application.
- If enabling does not improve performance, then disable it.
- If enabling improves performance, the application must be once again thoroughly tested to prove that it still meets the same robustness requirements as before.

Installing Xtium3-CLHS PX8

Supported Windows Version

Drivers for the Xtium3-CHS PX8 frame grabber series are available for Windows 11 (64-bit).

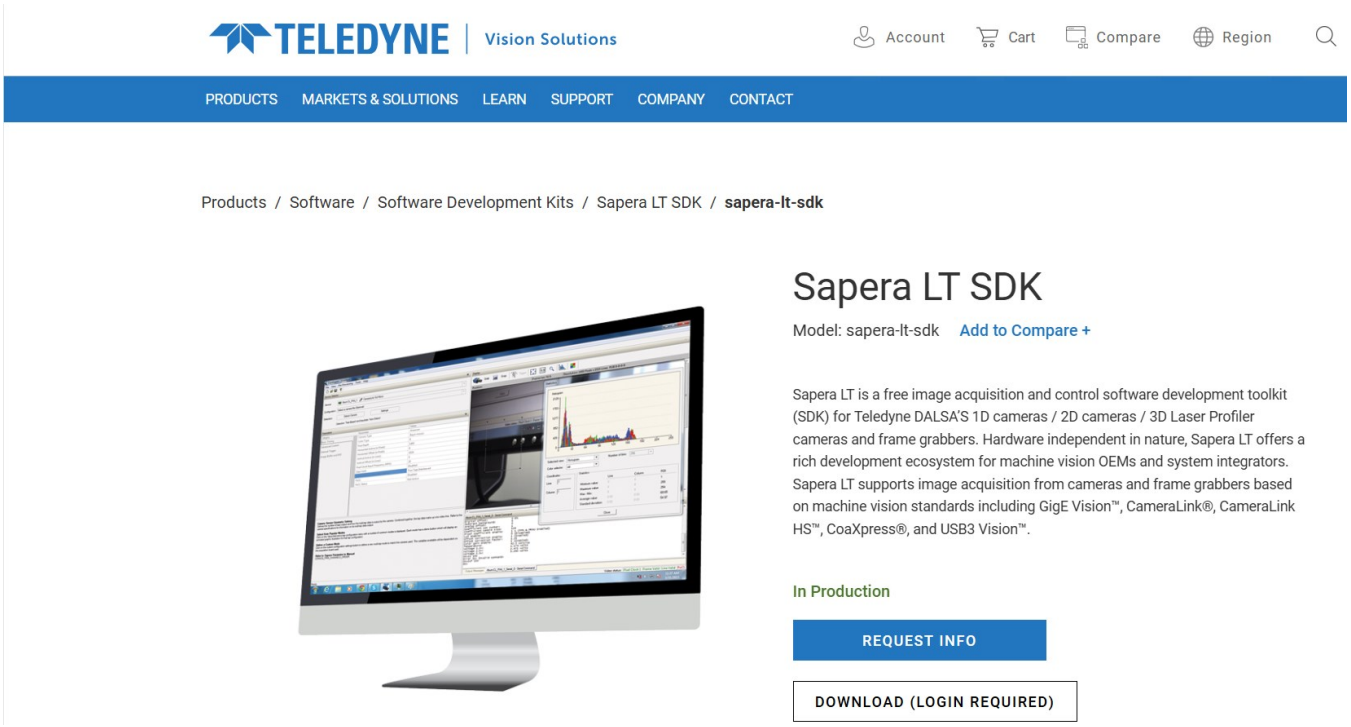
NOTE

Currently you cannot install a Teledyne board driver without Sapera LT installed on the computer. Future releases will provide GenTL support allowing the use of any 3rd party GenTL consumer software.

Software Download

Sapera LT SDK (full version for Windows), the image acquisition and control SDK for Teledyne cameras and frame grabbers is available for download from the Teledyne Visions Solutions website.

For Windows, run-time versions are also available for download at this location.



NOTE

The Sapera LT SDK installation includes compiled demo and example programs, along with project source code, in both C++ and .NET languages, for most Microsoft Visual Studio development platforms. The Sapera LT ++ and Sapera LT .NET demo source code are found in the Sapera\Demos directory.

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

Xtium3-CLHS PX8 Device Drivers

All Teledyne Xtium3-CHLS PX8 series frame grabber drivers are available for download from the Teledyne Visions Solutions website

Secure Boot

Secure Boot is a security feature is part of the UEFI (Unified Extensible Firmware Interface) system built into modern computers that ensures the system boots using only software that is trusted by the manufacturer. It works by verifying the digital signatures of bootloaders and essential system files during startup. When a device is powered on, the firmware checks the signatures of the bootloader and operating system against a set of trusted keys stored in the firmware. If the signatures match, the boot process continues; if not, the system halts, preventing potentially harmful software from executing. This process not only protects the system from rootkits and bootkits but also ensures that the operating system is genuine and has not been tampered with. If any unauthorized or tampered code tries to load, Secure Boot blocks it—protecting your device from rootkits and other low-level malware.

Windows utilizes a set of Microsoft-signed keys to verify the boot process. This means that only Windows-certified operating systems and drivers can be loaded, providing a layer of protection against unauthorized software. Under Windows, Teledyne frame grabber drivers are digitally signed to install and work under Windows platforms with Secure Boot enabled.

The Secure Boot enable state is set in the computer's BIOS. If secure boot is disabled, signed keys are not necessary.

Installation Overview

Administrative rights are required for software installation and device firmware updates.

- Install the board hardware into an available PCI Express x8 Gen4 slot; refer to the Hardware Installation section.
- Turn on the computer.
- Install the Sapera LT SDK. For Windows, a 'runtime library' only version for running applications is available. Sapera LT is installed before Teledyne board drivers.
- Install the Xtium3-CLHS PX8 Sapera board driver.
- Update the board firmware if required.
- Reboot the computer.
- Connect a CLHS camera and test.

See the Quick Start Setup & Installation section for a step-by-step guide to this procedure.

Additional Installation Types:

Other installation types include:

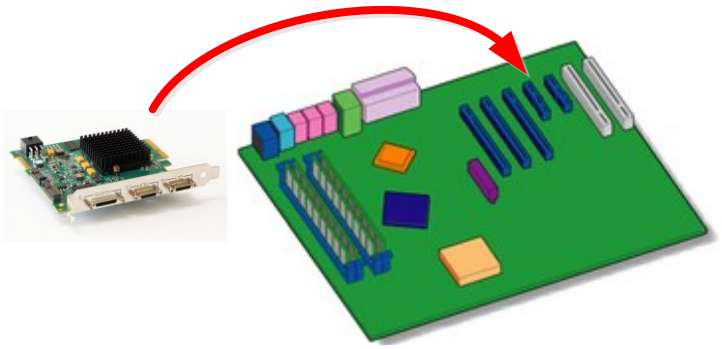
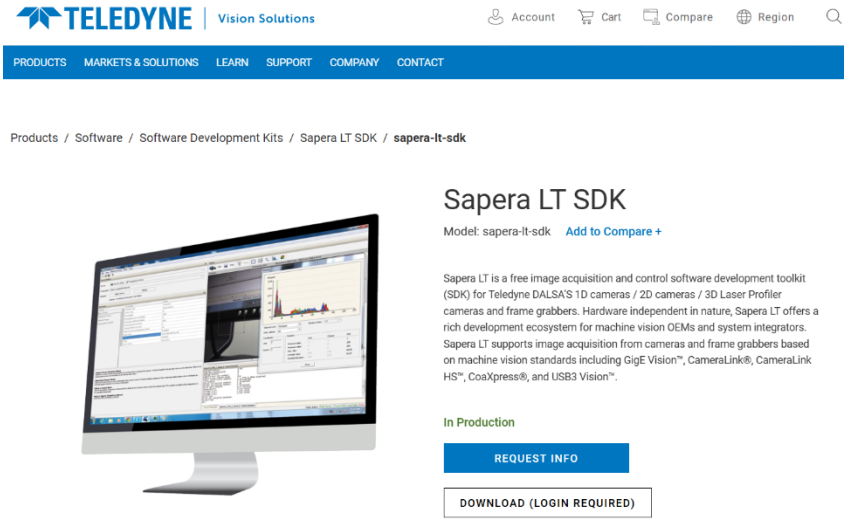
- Upgrading Sapera or Board Driver
- Preserving Board Parameters during Board Replacement
- Silent Installation

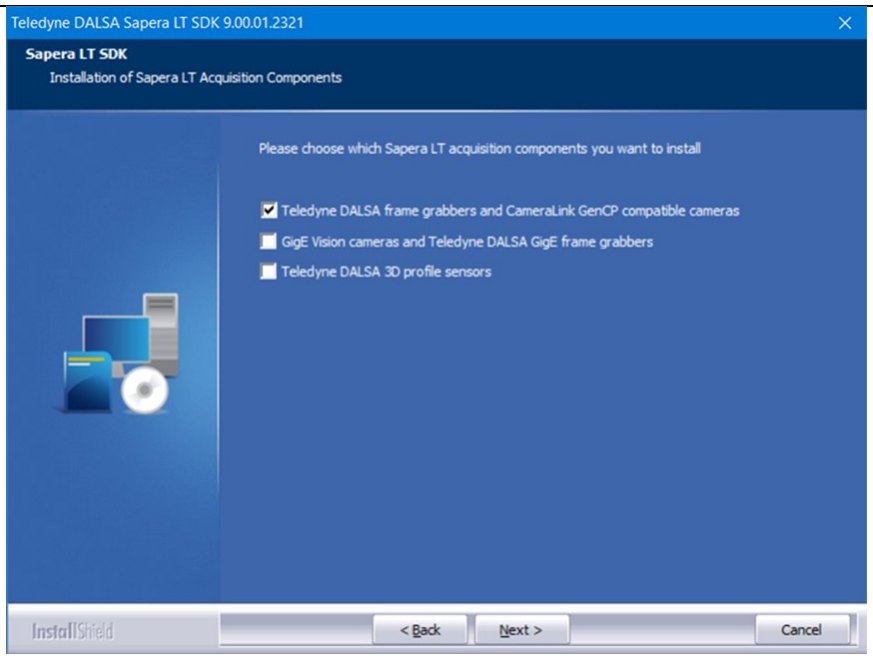
Refer to the Appendix: Additional Installation Types for more information on these installations.

NOTE
For troubleshooting installation problems, refer to Appendix B: Troubleshooting Problems.

Quick Start Setup & Installation

The following procedure outlines the basic steps required to install the Teledyne Xtium3-CLHS PX8. For complete installation details and information, see Installing Xtium3-CLHS PX8.

1	Install the Xtium3-CLHS PX8 in an available PCIe4 x8 slot on the host computer. 
2	Download and install the Sapera LT SDK software from the Teledyne Vision Solutions website.  Launch the installer application executable. Make sure to select the 'Teledyne DALSA frame grabbers and CameraLink GenCP compatible cameras' component.

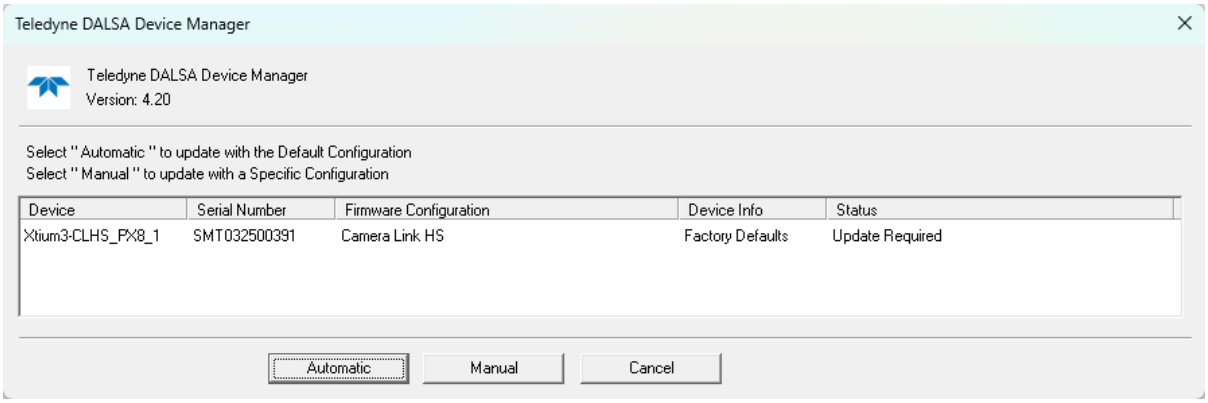


3

Download and install the Xtium3-CLHS PX8 device driver from the [Teledyne Visions Solutions] website.

4

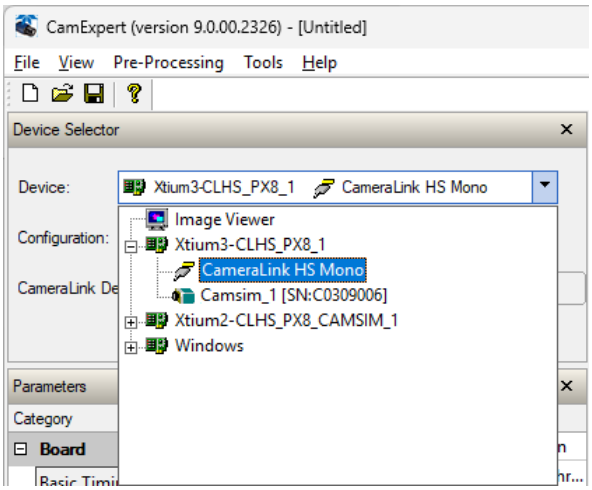
To complete the installation, update the Xtium3-CLHS PX8 firmware when prompted; select Automatic to update the firmware (default configuration) or select Manual to choose an alternate configuration.



Reboot when all software and board drivers are installed.

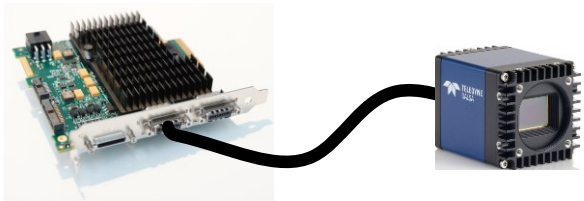
5

Launch Sapera LT CamExpert to verify the installation; the board should be present in the list of available devices.



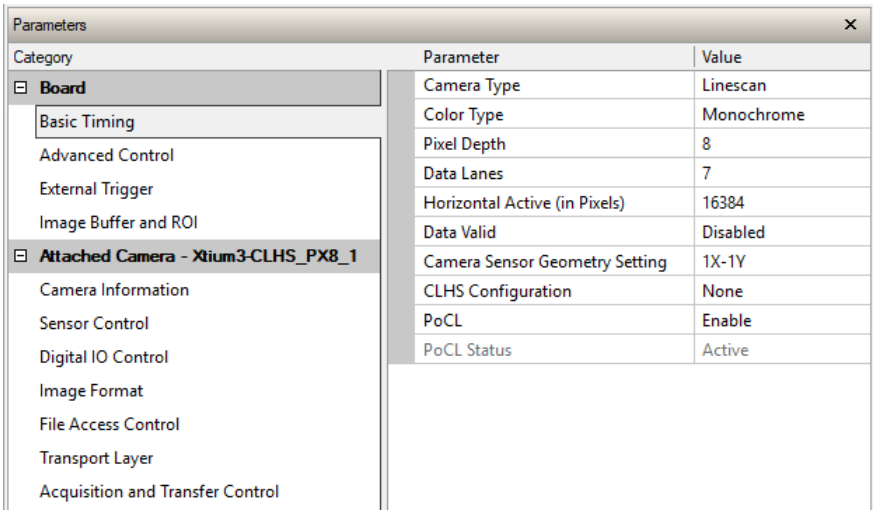
6

Connect camera to the board Camera Link HS input connector. Ensure camera is properly powered.

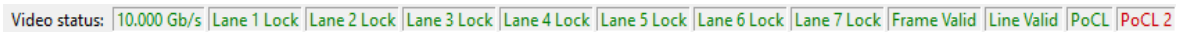


7

When CamExpert detects a camera (as per the CLHS device discovery protocol), camera parameters are displayed along with the board parameters.

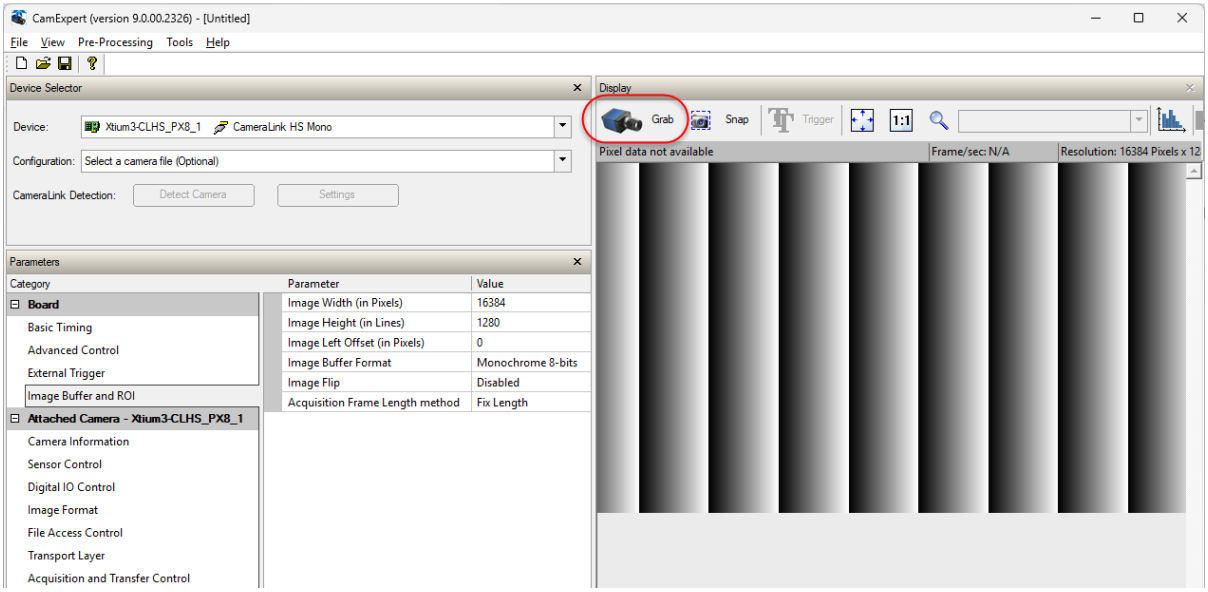


When properly connected, the video status bar displays camera signals in green.



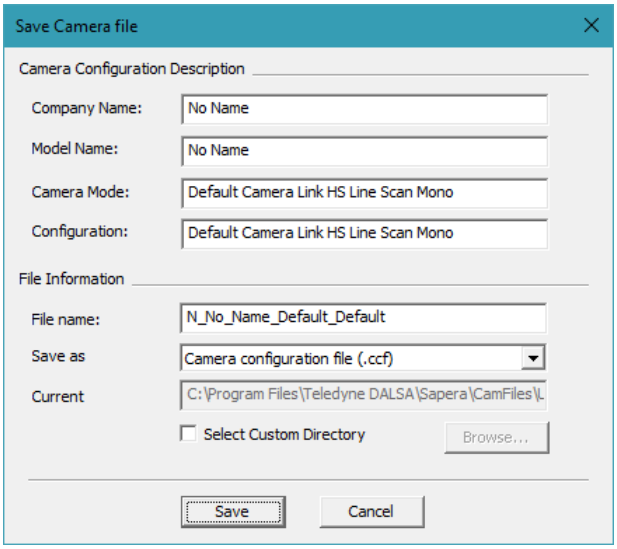
8

Click Grab to acquire a test image to validate the setup.



9

Modify the board and camera parameters as necessary. When completed, save the camera configuration file.



The Xtium3-CLHS PX8 can be configured using the parameter settings in this file when using the Sapera LT API in your application to acquire images

Hardware Installation

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.

WARNING

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.

Hardware Installation Procedure:

- 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 Xtium3-CLHS PX8 into a free PCI Express x8 Gen3 expansion slot. Note that some computer's x16 slot may support boards such as the Xtium3-CLHS PX8, not just display adapters.
- Close the computer chassis and turn the computer on.
- Logon to the computer as administrator or with an account that has administrator privileges.

Multi-board Data Forwarding Hardware Setup

For distributed processing applications, see Multi-board Data Forwarding Hardware Setup for information about data forwarding cabling with multiple Xtium3-CLHS boards.

Multi-board Sync & I/O Setup

- For multi-board sync applications, see Multi-board Sync & I/O Setup for information on using two to four Xtium3-CLHS boards in one computer.

Xtium3-CLHS PX8 Driver Installation on Windows

To install a Xtium3-CLHS PX8 series frame grabber driver (standard, LC or HR), simple launch the executable (for example, Xtium3-CLHS_PX8_1.50.01.0601.exe) and follow the installation prompts.

Refer to the Sapera LT documentation for more information on using the Xitum2-CLHS PX8 series with the Sapera LT API. The Sapera LT installation also includes the Getting Started Manual for Frame Grabbers which provides information specific to using Teledyne frame grabbers with Sapera LT.

Multi-board Data Forwarding Hardware Setup

Distributed processing of high bandwidth image data is easily configured by inter-connecting multiple Xtium3-CLHS boards. The user has full control of the camera data portion transferred, distributed processing task divisions and algorithms.

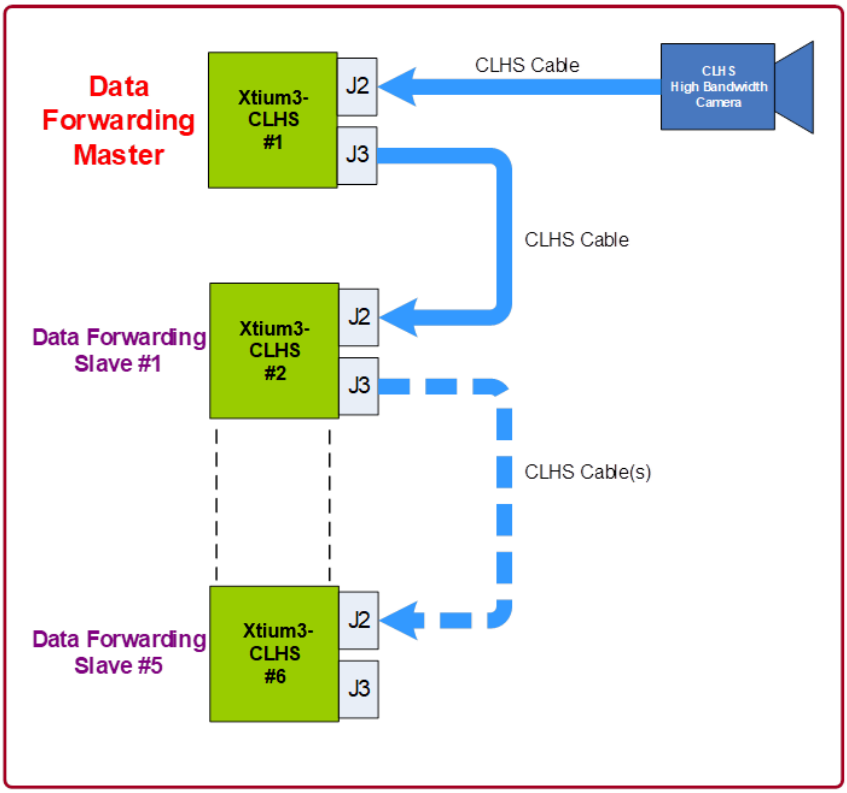


Figure 44: Data Forwarding Block Diagram

- Connect a camera to J3 of the first Xtium3-CLHS board installed in the first PC. This board is defined as the "Data Forwarding Master".
- Using a second Camera Link HS cable (interconnect cables can be up to 15 meters), connect J3 of the Data Forwarding Master Xtium3-CLHS board to J3 of second Xtium3-CLHS installed in a separate computer or the same as the first Xtium3 if it can manage the processing. This second board is defined as the "Data Forwarding Slave #1".

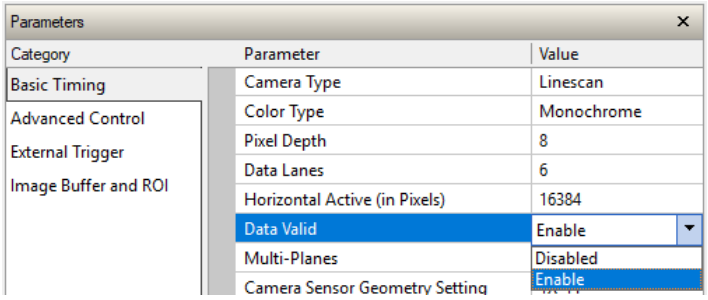
  Optionally, the J3 connector of the Data Forwarding Slave #1 Xtium3-CLHS board can connect to J3 of another Xtium3-CLHS ("Data Forwarding Slave #2").

NOTE

The data forwarding signal output is regenerated on each board without signal loss therefore there is no limit to the number of additional boards that can be connected in this manner.
The Xtium3 CLHS boards can be installed in separate computers or share computers depending on the distributed processing requirements.

- The Xtium3-CLHS driver will automatically detect whether a board is connected to a camera (becoming the Data Forwarding Master) or is connected to forwarded data (becoming a Data Forwarding Slave).
- The data valid signal can be used in order for all frame grabbers to acquire the same triggered frames and/or lines. Once the parameter is enabled on the master frame grabber, a data valid signal is initiated by the master through the camera trigger message. The camera will then pass the data valid control in the image packets.

This parameter is available in CamExpert the board's Basic Timing category:



On slave frame grabbers that also have the data valid enabled, only the video packets that have the data valid control enabled will be acquired. In the case the acquired images are of type variable frame length, the slave frame grabber must select 'Data Valid' as the 'External Frame Trigger Source'

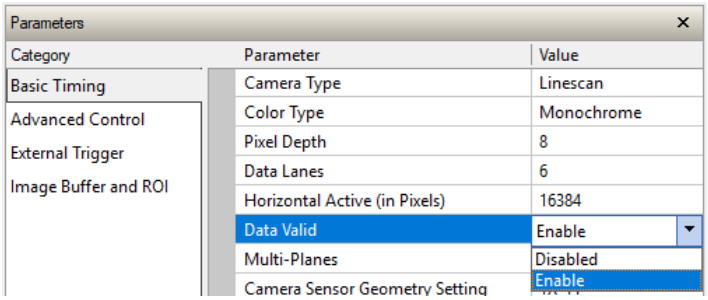
NOTE

During the inactive level of the external frame trigger, there must be at least one line triggered out of the camera in order for the frame grabbers to detect the actual end of the variable frame.

Connecting to Dual Output CLHS Camera

- Connect camera output #1 to J3 of the first Xtium3-CLHS board installed in the first PC. This board is defined as the "Master Frame Grabber".
- Using a second Camera Link HS cable, connect camera output #2 to J3 of a second Xtium3-CLHS board installed in a separate computer or the same as the first Xtium3 if it can manage the processing. This second board is defined as the "Camera Slave Port Frame Grabber".
- Control of the data portion transferred on each camera output is controlled by the camera.
- The data valid signal can be used in order for all frame grabbers to acquire the same triggered frames and/or lines. Once the data valid parameter is enabled on the master frame grabber, a data valid signal is initiated by the master frame grabber through the camera trigger message. The camera will then pass the data valid control in the image packets.

This parameter is available in CamExpert the board's Basic Timing category:



On Camera Slave Port Frame Grabbers (that also have the data valid enabled) only the image packets that have the data valid control enabled will be acquired.

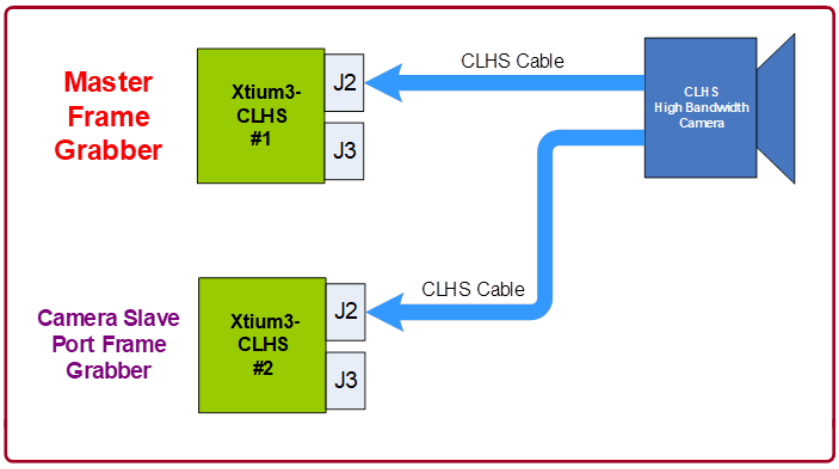
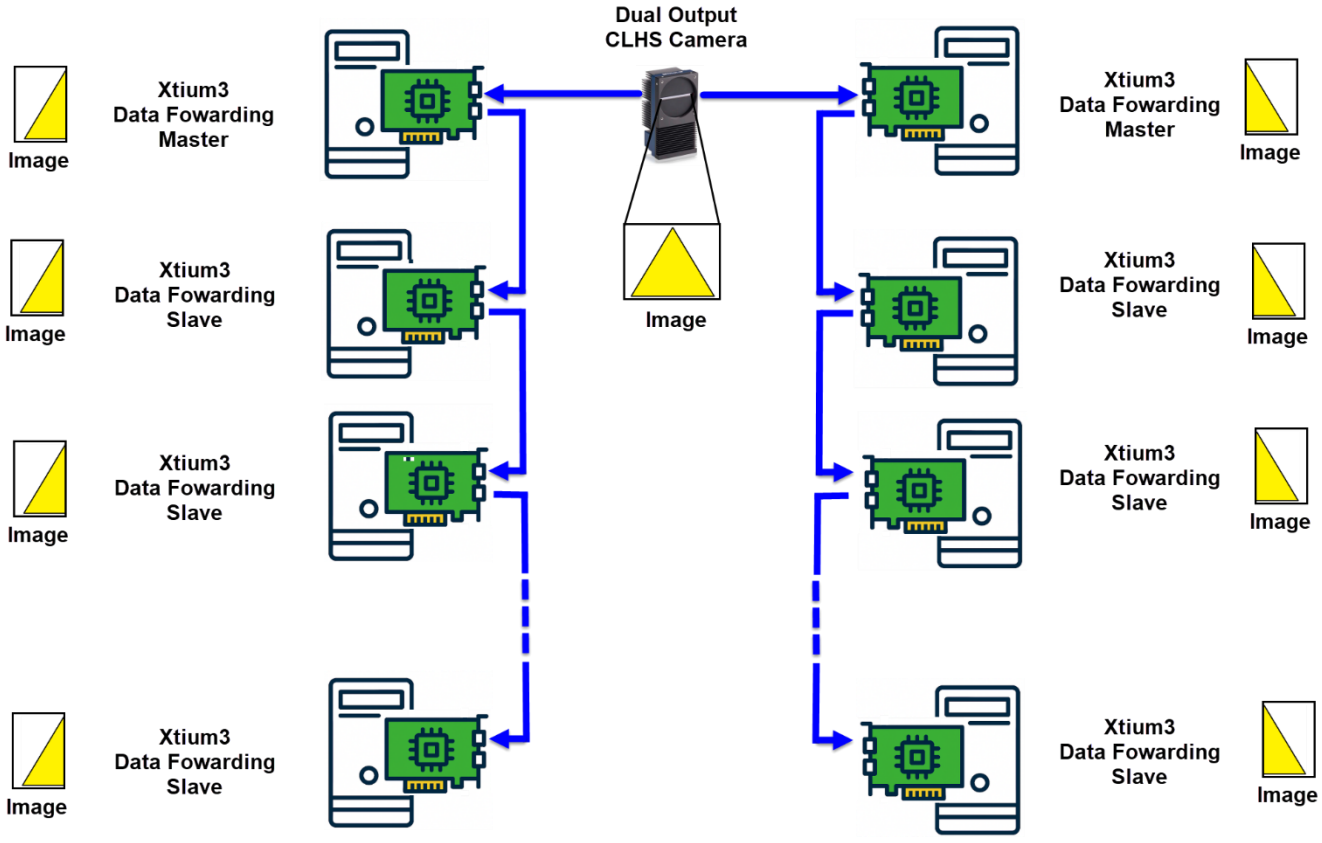


Figure 45: Dual Camera Output Connection Block Diagram

For example, with a dual output CLHS camera, such as the Teledyne Linea HS2, the left and right images are output to separate Xtium3 frame grabbers to maximize bandwidth and acquisition frame rates. Given the large amount of image date (~140 Gbps), this is an ideal case for data forwarding to perform parallel processing.



Related Sapera Parameters

CORACQ_PRM_DATA_VALID_ENABLE

Multi-board Sync & I/O Setup

The multi-board sync feature permits interconnecting multiple Xtium3 boards to synchronize acquisitions to one or two triggers or events. Boards are interconnected using the J5 connector.

The trigger source origin can be either an external signal or a software control signal. The board sending the trigger(s) is the “Sync Master” board, while the one or more boards receiving the control signal(s) are “Sync Slaves”.

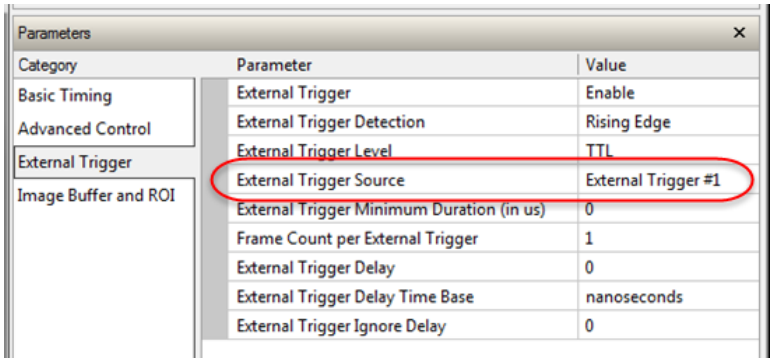
There are 8 bi-directional General I/Os that can be interconnected between multiple boards. These bi-directional I/Os can be read/written by Sapera applications.

Bi-directional General I/Os 1, 2 and 3 can also act as the multi-board sync I/Os. By default, Output 1 and 2 are reserved for the Board Sync; if necessary, these settings can be modified using the Device Manager using the Open Interface GIOs Reservation dialog.

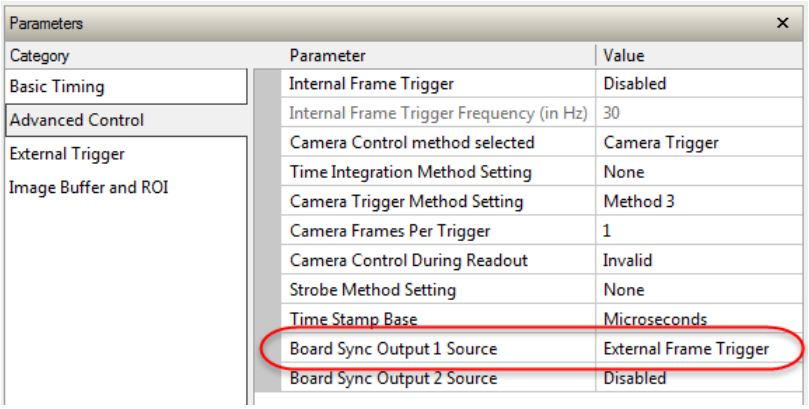
Setup of the boards is done either by setting parameters via a Sapera application or by using CamExpert to configure two camera files (.ccf). For testing purposes, two instances of CamExpert (one for each board) can be run on the system where the frame grabbers are installed.

Configuration via Sapera CamExpert

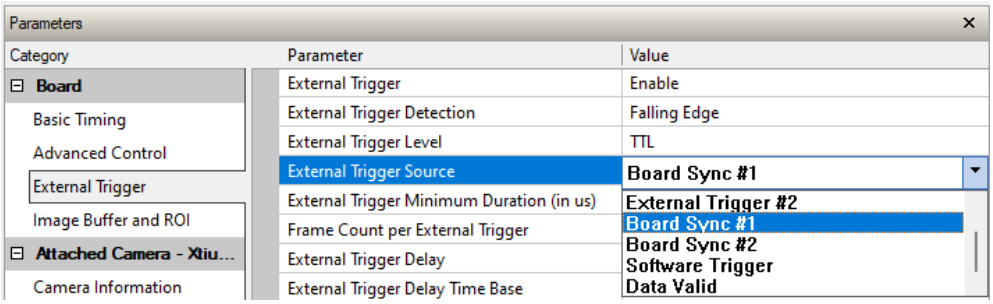
Start the first instance of CamExpert and select one installed **Xtium3 board** to be the **sync master**. This board is configured to use an external frame trigger on input #1 by setting the External Trigger Source parameter (available in the External Trigger category) to External Trigger #1.



The **Sync Master Xtium3 board** is also configured to output the external frame trigger on board sync #1, by setting the Board Sync Output 1 Source parameter (in the Advanced Control category) to External Frame Trigger.



The **Sync Slave Xtium3 board** External Trigger Source is configured to receive its trigger from Board Sync #1 or Board Sync #2.



Test Setup: Start the acquisition on all slave boards. The acquisition process is now waiting for the control signal from the master board. Trigger master board acquisition and the acquisition start signal is sent to each slave board.

Related Sapera Parameters

CORACQ_PRM_BOARD_SYNC_OUTPUT1_SOURCE, CORACQ_PRM_BOARD_SYNC_OUTPUT2_SOURCE, CORACQ_PRM_EXT_TRIGGER_SOURCECORACQ_PRM_EXT_LINE_TRIGGER_SOURCE

Cables & Accessories

The following cables and accessories are available for purchase via third party vendors or Teledyne sales.

Camera Link HS Cables Overview and Resources

In general, a CX4 Active optical compliant cable assembly is mandatory for use as a CLHS interface cable with a thumbscrew type junction shell.

NOTE
These cables are unidirectional and end cable labelled FG must be connected to the Xtium3.

Visit our web site for additional information on the CLHS interface:

https://www.teledynevisionsolutions.com/learn/learning-center/machine-vision/camera-link-hs/

For additional information on cables and their specifications, visit the following web sites and search for “Camera Link HS” cables:

Components Express	http://www.componentsexpress.com/
Alysium-Tech	https://www.alysium.com/

DH40-27S Cable to Blunt End (OR-YXCC-27BE2M1, Rev B1)

Cable assembly consists of a 2000 mm (~6 ft.) blunt end cable to mate to Xtium3 external connector **J1**.

NOTE

The applicable wiring color code table is included with the printed Product Notice shipped with the cable package — no other wiring table should be used.

Important: Cable part number OR-YXCC-27BE2M0 rev.3 is obsolete and should not be used with any Xtium3 series boards.

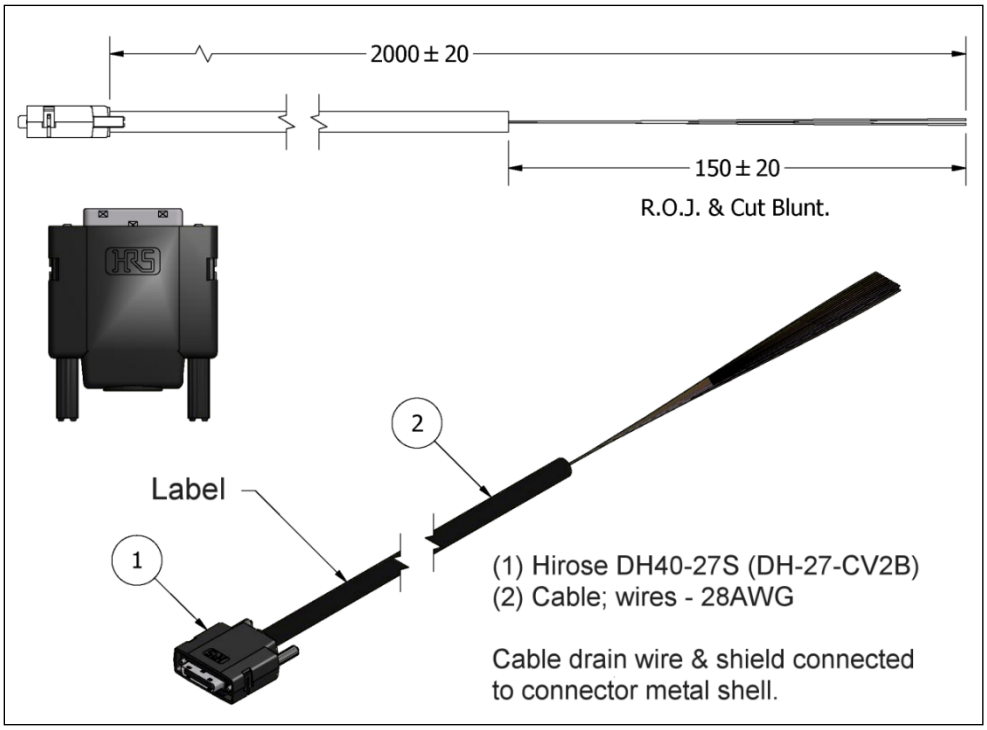
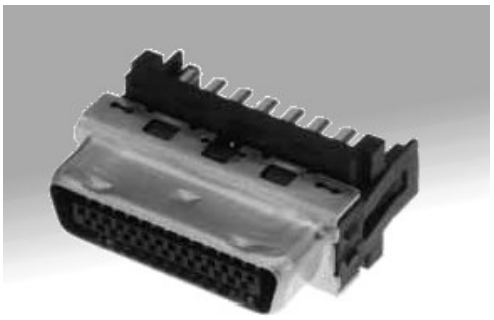


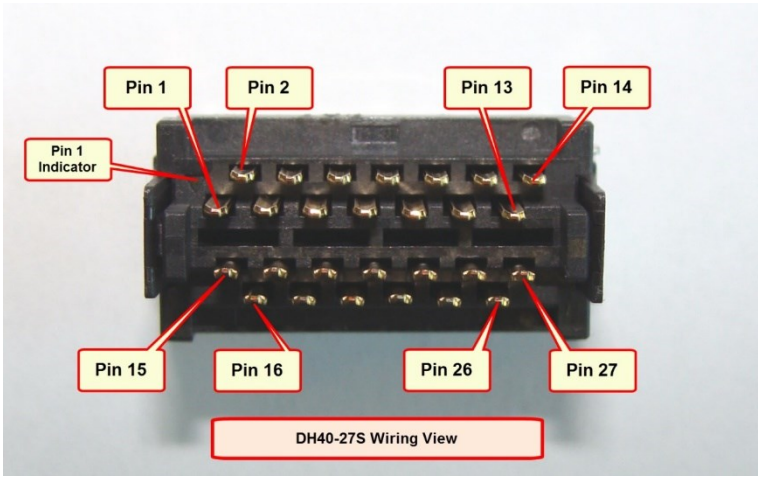
Figure 46: DH60-27P Cable No. OR-YXCC-27BE2M1 Detail

Figure 47: Photo of cable OR-YXCC-27BE2M1

DH40-27S Connector Kit for Custom Wiring

Teledyne makes available a kit comprised of the DH40-27S connector plus a screw lock housing package, for clients interested in assembling their own custom I/O cable. Order part number “OR-YXCC-H270000”, (package as shown below).





Board Sync Cable Assembly

Board sync cables are available to connect Xtium3 boards for the board sync function as described in section J5: Multi-Board Sync / Bi-directional General I/Os.

Interconnect up to 6 Xtium3 boards via their J5 connector using:

- OR-YXCC-BSYNC20 cable for 2 boards
- OR-YXCC-BSYNC40 cable for 3 to 4 boards
- OR-YXCC-BSYNC60 cable for 5 to 6 boards

For a third part source of cables, see https://www.samtec.com/products/ffsd.

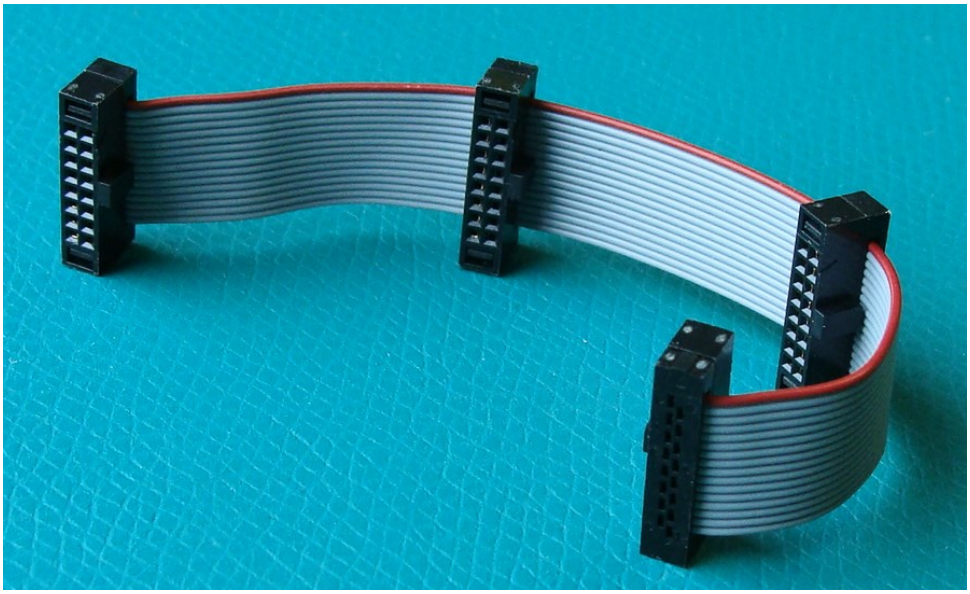


Figure 48: Photo of cable OR-YXCC-BSYNC40

Appendix A: Additional Installation Types

Upgrading Sapera or Board Driver

When installing a new version of Sapera or a Teledyne acquisition board driver in a computer with a previous installation, the current version **must** be un-installed first. Described below are two upgrade situations.

If the board is installed in a different slot, the new hardware wizard opens.

Board Driver Upgrade Only

Minor upgrades to acquisition board drivers are distributed as ZIP files available in the Teledyne Visions Solutions web site.

These minor upgrades typically 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 file before installing, for information on the minimum Sapera version required.
- If the ReadMe file does not specify the Sapera version required, contact Teledyne Technical Support (see Contact Information).

To upgrade the board driver only

1. Logon the computer as an administrator or with an account that has administrator privileges.
2. In the taskbar search box, type *Control Panel* and open it.
3. Select **Programs and Features**.
4. Select Teledyne DALSA Xtium3-CLHS PX8 Device Driver and click **Uninstall**.
5. Install the new board driver. Run **Setup.exe** if installing manually from a downloaded driver file.

NOTE

You cannot install a Teledyne board driver without Sapera LT installed on the computer.

Upgrading both Sapera and Board Driver

When upgrading both Sapera and the acquisition board driver, follow the procedure described below.

To upgrade the board driver and Sapera LT

1. Logon the computer as an administrator or with an account that has administrator privileges.
2. In the taskbar search box, type *Control Panel* and open it.
3. Select **Programs and Features**.
4. Select Teledyne DALSA Xtium3-CLHS PX8 Device Driver and click **Uninstall**.
5. Select Teledyne DALSA Sapera LT and click **Uninstall**.
6. Restart the computer and logon the computer as an administrator again.
7. Install the new versions of Sapera LT and the board driver as in a first-time installation. See Sapera LT Library & Xtium3-CLHS PX8 Driver Installation for installation procedures.

Preserving Board Parameters during Driver Upgrade

User defined parameter settings for previously installed boards can be preserved when upgrading a device driver by using an *install.ini* file as described in the Custom Driver Installation using install.ini section. Clicking **Automatic** on the Device Manager Start-up dialog will apply the settings specified in the *install.ini* file.

To verify the settings specified in the *install.ini* file, click **Manual**; differences between the current device settings are shown in **green** in both the Device Info and Device Update tabs.

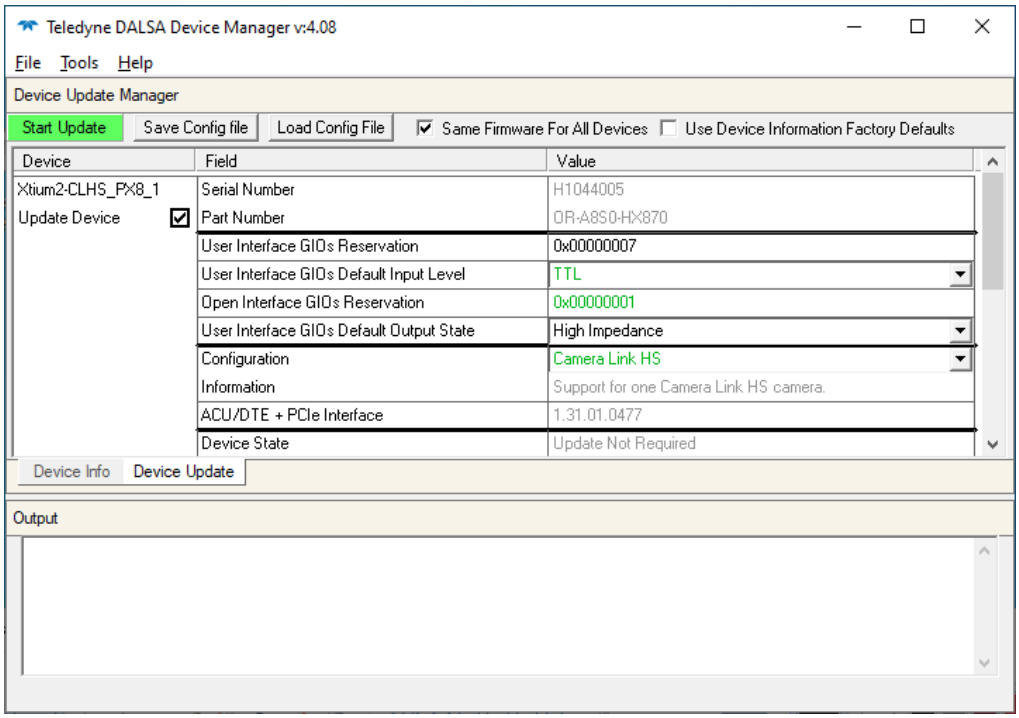


Figure 49: Device Manager Parameter Setting Differences

Upgrading without an *install.ini* file requires selecting **Manual** update on the Device Manager Start-up dialog and setting the required parameters manually.

NOTE

Without an *install.ini, c*onfiguration information is not preserved and is always set to factory default.

Preserving Board Parameters during Board Replacement or System Cloning

When replacing a board in a system or cloning a system configuration using a hard drive image, if the previous device parameter settings differ from the factory default driver settings it is indicated as “User Defined” or “Manual Configuration” in the Teledyne DALSA Device Manager start-up dialog under the Device Info column. User-defined settings are specific to the PCI Express slot on the system.

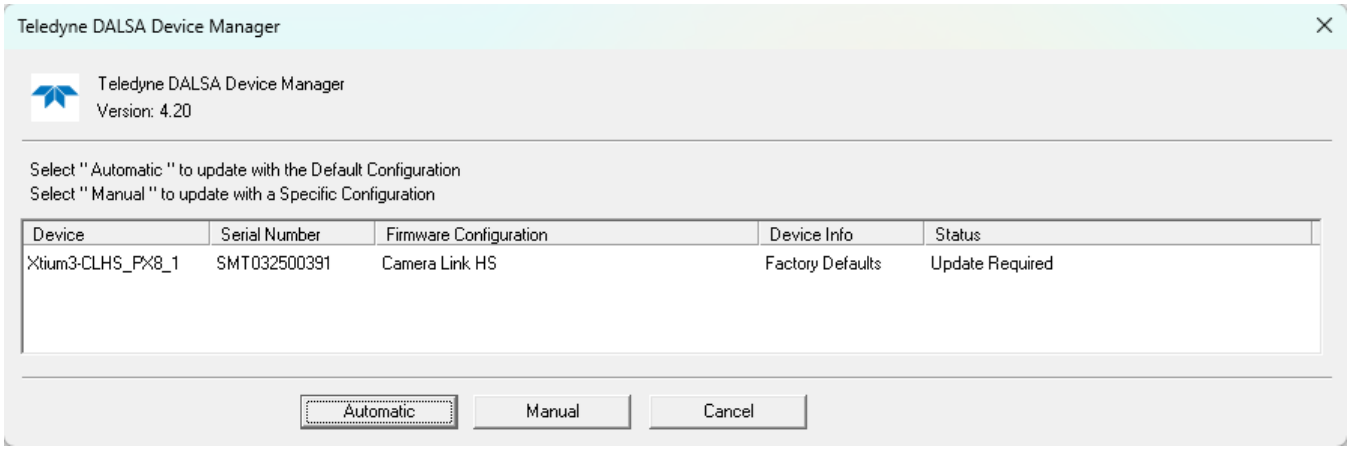


Figure 50: Firmware Update Status

To preserve the user defined parameter settings, select “Manual” and proceed with the update; differences between the current settings are shown in green in both the Device Info and Device Update tabs.

For systems with mulitple boards, if boards use different firmware configurations, disable the **Same Firmware For All Devices** option (otherwise the configuration specified for the first board according to slot position is applied to all boards in the system).

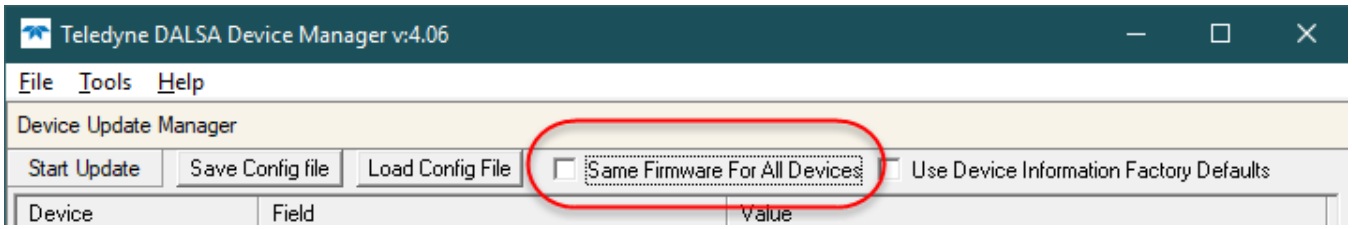


Figure 51: Same Firmware For All Devices Checkbox

Silent Installation

Both Sapera LT and the Xtium3-CLHS PX8 driver installations share the same installer technology. When the installations of Teledyne products are embedded within a third party's product installation, the mode can either have user interaction or be completely silent.

A Silent Mode installation is recommended when integrating Teledyne products into your software installation. The silent installation mode allows the device driver installation to proceed without the need for mouse clicks or other input from a user.

NOTE

You must reboot after the installation of Sapera LT. However, to streamline the installation process, Sapera LT can be installed without rebooting before installing the board hardware device drivers. The installations then complete with a single final system reboot.

Perform Teledyne embedded installations in either of these two ways:

- **Normal Mode**
  The default mode is interactive. This is identical to running the setup.exe program manually from Windows (either run from Windows Explorer or the Windows command line).
- **Silent Mode**
  This mode requires no user interaction. A preconfigured "response" file provides the user input. The installer displays nothing.

Preparing a Silent Mode Installation requires two steps:

- Prepare the response file, which emulates a user.
- Invoke the device driver installer with command options to use the prepared response file.

Creating a Response File

Create the installer response file by performing a device driver installation with a command line switch "-r". The response file is automatically named setup.iss and is saved in the \windows folder. If a specific directory is desired, the switch –f1 is used.

As an example, to save a response file in the same directory as the installation executable of the Xtium3-CLHS PX8, the command line would be:

```
Xtium3-CLHS_PX8_1.00.00.0000 -r -f1".\setup.iss"
```

Running a Silent Mode Installation

A device driver silent installation, whether done alone or within a larger software installation requires the device driver executable and the generated response file **`setup.iss`**.

Execute the device driver installer with the following command line:

```
Xtium3-CLHS_PX8_1.00.00.0000 -s -f1".\setup.iss"
```

Where the **–s** switch specifies the silent mode and the **–f1** switch specifies the location of the response file. In this example, the switch –f1".\setup.iss" specifies that the **`setup.iss`** file be in the same folder as the device driver installer.

NOTE

The Windows Security dialog box will appear unless one has already notified Windows to 'Always trust software from "Teledyne DALSA Inc." during a previous installation of a driver.

Silent Mode Uninstall

Similar to a silent installation, a response file must be prepared first as follows.

Creating a Response File

The installer response file is created by performing a device driver un-installation with a command line switch "-r". The response file is automatically named setup_uninstall.iss which is saved in the \windows folder. If a specific directory is desired, the switch "–f1" is used.

As an example, to save a response file in the same directory as the installation executable of the Xtium3-CLHS PX8, the command line would be:

```
Xtium3-CLHS_PX8_1.00.00.0000 -r -f1".\setup_uninstall.iss"
```

Running a Silent Mode Uninstall

Similar to the device driver silent mode installation, the un-installation requires the device driver executable and the generated response file setup.iss.

Execute the device driver installer with the following command line:

```
Xtium3-CLHS_PX8_1.00.00.0000 -s -f1".\setup_uninstall.iss"
```

Where the **–s** switch specifies the silent mode and the **–f1** switch specifies the location of the response file. In this example, the switch –f1".\setup_uninstall.iss" specifies that the **`setup_uninstall.iss`** file be in the same folder as the device driver installer.

Silent Mode Installation Return Code

A silent mode installation creates a file “corinstall.ini” in the Windows directory. A section called [SetupResult] contains the ‘status’ of the installation. A value of 1 indicates that the installation has started and a value of 2 indicates that the installation has terminated.

A silent mode installation also creates a log file “setup.log” which by default is created in the same directory and with the same name (except for the extension) as the response file. The /f2 option enables you to specify an alternative log file location and file name, as in

Setup.exe /s /f2”C:\Setup.log”.

The “setup.log” file contains three sections. The first section, [InstallShield Silent], identifies the version of InstallShield used in the silent installation. It also identifies the file as a log file. The second section, [Application], identifies the installed application name, version, and the company name. The third section, [ResponseResult], contains the ‘ResultCode’ indicating whether the silent installation succeeded. A value of 0 means the installation was successful.

Installation Setup with CorAppLauncher.exe

The installation setup can be run with the CorAppLauncher.exe tool provided with the driver.

- Install the board driver and get CorAppLauncher.exe from the \bin directory of the installation.
- When running the installation, CorAppLauncher.exe will return only when the installation is finished.
- When run from within a batch file, obtain the installation exit code from the ERRORLEVEL value.
- The arguments to CorAppLauncher.exe are:
  -l: Launch application
  -f: Application to launch. Specify a fully qualified path.

As an example:

- CorAppLauncher –l –f”c:\driver_install\Xtium3-CLHS_PX8_1.00.00.0000.exe”
- IF %ERRORLEVEL% NEQ 0 goto launch error

Custom Driver Installation using install.ini

Customize the driver installation by parameters defined in the file “install.ini”.

By using this file, the user can:

- Select the user default configuration.
- Select different configurations for systems with multiple boards.
- Assign a standard Serial COM port to board.

Creating the install.ini File

- Install the driver in the target computer. All Xtium3-CLHS PX8 boards required in the system must be installed.
- Configure each board’s acquisition firmware using the Teledyne DALSA Device Manager tool (see Device Manager – Board Viewer).
- When each board setup is complete, using the Teledyne DALSA Device Manager tool, click on the **Save Config File** button. This will create the “install.ini” file.

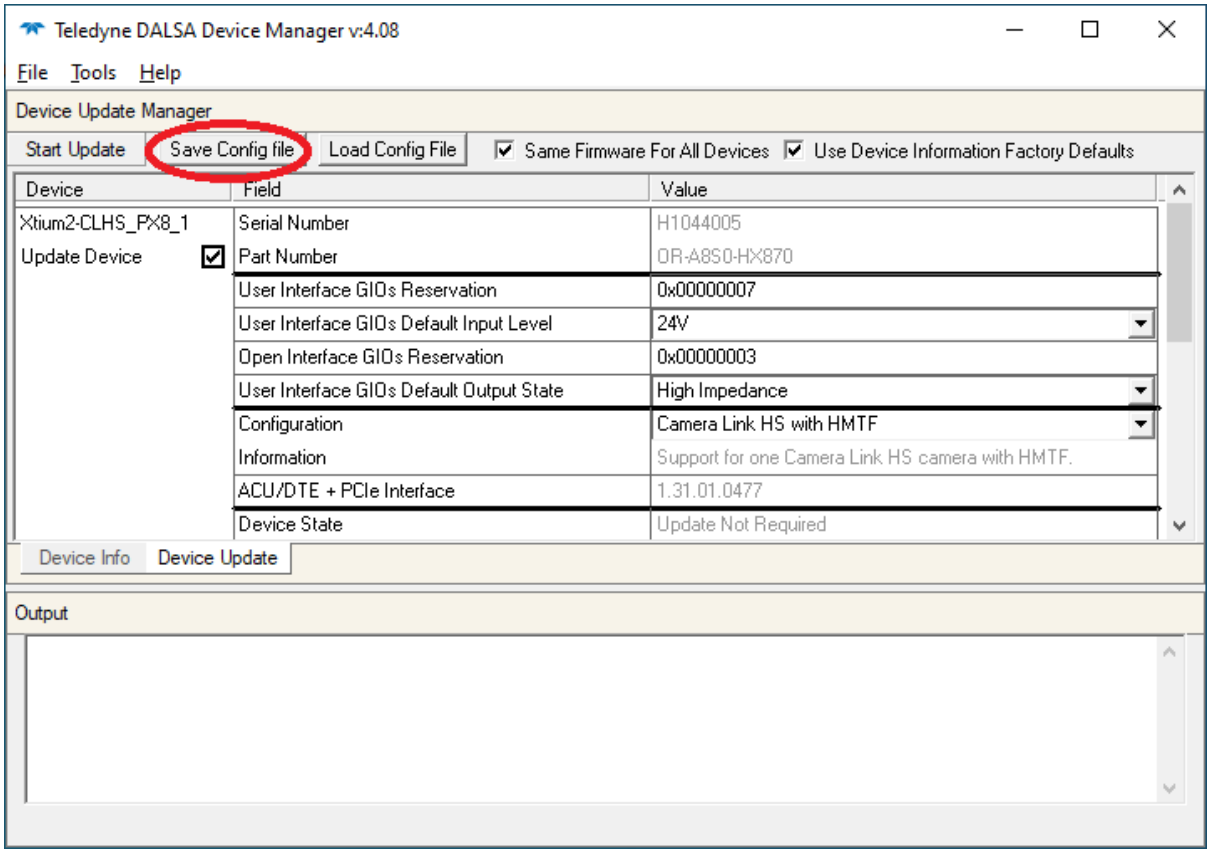


Figure 52: Create an install.ini File

Run the Installation using install.ini

Copy the install.ini file into the same directory as the setup installation file. Run the setup installation as normal. The installation will automatically check for an install.ini file and if found, use the configuration defined in it.

Appendix B: Troubleshooting Problems

Overview

The Xtium3-CLHS PX8 (and the Xtium3 family of products) is tested by Teledyne 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 Teledyne Technical Support.

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

Problem Type Summary

Xtium3-CLHS PX8 problems are either installation types where the board hardware is not recognized on the PCIe bus (that is, 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

Just after power up, if status LED L1 remains solid **RED** or flashing **RED**, the board firmware did not load correctly. If LED L1 is **BLUE** or flashing **BLUE**, the board is running from the safe mode load. This could indicate that the normal load in the flash is corrupted or not present.

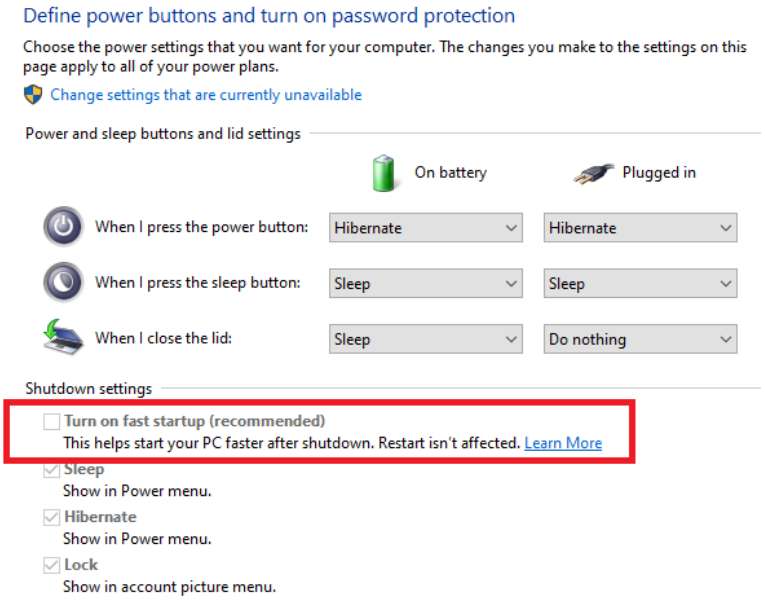
CLHS Link status is indicated by LED L2. The status colors displayed follow industry specifications for Camera Link HS.

The complete status LED descriptions are available in the technical reference section, (see Status LEDs Functional Descriptions).

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 Teledyne DALSA PCI Diagnostic tool as described in Checking for PCI Bus Conflicts. Also verify the installation via the Windows Device Manager.
- **BSOD (blue screen) following a board reset:** After programming the board with different firmware, the computer displays the BSOD when the board is reset (see BSOD (blue screen) Following a Board Reset).
- **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 for details. In addition, Teledyne technical support will ask for the log file of messages by Teledyne DALSA drivers. Follow the instructions describe in Teledyne DALSA Log Viewer.
- **Firmware update error:** There was an error during the Xtium3-CLHS PX8 firmware update procedure. The user can usually easily correct this. Follow the instructions Recovering from a Firmware Update Error.
- Installation went well but the board does not work or stopped working. Review these steps described in Symptoms: CamExpert Detects no Boards.

- **Using Windows 11 Fast Startup option:** When adding, removing, or moving boards while the PC is shutdown with the Windows Fast Startup option activated, it is possible that the boards do not get mapped properly on the next reboot of the computer. The driver will detect such a situation and the Device Manager launched at startup will display a message indicating that a reboot is required.



Possible Functional Problems

- **Driver Information:** Use the Teledyne DALSA device manager program to view information about the installed Xtium3-CLHS PX8 board and driver. See Driver Information via the Device Manager Program.
- **On-Board Image Memory Requirements:** The Xtium3-CLHS PX8 on-board memory can provide two frame buffers large enough for most imaging situations. See On-board Image Memory Requirements for Acquisitions for details on the on board memory and possible limitations.

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 Xtium3-CLHS PX8 functional problems.

- Symptoms: Xtium3-CLHS PX8 Does Not Grab
- Symptoms: Card grabs black
- Symptoms: Card acquisition bandwidth is less than expected

Troubleshooting Procedures

The following sections provide information and solutions to possible Xtium3-CLHS PX8 installation and functional problems. The previous section of this manual summarizes these topics.

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 • Teledyne 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 Teledyne 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.

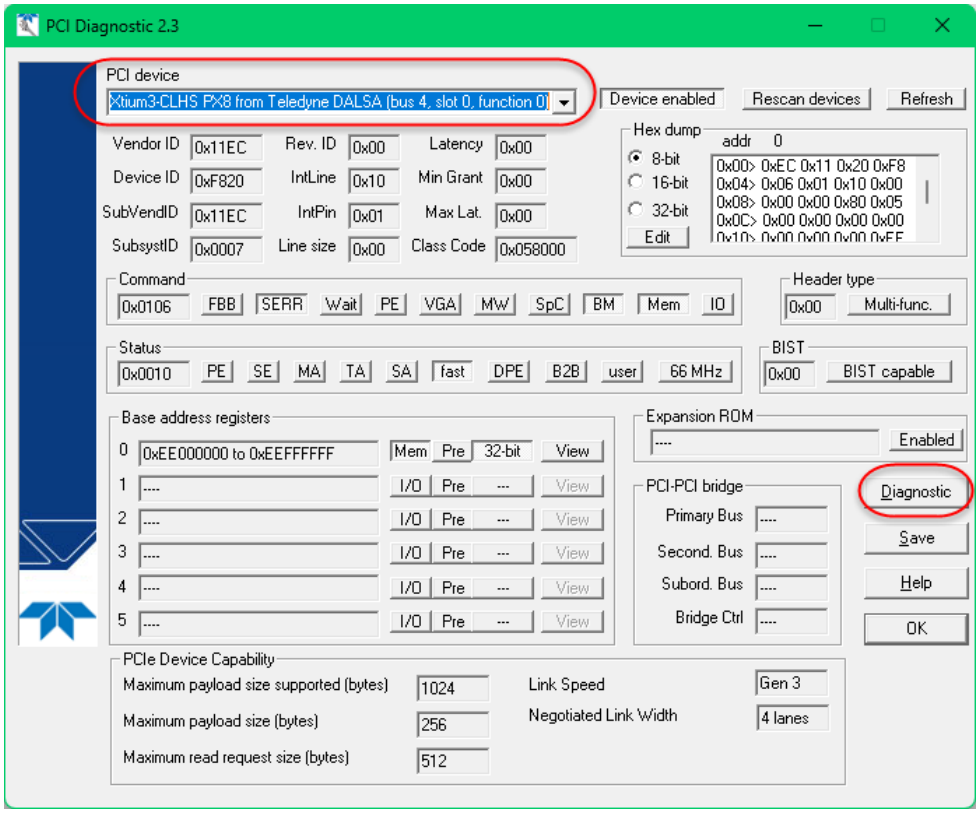


Figure 53: PCI Diagnostic Program

Clicking on the **Diagnostic** button opens a new window with the diagnostic report. From the PCI Bus Number drop menu, select the bus number that the Xtium3-CLHS PX8 is installed in—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 Teledyne Technical Support group along with a full description of your computer.

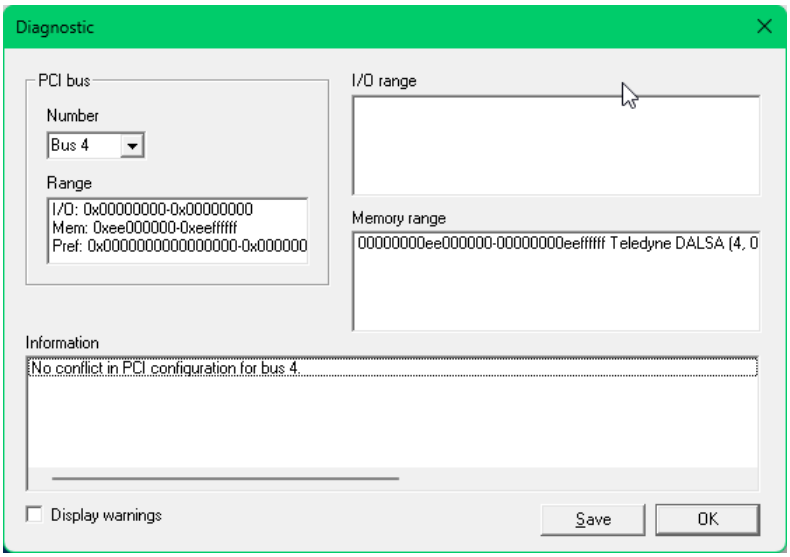


Figure 54: PCI Diagnostic Program – PCI bus info

Windows Device Manager

An alternative method to confirm the installation of the Xtium3-CLHS PX8 board and driver is to use the Windows Device manager tool. Use the Start Menu shortcut **Start • Control Panel • System • Device Manager**. As shown in the following screen images, look for Xtium3-*CLHS PX8* 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 an interrupt assigned to it, without conflicts.

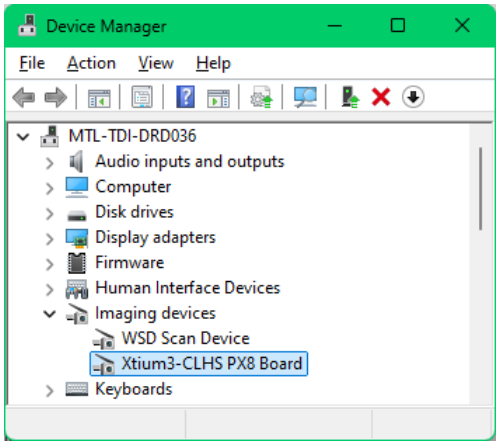


Figure 55: Using Windows Device Manager

BSOD (blue screen) Following a Board Reset

There are cases where a PC will falsely report a hardware malfunction when resetting the Xtium3-CLHS PX8 board. Ensure that you are using Sapera LT 7.50 or later.

Sapera and Hardware Windows Drivers

Any problem seen after installation, such as an error message running CamExpert, first make certain the appropriate Teledyne drivers have started successfully during the boot sequence. Example, click on the **Start • Programs • Accessories • System Tools • System Information • Software Environment** and click on **System Drivers**. Make certain the following drivers have started for the **Xtium3-CLHS PX8**.

Device	Description	Type	Started
CorXtium3CLHSPX8	Xtium3-CLHS PX8 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

Teledyne 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 Xtium3-CLHS PX8 firmware on installation or during a manual firmware upgrade. If on the case the board has corrupted firmware, any Sapera application such as CamExpert or the grab demo program will not find an installed board to control.

Possible reasons for firmware loading errors or corruption are:

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

When the Xtium3-CLHS PX8 firmware is corrupted, the board will automatically run from the Safe load after a PC reset.

***Solution*:** Update the board using the standard method described in section Firmware Update: Automatic Mode.

Driver Information via the Device Manager Program

The Device Manager program provides a convenient method of collecting information about the installed Xtium3-CLHS PX8. System information such as operating system, computer CPU, system memory, PCI configuration space, plus Xtium3-CLHS PX8 firmware information is displayed or written to a text file (default file name – BoardInfo.txt). Note that this program also manually uploads firmware to the Xtium3-CLHS PX8 (described elsewhere in this manual).

Execute the program via the Windows Start Menu shortcut .

Start • Programs • Teledyne DALSA • Xtium3-CLHS PX8 Device Driver • Device Manager.

If the Device Manager Program does not run, it will exit with a board was not found message.

Possible reasons for an error are:

- Board is not in the computer
- 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 its information shows in the right hand window, as described below.

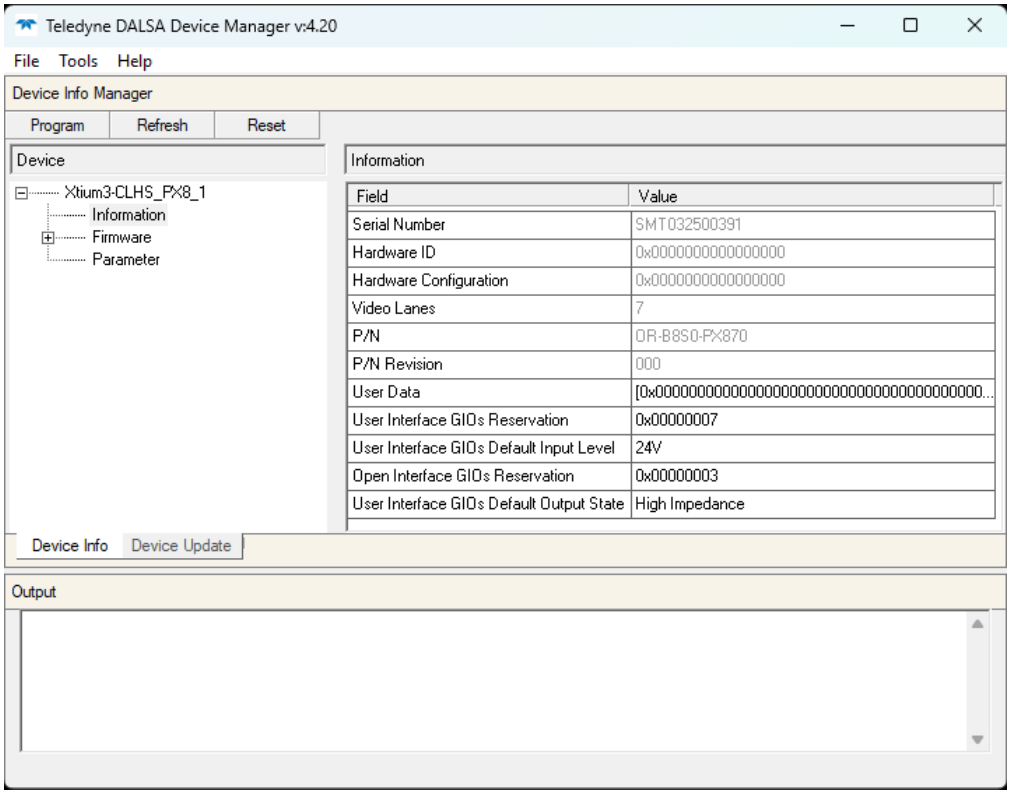


Figure 56: Board Firmware Version

- Select **Information** to display identification and information stored in the Xtium3-CLHS PX8 firmware.
- Select **Firmware** to display version information for the firmware components.
- Select one of the firmware components to load *custom* firmware when supplied by Teledyne 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.

Teledyne DALSA Log Viewer

The third step in the verification process is to save in a text file the information collected by the Log Viewer program. Run the program via the Windows Start Menu shortcut:

Teledyne DALSA • Sapera LT • Tools • Log Viewer

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

On-board Image Memory Requirements for Acquisitions

The Xtium3-CLHS PX8 by default will allocate the maximum number of buffers that can fit in on-board memory based on the size of the acquired image before cropping, to a maximum of 65535 buffers. Note that an application can change the default number of on-board frame buffers using the Sapera LT API. Usually two buffers will ensure that the acquired image 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, there is no interruption to the image acquisition of one buffer by any delays in transfer of the other buffer (which contains the previously acquired image frame) to system memory.

If allocation for the requested number of buffers fails, the driver will reduce the number of on-board frame buffers requested until they can all fit. If there is not enough memory for 2 on-board buffers, the driver will reduce the size such that it allocates two partial buffers. This mode is dependent on reading out the image data to the host computer faster than the incoming acquisition.

The maximum number of buffers that can fit in on-board memory can be calculated as follows:

- (Total On-Board memory / (Buffer Size in Bytes + 256 Bytes used to store the DMA)).

For example, assuming 512MB of on-board memory and acquiring 1024 x 1024 x 8 bit images, the number of on-board buffers would be:

- 512 MB / [(1024 x 1024) + 256] = 511.875 => 511 on- board buffers.

Symptoms: CamExpert Detects no Boards

- When starting CamExpert, with no Teledyne 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 an installed board frame grabber, troubleshoot the installation problem as described below.

Troubleshooting Procedure

When CamExpert detects no installed Teledyne board, there could be a hardware problem, a system bus problem, a kernel driver problem, or a software installation problem.

- Make certain that the card is seated in PCIe slot.
- Perform all installation checks described in this section before contacting Technical Support.
- Try the board in a different PCIe slot if available.

Symptoms: Xtium3-CLHS PX8 Does Not Grab

Sapera CamExpert does start but you do not see an image and the frame rate displayed is 0.

- Verify the camera has power.
- Verify the Camera Link HS cable is connected to the camera.
- Verify the camera and timing parameters with the camera in free run mode.
- Verify you can grab with the camera in free run mode.
- Make certain that you provide an external trigger if the camera configuration file requires one. Use the software trigger feature of CamExpert if you do not have a trigger source.
- Make certain that the camera configuration is the required mode. This must match the camera configuration file. Refer to your camera datasheet.
- Try to snap one frame instead of continuous grab.
- Perform all installation checks described in this section before contacting Technical Support.

Symptoms: Card grabs black

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

- Set your camera to manual exposure mode and set the exposure to a longer period, plus open the lens iris.
- Try to snap one frame instead of continuous grab.
- Make certain that the input LUT is not programmed to output all ‘0’s.
- 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 **BM** button under “Command” group. Make certain that the **BM** button is activated.

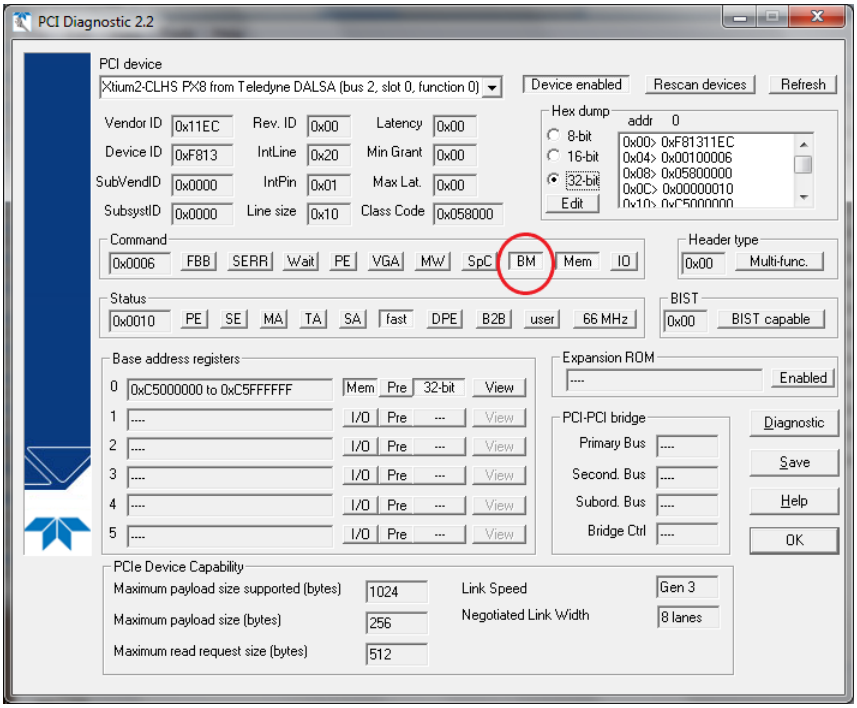


Figure 57: PCI Diagnostic Tool BM Button

- Perform all installation checks described in this section before contacting Technical Support.

Symptoms: Card acquisition bandwidth is less than expected

The Xtium3-CLHS PX8 acquisition bandwidth is less than expected.

- Review the system for problems or conflicts with other expansion boards or drivers.
- Remove other PCI Express 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 must be tested for bandwidth limitations affecting the imaging application.
- Is the Xtium3-CLHS PX8 installed in a PCI Express x16 slot?
- Note that some computer's x16 slot may only support non x16 boards at x1 or not at all. Check the computer documentation or test an Xtium3-CLHS PX8 installation. The speed at which the board is running can be viewed using the Diagnostic Tool provided with the driver.
- Is the Xtium3-CLHS PX8 installed in a PCI Express Gen1, Gen2 or Gen3 slot?
- Some older computers only have PCIe Gen1, Gen2 or Gen3 slots. The Generation at which the board is running is viewed using the Diagnostic Tool provided with the driver.
- Is the PCI maximum payload size smaller than 256 bytes?
- On some computers, this parameter can be changed in the PC's BIOS.

Appendix C: Sapera Parameter Reference

Xtium3-CLHS PX8 Supported Sapera Parameters

The tables below describe the Sapera capabilities supported by the Xtium3-CLHS PX8. Unless specified, each capability applies to all configuration modes and all acquisition modes.

Note: The information here is subject to change. The application needs to verify capabilities. New board driver releases may change product specifications.

Sapera describes the Xtium3-CLHS PX8 family as:

- Board Server: Xtium3-CLHS_PX8_1
- Acquisition Module: *dependent on firmware used*

Camera Related Capabilities

Capability	Values
CORACQ_CAP_CONNECTOR_TYPE	CORACQ_VAL_CONNECTOR_TYPE_CLHS (0x10)

Camera Related Parameters

Parameter		Values
CORACQ_PRM_CHANNEL	*Mono Standard*	CORACQ_VAL_CHANNEL_SINGLE (0x1) CORACQ_VAL_CHANNEL_DUAL (0x2)
CORACQ_PRM_FRAME		CORACQ_VAL_FRAME_PROGRESSIVE (0x2)
CORACQ_PRM_INTERFACE		CORACQ_VAL_INTERFACE_DIGITAL (0x2)
CORACQ_PRM_SCAN		CORACQ_VAL_SCAN_AREA (0x1) CORACQ_VAL_SCAN_LINE (0x2)
CORACQ_PRM_SIGNAL		CORACQ_VAL_SIGNAL_DIFFERENTIAL (0x2)
CORACQ_PRM_VIDEO	Mono	CORACQ_VAL_VIDEO_MONO (0x1)
CORACQ_PRM_PIXEL_DEPTH	Mono	8 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO8 8 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO16 10 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO8 10 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO16 12 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO8 12 bits, # LUT = 1, LUT format = CORDATA_FORMAT_MONO16
CORACQ_PRM_VIDEO_STD		CORACQ_VAL_VIDEO_STD_NON_STD (0x1)
CORACQ_PRM_FIELD_ORDER		CORACQ_VAL_FIELD_ORDER_NEXT_FIELD (0x4)
CORACQ_PRM_HACTIVE	Mono	min = 32 pixel max = 131072 pixel step = 32 pixel
CORACQ_PRM_VACTIVE		min = 1 line max = 16777215 line step = 1 line
CORACQ_PRM_TIME_INTEGRATE_METHOD		CORACQ_VAL_TIME_INTEGRATE_METHOD_10 (0x200) CORACQ_VAL_TIME_INTEGRATE_METHOD_11 (0x400) CORACQ_VAL_TIME_INTEGRATE_METHOD_12 (0x800)
CORACQ_PRM_CAM_TRIGGER_METHOD		CORACQ_VAL_CAM_TRIGGER_METHOD_3 (0x4)
CORACQ_PRM_CAM_NAME	Mono	Default Camera Link HS Line Scan Mono

CORACQ_PRM_LINE_INTEGRATE_METHOD		CORACQ_VAL_LINE_INTEGRATE_METHOD_8 (0x80) CORACQ_VAL_LINE_INTEGRATE_METHOD_9 (0x100) CORACQ_VAL_LINE_INTEGRATE_METHOD_10 (0x200)
CORACQ_PRM_LINE_TRIGGER_METHOD		CORACQ_VAL_LINE_TRIGGER_METHOD_2 (0x2)
CORACQ_PRM_LINE_TRIGGER_DELAY		min = 0 nsec max = 4294967295 nsec step = 1 nsec
CORACQ_PRM_TAPS		Max = 1 tap
CORACQ_PRM_TAP_1_DIRECTION	Mono Standard	CORACQ_VAL_TAP_DIRECTION_LR (0x1) 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_CHANNELS_ORDER	Mono Standard	CORACQ_VAL_CHANNELS_ORDER_NORMAL (0x1) CORACQ_VAL_CHANNELS_ORDER_REVERSE (0x2)
CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MIN		1 Hz
CORACQ_PRM_CAM_LINE_TRIGGER_FREQ_MAX		10000000 Hz
CORACQ_PRM_CAM_TIME_INTEGRATE_DURATION_MIN		1 µs
CORACQ_PRM_CAM_TIME_INTEGRATE_DURATION_MAX		42949672 µs
CORACQ_PRM_CAM_IO_CONTROL (*)		
CORACQ_PRM_LINE_INTEGRATE_PULSE0_DELAY		min = 0 nsec max = 4294967295 nsec step = 1 nsec
CORACQ_PRM_DATA_VALID_ENABLE		TRUE FALSE
CORACQ_PRM_CAM_CONTROL_DURING_READOUT		CORACQ_VAL_CAM_CONTROL_DURING_READOUT_INVALID (0x0) CORACQ_VAL_CAM_CONTROL_DURING_READOUT_VALID (0x01) CORACQ_VAL_CAM_CONTROL_DURING_READOUT_IGNORE (0x2)
CORACQ_PRM_DATA_LANES	PX8	min = 1 lane, max = 7 lanes, step = 1 lane
CORACQ_PRM_BIT_TRANSFER_RATE		10 Gbps
CORACQ_PRM_CLHS_CONFIGURATION		CORACQ_VAL_CLHS_CONFIGURATION_BOARD_MASTER (0x1) CORACQ_VAL_CLHS_CONFIGURATION_BOARD_SLAVE (0x2) CORACQ_VAL_CLHS_CONFIGURATION_MANUAL_ACQ_START_STOP (0x8)
CORACQ_PRM_MULTI_PLANES		Number of planes output by the camera.

VIC Related Parameters

Parameter		Values
CORACQ_PRM_CAMSEL	Mono	CAMSEL_MONO = 0
CORACQ_PRM_CROP_LEFT	Mono	min = 0 pixel max = 131040 pixel step = 4 pixel
CORACQ_PRM_CROP_TOP		min = 0 line max = 16777215 line step = 1 line
CORACQ_PRM_CROP_WIDTH	Mono	min =32 pixel max = 131072 pixel step =4 pixel * Maximum allowed is 128 kBytes, so maximum in pixels is dependent on the output pixel depth.
CORACQ_PRM_CROP_HEIGHT		min = 1 line max = 16777215 line step = 1 line
CORACQ_PRM_DECIMATE_METHOD		CORACQ_VAL_DECIMATE_DISABLE (0x1)
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_3 (0x4) CORACQ_VAL_STROBE_METHOD_4 (0x8) CORACQ_VAL_STROBE_METHOD_5 (0x10)
CORACQ_PRM_STROBE_POLARITY		CORACQ_VAL_ACTIVE_LOW (0x1) CORACQ_VAL_ACTIVE_HIGH (0x2)
CORACQ_PRM_STROBE_DURATION		min = 1 µs max = 17179869 µs step = 1 µs
CORACQ_PRM_STROBE_DELAY		min = 0 µs max = 17179869 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_ENABLE		TRUE FALSE
CORACQ_PRM_TIME_INTEGRATE_DURATION		min = 1 µs max = 17179869 µs step = 1 µs
CORACQ_PRM_CAM_TRIGGER_ENABLE		TRUE FALSE
CORACQ_PRM_OUTPUT_FORMAT	Mono	CORACQ_VAL_OUTPUT_FORMAT_MONO8 CORACQ_VAL_OUTPUT_FORMAT_MONO16 CORACQ_VAL_OUTPUT_FORMAT_MONO8P2 CORACQ_VAL_OUTPUT_FORMAT_MONO8P3 CORACQ_VAL_OUTPUT_FORMAT_MONO8P4 CORACQ_VAL_OUTPUT_FORMAT_MONO16P2 CORACQ_VAL_OUTPUT_FORMAT_MONO16P3 CORACQ_VAL_OUTPUT_FORMAT_MONO16P4
CORACQ_PRM_EXT_TRIGGER_ENABLE		CORACQ_VAL_EXT_TRIGGER_OFF (0x1) CORACQ_VAL_EXT_TRIGGER_ON (0x8)
CORACQ_PRM_VIC_NAME	Mono	Default Camera Link HS Line Scan Mono
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_LUT_FORMAT	Mono	Default = CORDATA_FORMAT_MONO8
CORACQ_PRM_LINE_INTEGRATE_ENABLE		TRUE FALSE
CORACQ_PRM_LINE_INTEGRATE_DURATION		min = 1 nsec max = 4294967295 nsec step = 1 nsec
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_VAL_DOUBLE_PULSE_RISING_EDGE_ORDER_REVERSE (0x80) CORACQ_VAL_DOUBLE_PULSE_FALLING_EDGE_ORDER_REVERSE (0x100)
CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE		TRUE FALSE
CORACQ_PRM_EXT_LINE_TRIGGER_DETECTION		CORACQ_VAL_RISING_EDGE (0x4) CORACQ_VAL_FALLING_EDGE (0x8)
CORACQ_PRM_SNAP_COUNT		Not Available
CORACQ_PRM_INT_LINE_TRIGGER_ENABLE		TRUE FALSE
CORACQ_PRM_INT_LINE_TRIGGER_FREQ		Default = 5000 Hz
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_VAL_LEVEL_12VOLTS (0x040) CORACQ_VAL_LEVEL_24VOLTS (0x8)
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_VAL_LEVEL_12VOLTS (0x040) CORACQ_VAL_LEVEL_24VOLTS (0x8)
CORACQ_PRM_EXT_LINE_TRIGGER_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1) CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_INT_LINE_TRIGGER_FREQ_MIN	8 Hz
CORACQ_PRM_INT_LINE_TRIGGER_FREQ_MAX	1000000 Hz
CORACQ_PRM_MASTER_MODE	Not available
CORACQ_PRM_SHAFT_ENCODER_DROP	min = 0 tick max = 254 tick step = 1 tick
CORACQ_PRM_SHAFT_ENCODER_ENABLE	TRUE FALSE
CORACQ_PRM_EXT_TRIGGER_FRAME_COUNT	min = 1 frame max = 16777214 frame step = 1 frame
CORACQ_PRM_INT_FRAME_TRIGGER_ENABLE	TRUE FALSE
CORACQ_PRM_INT_FRAME_TRIGGER_FREQ	min = 1 milli-Hz max = 41000000 milli-Hz step = 1 milli-Hz
CORACQ_PRM_STROBE_DELAY_2	Not Available
CORACQ_PRM_FRAME_LENGTH	CORACQ_VAL_FRAME_LENGTH_FIX (0x1) CORACQ_VAL_FRAME_LENGTH_VARIABLE (0x2)
CORACQ_PRM_FLIP	CORACQ_VAL_FLIP_OFF (0x00) CORACQ_VAL_FLIP_HORZ (0x01)
CORACQ_PRM_EXT_TRIGGER_DURATION	min = 0 µs max = 255 µs step = 1 µs
CORACQ_PRM_TIME_INTEGRATE_DELAY	min = 0 µs max = 17179869 µs step = 1 µs
CORACQ_PRM_CAM_TRIGGER_DELAY	min = 0 µs max = 17179869 µs step = 1 µs
CORACQ_PRM_SHAFT_ENCODER_LEVEL	CORACQ_VAL_LEVEL_TTL (0x1) CORACQ_VAL_LEVEL_422 (0x2)
CORACQ_PRM_LUT_NENTRIES 8-bit/pixel component 10-bit/pixel component 12-bit/pixel component	256 entries 1024 entries 4096 entries
CORACQ_PRM_EXT_FRAME_TRIGGER_SOURCE (*)	min = 0 max = 7 step = 1
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE (*)	min = 0 max = 5 step = 1
CORACQ_PRM_EXT_TRIGGER_SOURCE (*)	min = 0 max = 7 step = 1
CORACQ_PRM_SHAFT_ENCODER_MULTIPLY	min = 1 max = 32 step = (2^N)
CORACQ_PRM_SHAFT_ENCODER_FRACTIONAL_MULTIPLIER	min = 0.0000001 max = 268
CORACQ_PRM_EXT_TRIGGER_DELAY	min = 0 max = 16777215 step = 1
CORACQ_PRM_EXT_TRIGGER_DELAY_TIME_BASE	CORACQ_VAL_TIME_BASE_LINE_VALID (0x4) CORACQ_VAL_TIME_BASE_LINE_TRIGGER (0x8) CORACQ_VAL_TIME_BASE_SHAFT_ENCODER (0x40) CORACQ_VAL_TIME_BASE_NS (0x80)
CORACQ_PRM_COLOR_DECODER_ENABLE	Not Available
CORACQ_PRM_EXT_TRIGGER_IGNORE_DELAY	min = 0 max = 42949672 step = 1

CORACQ_PRM_BOARD_SYNC_OUTPUT1_SOURCE (*)	min = 0 max = 6 step = 1
CORACQ_PRM_BOARD_SYNC_OUTPUT2_SOURCE (*)	min = 0 max = 6 step = 1
CORACQ_PRM_EXT_TRIGGER_SOURCE_STR	[0] = Automatic [1] = External Trigger #1 [2] = External Trigger #2 [3] = Board Sync #1 [4] = Board Sync #2 [5] = Software Trigger [6] = Data Valid [7] = Board Sync #3
CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE_STR	[0] = Automatic [1] = Shaft Encoder Phase A [2] = Shaft Encoder Phase B [3] = Shaft Encoder Phase A & B [4] = Board Sync #1 [5] = Board Sync #2
CORACQ_PRM_VERTICAL_TIMEOUT_DELAY	Not Available
CORACQ_PRM_POCL_ENABLE	TRUE FALSE
CORACQ_PRM_SHAFT_ENCODER_DIRECTION	CORACQ_VAL_SHAFT_ENCODER_DIRECTION_IGNORE (0x00) CORACQ_VAL_SHAFT_ENCODER_DIRECTION_FORWARD (0x01) CORACQ_VAL_SHAFT_ENCODER_DIRECTION_REVERSE (0x02) CORACQ_VAL_SHAFT_ENCODER_DIRECTION_RESCAN (0x4) CORACQ_VAL_SHAFT_ENCODER_DIRECTION_COUNT (0x8)
CORACQ_PRM_LINE_TRIGGER_AUTO_DELAY	CORACQ_VAL_LINE_TRIGGER_AUTO_DELAY_DISABLE (0x0) CORACQ_VAL_LINE_TRIGGER_AUTO_DELAY_FREQ_MAX (0x2)
CORACQ_PRM_TIME_STAMP_BASE	CORACQ_VAL_TIME_BASE_US (0x1) CORACQ_VAL_TIME_BASE_LINE_VALID (0X4) CORACQ_VAL_TIME_BASE_LINE_TRIGGER (0X8) CORACQ_VAL_TIME_BASE_SHAFT_ENCODER (0X40) CORACQ_VAL_TIME_BASE_100NS (0x200)
CORACQ_PRM_BOARD_SYNC_OUTPUT1_SOURCE_STR	[0] = Disabled [1] = External Frame Trigger [2] = Reserved [3] = External Trigger Ignore Region [4] = Shaft Encoder Before Mult/Drop [5] = Shaft Encoder After Mult/Drop [6] = Internal Line Trigger
CORACQ_PRM_BOARD_SYNC_OUTPUT2_SOURCE_STR	[0] = Disabled [1] = External Frame Trigger [2] = Reserved [3] = External Trigger Ignore Region [4] = Shaft Encoder Before Mult/Drop [5] = Shaft Encoder After Mult/Drop [6] = Internal Line Trigger
CORACQ_PRM_SHAFT_ENCODER_ORDER	CORACQ_VAL_SHAFT_ENCODER_ORDER_AUTO (0X0) CORACQ_VAL_SHAFT_ENCODER_ORDER_DROP_MULTIPLY (0X1) CORACQ_VAL_SHAFT_ENCODER_ORDER_MULTIPLY_DROP (0X2) * For auto mode, the order is multiply/drop.
CORACQ_PRM_CAM_FRAMES_PER_TRIGGER	min = 0 max = 65534 step = 1
CORACQ_PRM_LINE_INTEGRATE_TIME_BASE	CORACQ_VAL_TIME_BASE_NS (0X80)
CORACQ_PRM_EXT_TRIGGER_IGNORE_REGION_DURATION	min = 0 µs max = 6553 µs step = 1 µs
CORACQ_PRM_STROBE_DESTINATION	min = 0 max = 4 step = 1
CORACQ_PRM_STROBE_DESTINATION_STR	[0] = Automatic [1] = Output #1 (LVTTL) [2] = Output #2 (LVTTL) [3] = Output #5 (Open Collector) [4] = Output #6 (Open Collector)
CORACQ_PRM_SHAFT_ENCODER_AVERAGING_ENABLE	TRUE FALSE

CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PULSES	max 'N' = 8 (ie 2**N Pulses)
CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PERIOD_MIN	50 nsec
CORACQ_PRM_SHAFT_ENCODER_AVERAGING_PERIOD_MAX	2000000 nsec
CORACQ_PRM_PLANE_OPS_PASSTHROUGH	Sets the passthough plane used for multiplane processing operations set by CORACQ_PRM_PLANE_OPS.
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_DISABLE (0x00000000) CORACQ_VAL_PLANE_OPS_AVG (0x00000001) CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION (0x00000002) CORACQ_VAL_PLANE_OPS_SUM (0x00000004) CORACQ_VAL_PLANE_OPS_PASSTHROUGH (0x00000008)

The following table describes the line-trigger source types supported by the Xtium3-CLHS PX8.

CORACQ_PRM_EXT_LINE_TRIGGER_SOURCE

PRM Value	Input used as: External Line Trigger	Input used as: External Shaft Encoder
	if CORACQ_PRM_EXT_LINE_TRIGGER_ENABLE = *true*	*if* CORACQ_PRM_SHAFT_ENCODER_ENABLE =*true*
0	From Shaft Encoder Phase A	From Shaft Encoder Phase A & B
1	From Shaft Encoder Phase A	From Shaft Encoder Phase A
2	From Shaft Encoder Phase B	From Shaft Encoder Phase B
3	n/a	From Shaft Encoder Phase A & B
4	From Board Sync #1	n/a
5	From Board Sync #2	n/a

ACQ Related Parameters

Parameter		Values
CORACQ_PRM_LABEL		Camera Link HS Mono
CORACQ_PRM_EVENT_TYPE		CORACQ_VAL_EVENT_TYPE_START_OF_FRAME CORACQ_VAL_EVENT_TYPE_END_OF_FRAME CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER CORACQ_VAL_EVENT_TYPE_VERTICAL_SYNC CORACQ_VAL_EVENT_TYPE_FRAME_LOST CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED CORACQ_VAL_EVENT_TYPE_EXT_LINE_TRIGGER_TOO_SLOW CORACQ_VAL_EVENT_TYPE_LINK_ERROR CORACQ_VAL_EVENT_TYPE_SHAFT_ENCODER_REVERSE_COUNT_OVERFLOW CORACQ_VAL_EVENT_TYPE_LINE_TRIGGER_TOO_FAST CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_END
CORACQ_PRM_EVENT_TYPE_EX		CORACQ_VAL_EVENT_TYPE_START_OF_FRAME CORACQ_VAL_EVENT_TYPE_END_OF_FRAME CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER CORACQ_VAL_EVENT_TYPE_VERTICAL_SYNC CORACQ_VAL_EVENT_TYPE_FRAME_LOST CORACQ_VAL_EVENT_TYPE_DATA_OVERFLOW CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_IGNORED CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_TOO_SLOW CORACQ_VAL_EVENT_TYPE_LINK_ERROR CORACQ_VAL_EVENT_TYPE_SHAFT_ENCODER_REVERSE_COUNT_OVERFLOW CORACQ_VAL_EVENT_TYPE_LINE_TRIGGER_TOO_FAST CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER_END CORACQ_VAL_EVENT_TYPE_LINK_LOCK\ CORACQ_VAL_EVENT_TYPE_LINK_UNLOCK CORACQ_VAL_EVENT_TYPE_CAMERA_MISSED_TRIGGER CORACQ_VAL_EVENT_TYPE_CAMERA_OVERRUN CORACQ_VAL_EVENT_TYPE_SLAVE_LINK_LOCK CORACQ_VAL_EVENT_TYPE_SLAVE_LINK_UNLOCK
CORACQ_PRM_SIGNAL_STATUS		CORACQ_VAL_SIGNAL_HSYNC_PRESENT CORACQ_VAL_SIGNAL_VSYNC_PRESENT CORACQ_VAL_SIGNAL_POWER_PRESENT CORACQ_VAL_SIGNAL_POCL_ACTIVE CORACQ_VAL_SIGNAL_POCL_ACTIVE_2 CORACQ_VAL_SIGNAL_LINK_LOCK CORACQ_VAL_SIGNAL_LANE1_LOCK CORACQ_VAL_SIGNAL_LANE2_LOCK CORACQ_VAL_SIGNAL_LANE3_LOCK CORACQ_VAL_SIGNAL_LANE4_LOCK CORACQ_VAL_SIGNAL_LANE5_LOCK CORACQ_VAL_SIGNAL_LANE6_LOCK CORACQ_VAL_SIGNAL_LANE7_LOCK CORACQ_VAL_SIGNAL_SLAVE_LINK_LOCK
CORACQ_PRM_FLAT_FIELD_ENABLE	PX8	TRUE / FALSE
CORACQ_CAP_FLAT_FIELD_OFFSET	8-bit Mono	min = 0 max = 255 step = 1
	10-bit Mono	min = 0 max = 1023 step = 1
	12-bit Mono	min = 0 max = 4095 step = 1
CORACQ_CAP_FLAT_FIELD_GAIN	8-bit Mono	min = 0 max = 255 step = 1
	10-bit Mono	min = 0 max = 1023 step = 1
	12-bit Mono	min = 0 max = 4095 step = 1

CORACQ_CAP_FLAT_FIELD_GAIN_DIVISOR	8-bit Mono 10-bit Mono 12-bit Mono	128 512 2048
CORACQ_PRM_FLAT_FIELD_PIXEL_REPLACEMENT _METHOD		CORACQ_VAL_FLAT_FIELD_PIXEL_REPLACEMENT_METHOD_2 (Pixel replacement is done by averaging the 2 neighborhood pixels. When one of the neighbors is not available (border image pixels, the pixel is simply replaced with the available neighbor)
CORACQ_PRM_FLAT_FIELD_SET_SELECT		min = 0 max = 16 step = 1
CORACQ_PRM_TIME_STAMP		Available
CORACQ_CAP_SERIAL_PORT_INDEX		Not Available
CORACQ_PRM_IMAGE_FILTER_ENABLE	Mono	Not Available
CORACQ_PRM_SHAFT_ENCODER_REVERSE_COUNT		Max = 65536 ticks
CORACQ_PRM_META_DATA		Not available
CORACQ_PRM_SHAFT_ENCODER_STATUS		CORACQ_VAL_SHAFT_ENCODER_STATUS_DIRECTION_FORWARD / CORACQ_VAL_SHAFT_ENCODER_STATUS_DIRECTION_REVERSE (0x1) CORACQ_VAL_SHAFT_ENCODER_STATUS_TOO_SLOW (0x2) CORACQ_VAL_SHAFT_ENCODER_STATUS_REVERSE_COUNT_OVERFLOW (0x4)
CORACQ_PRM_SHAFT_ENCODER_COUNT		Available

Transfer Related Capabilities

Capability	Values
CORXFER_CAP_NB_INT_BUFFERS	CORXFER_VAL_NB_INT_BUFFERS_AUTO (0x2)
CORXFER_CAP_MAX_XFER_SIZE	4294967040 Bytes
CORXFER_CAP_MAX_FRAME_COUNT	16777215 Frames
CORXFER_CAP_COUNTER_STAMP_AVAILABLE	FALSE
CORXFER_CAP_TRANSFER_SYNC	CORXFER_VAL_TRANSFER_SYNC_SUPPORTED (0x1)

Transfer Related Parameters

Parameter	Values
CORXFER_PRM_EVENT_TYPE CORXFER_PRM_EVENT_TYPE_EX	CORXFER_VAL_EVENT_TYPE_START_OF_FRAME CORXFER_VAL_EVENT_TYPE_END_OF_FRAME CORXFER_VAL_EVENT_TYPE_END_OF_LINE CORXFER_VAL_EVENT_TYPE_END_OF_NLINES CORXFER_VAL_EVENT_TYPE_END_OF_TRANSFER
CORXFER_PRM_START_MODE	CORXFER_VAL_START_MODE_ASYNCHRONOUS (0x0) CORXFER_VAL_START_MODE_SYNCHRONOUS (0x1) CORXFER_VAL_START_MODE_HALF_ASYNCHRONOUS (0x2) CORXFER_VAL_START_MODE_SEQUENTIAL (0x3)
CORXFER_PRM_CYCLE_MODE	CORXFER_VAL_CYCLE_MODE_ASYNCHRONOUS (0x0) CORXFER_VAL_CYCLE_MODE_SYNCHRONOUS_WITH_TRASH (0x2) CORXFER_VAL_CYCLE_MODE_OFF (0x3) CORXFER_VAL_CYCLE_MODE_SYNCHRONOUS_NEXT_EMPTY_WITH_TRASH (0x5)
CORXFER_PRM_FLIP	CORXFER_VAL_FLIP_OFF (0x0) CORXFER_VAL_FLIP_VERT (0x2)
CORXFER_PRM_INT_BUFFERS	* Depends on acquired image size. By default driver will optimize the number of on-board buffers.
CORXFER_PRM_EVENT_COUNT_SOURCE	CORXFER_VAL_EVENT_COUNT_SOURCE_DST (0x1) CORXFER_VAL_EVENT_COUNT_SOURCE_SRC (0x2)
CORXFER_PRM_BUFFER_TIMESTAMP_MODULE	CORXFER_VAL_BUFFER_TIMESTAMP_MODULE_ACQ (0x1) CORXFER_VAL_BUFFER_TIMESTAMP_MODULE_XFER (0x13)
CORXFER_PRM_BUFFER_TIMESTAMP_EVENT (ACQ Related)	CORACQ_VAL_EVENT_TYPE_START_OF_FRAME (0x80000) CORACQ_VAL_EVENT_TYPE_EXTERNAL_TRIGGER (0x1000000)
CORXFER_PRM_BUFFER_TIMESTAMP_EVENT (XFER Related)	CORXFER_VAL_EVENT_TYPE_END_OF_FRAME (0x800000)
CORXFER_PRM_LINE_MERGING	CORXFER_VAL_LINE_MERGING_AUTO (0x0) CORXFER_VAL_LINE_MERGING_OFF (0x2)

General Outputs #1: Capabilities (for GIO Module #0)

Outputs are available on connector J1 and J4.

Capability	Values
CORGIO_CAP_IO_COUNT	8 I/Os
CORGIO_CAP_DIR_OUTPUT	0xff
CORGIO_CAP_DIR_TRISTATE	0xff
CORGIO_CAP_EVENT_TYPE	Not Available
CORGIO_CAP_READ_ONLY	0x03 (* depends on strobe outputs reserved for acquisition device)

General Outputs #1: Parameters (for GIO Module #0)

Parameter	Values
CORGIO_PRM_LABEL	General Outputs #1
CORGIO_PRM_DEVICE_ID	0
CORGIO_PRM_OUTPUT_TYPE	CORGIO_VAL_OUTPUT_TYPE_LVTTL (0x20)
CORGIO_PRM_CONNECTOR	CORGIO_VAL_CONNECTOR_1 (0x1)

General Inputs #1: Capabilities (for GIO Module #1)

Inputs are available on connector J1 and J4.

Capability	Values
CORGIO_CAP_IO_COUNT	6 I/Os
CORGIO_CAP_DIR_OUTPUT	0x0
CORGIO_CAP_DIR_TRISTATE	0x0
CORGIO_CAP_EVENT_TYPE	CORGIO_VAL_EVENT_TYPE_RISING_EDGE (0x1) CORGIO_VAL_EVENT_TYPE_FALLING_EDGE (0x2)
CORGIO_CAP_READ_ONLY	0x03 (* depends on external trigger inputs reserved for acquisition device)

General Inputs #1: Parameters (for GIO Module #1)

Parameter	Values
CORGIO_PRM_LABEL	General Inputs #1
CORGIO_PRM_DEVICE_ID	1
CORGIO_PRM_INPUT_LEVEL	CORGIO_VAL_INPUT_LEVEL_TTL (0x1) CORGIO_VAL_INPUT_LEVEL_422 (0x2) CORGIO_VAL_INPUT_LEVEL_24VOLTS (0x8) CORGIO_VAL_INPUT_LEVEL_12VOLTS (0x40)
CORGIO_PRM_CONNECTOR	CORGIO_VAL_CONNECTOR_1 (0x1)

Bidirectional Board-to-Board I/Os: Capabilities (for GIO Module #2)

These I/Os are available on connector J5.

Capability	Values
CORGIO_CAP_IO_COUNT	8 I/Os
CORGIO_CAP_DIR_OUTPUT	0xff
CORGIO_CAP_DIR_TRISTATE	0xff
CORGIO_CAP_EVENT_TYPE	Not Available
CORGIO_CAP_READ_ONLY	0x03 (* depends on board syncs reserved for acquisition device)

Bidirectional Board-to-Board I/Os: Parameters (for GIO Module #2)

Parameter	Values
CORGIO_PRM_LABEL	Bidirectional General I/Os #1
CORGIO_PRM_DEVICE_ID	2
CORGIO_PRM_OUTPUT_TYPE	CORGIO_VAL_OUTPUT_TYPE_LVTTL (0x20)
CORGIO_PRM_INPUT_LEVEL	CORGIO_VAL_INPUT_LEVEL_LVTTL (0x20)
CORGIO_PRM_CONNECTOR	CORGIO_VAL_CONNECTOR_2 (0x2)

Open Collector General Outputs: Capabilities (for GIO Module #3)

Capability	Values
CORGIO_CAP_IO_COUNT	8 I/Os
CORGIO_CAP_DIR_OUTPUT	0xff
CORGIO_CAP_DIR_TRISTATE	0x0
CORGIO_CAP_EVENT_TYPE	Not Available
CORGIO_CAP_READ_ONLY	0x0

Open Collector General Outputs: Parameters (for GIO Module #3)

Parameter	Values
CORGIO_PRM_LABEL	General Outputs Open Collector #2
CORGIO_PRM_DEVICE_ID	3
CORGIO_PRM_OUTPUT_TYPE	Not available
CORGIO_PRM_INPUT_LEVEL	Not available
CORGIO_PRM_CONNECTOR	CORGIO_VAL_CONNECTOR_1 (0x1)

Multi-Plane Input Processing Parameters

For related Sapera LT parameters for using multi-plane image processing see:

- 2-Plane Processing Operations
- 3-Plane Processing Operations
- 4-Plane Processing Operations

2-Plane Processing Operations

Standard features.

Summation with Saturation

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	2
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8

10-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION
CORACQ_PRM_PIXEL_DEPTH	10
CORACQ_PRM_MULTI_PLANES	2
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

Averaging

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_AVG
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	2
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8

10-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_AVG
CORACQ_PRM_PIXEL_DEPTH	10
CORACQ_PRM_MULTI_PLANES	2
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

3-Plane Processing Operations

Standard features

Summation

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

Summation with Saturation

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8

10-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION
CORACQ_PRM_PIXEL_DEPTH	10
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

12-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION
CORACQ_PRM_PIXEL_DEPTH	12
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

Summation with Saturation Dual Output

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION | CORACQ_VAL_PLANE_OPS_PASSTHROUGH
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_PLANE_OPS_PASSTHROUGH	Specify passthrough (dark) plane; any input plane can be used. Possible values 0, 1 or 2.
CORACQ_PRM_OUTPUT_FORMAT	CCORACQ_VAL_OUTPUT_FORMAT_MONO82P

Averaging

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_AVG
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8

10-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_AVG
CORACQ_PRM_PIXEL_DEPTH	10
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

12-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_AVG
CORACQ_PRM_PIXEL_DEPTH	12
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO16

Averaging with Dual Output

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_AVG | CORACQ_VAL_PLANE_OPS_PASSTHROUGH
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	3
CORACQ_PRM_PLANE_OPS_PASSTHROUGH	Specify passthrough (dark) plane; any input plane can be used. Possible values 0, 1 or 2.
CORACQ_PRM_OUTPUT_FORMAT	CCORACQ_VAL_OUTPUT_FORMAT_MONO82P

4-Plane Processing Operations

4-Plane processing operations are advanced features and require a license; for information contact Teledyne sales.

Summation with Saturation

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS__SUM_WITH_SATURATION
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	4
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8

Summation with Saturation Dual Output

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_SATURATION | CORACQ_VAL_PLANE_OPS_PASSTHROUGH
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	4
CORACQ_PRM_PLANE_OPS_PASSTHROUGH	Specify passthrough (dark) plane; any input plane can be used. Possible values 0, 1, 2 or 3.
CORACQ_PRM_OUTPUT_FORMAT	CCORACQ_VAL_OUTPUT_FORMAT_MONO82P

Averaging

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_AVG
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	4
CORACQ_PRM_OUTPUT_FORMAT	CORACQ_VAL_OUTPUT_FORMAT_MONO8

Averaging with Dual Output

8-bits/pixel input:

Parameter	Value
CORACQ_PRM_PLANE_OPS	CORACQ_VAL_PLANE_OPS_SUM_WITH_AVG | CORACQ_VAL_PLANE_OPS_PASSTHROUGH
CORACQ_PRM_PIXEL_DEPTH	8
CORACQ_PRM_MULTI_PLANES	4
CORACQ_PRM_PLANE_OPS_PASSTHROUGH	Specify passthrough (dark) plane; any input plane can be used. Possible values 0, 1, 2 or 3.
CORACQ_PRM_OUTPUT_FORMAT	CCORACQ_VAL_OUTPUT_FORMAT_MONO82P

Appendix D: Xtium Frame Grabber User Commands

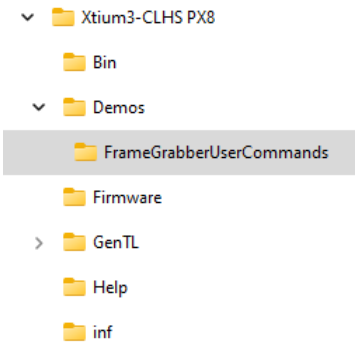
This appendix defines the Xtium User Commands available on frame grabbers to access features that do not have a direct equivalent in the SaperaLT API. In general, these details are available through the frame grabber utilities, such as the Device Manager and Diagnostic Tool. These commands allow these parameters to be accessed by user applications, as required.

Overview

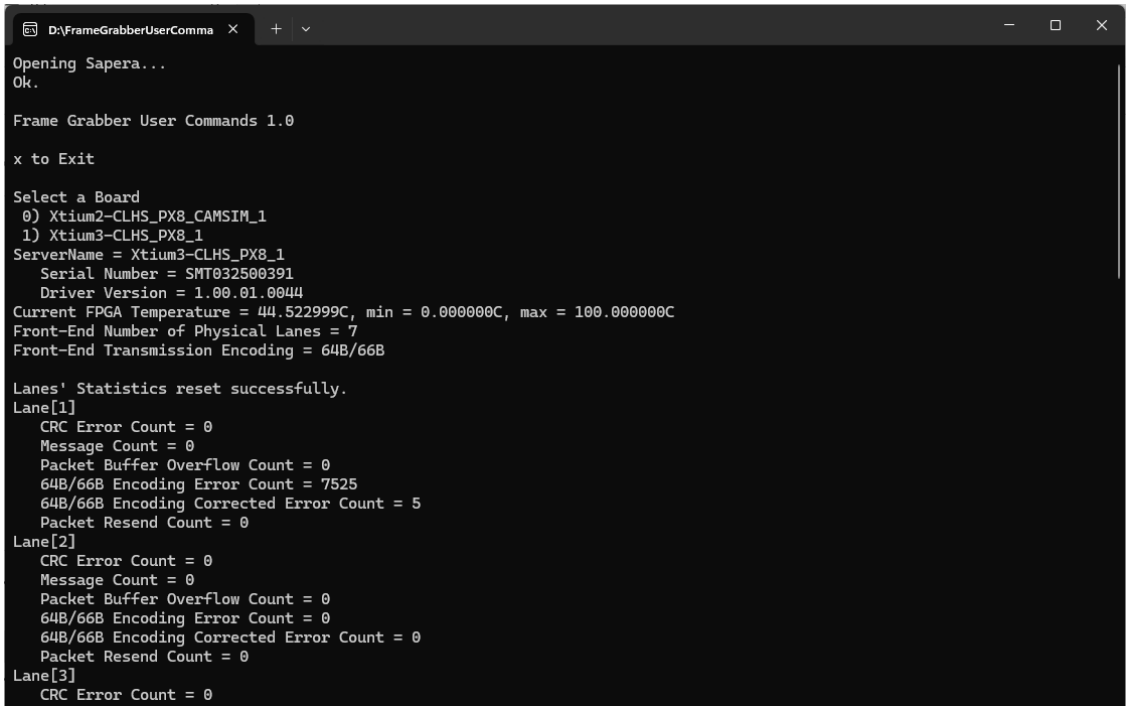
Certain features, some of them hardware-dependent, are not supported in SaperaLT. All these User Commands are accessible through the Xtium **CorManControl** function, a generic function that can send and receive variable sized data.

Demo Program

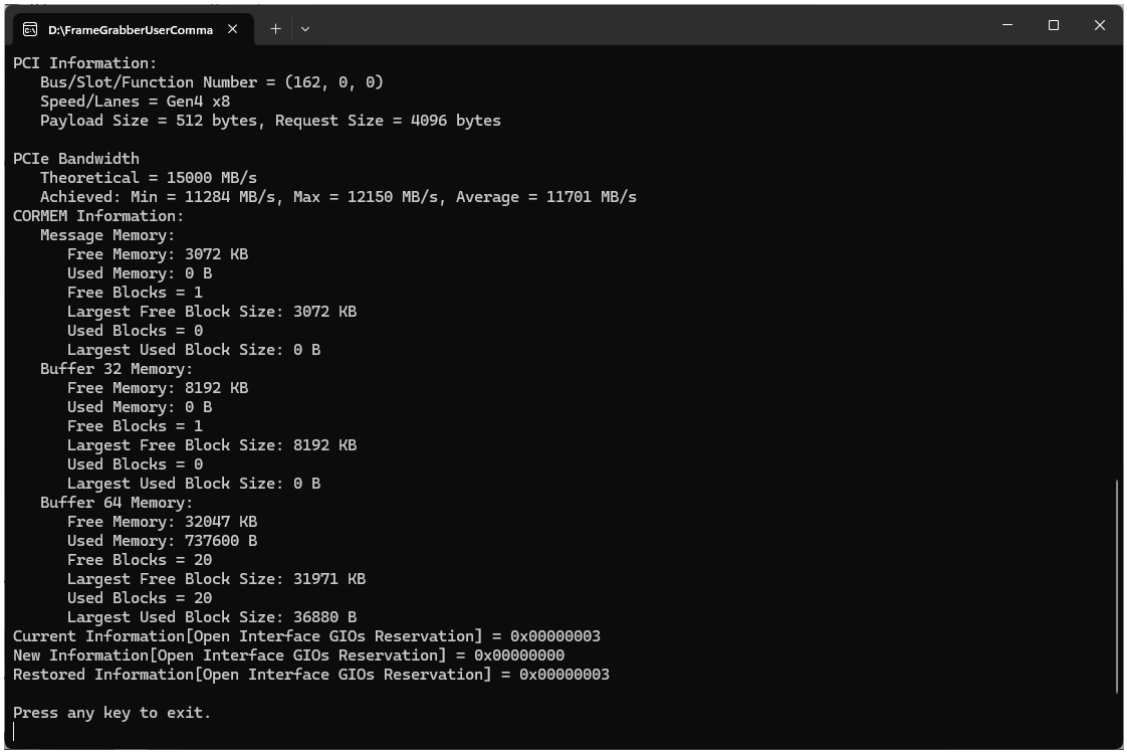
A demo program is included with the frame grabber installation, with source code (Microsoft Visual Studio 2019 solution in C) which can be modified and recompiled, that demonstrates how to use the frame grabber user commands. It is available in the following directory:



The demo is a console program which allows you to access information from frame grabbers installed on the system.



…



Generic Function

CorManControl		
Access to User Commands		
Prototype	CORSTATUS **CorManControl**(CORSERVER *hServer*, UINT32 *cmd*, void **inData*, UINT32 *inDataSize*, void **outData*, UINT32 *outDataSize*);	
Description	This generic function permits executing numerous User Commands usually not accessible through any of the SaperaLT standard functions.	
Input	*hServer*	Board server handle
	Cmd	Command to execute. See User Commands below for a description of available commands
	inData	Data passed to the command to execute. Its content depends on the command used.
	inDataSize	Size in bytes of the inData argument passed to the function
	outDataSize	Size in bytes of the outData argument passed to the function
Output	*outData*	Data returned to the application by the command executed. Its content depends on the command used
Return Value	CORSTATUS_OK if successful, otherwise a CORSTATUS error depending on the command executed.	
Header	*capiman.h*	

User Commands

- All User Commands are executed by using the function .
- Data structures and definitions of User Commands can be found in header: *manuser.h*

CORCMD_USER_DEVICE_PARAMETER_READ		
Description	This command permits reading fixed sized data from a device.	
Input	*hServer*	Board server handle
	Cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER.
	inDataSize	Size in bytes of the inData argument passed to the function
	outDataSize	Size in bytes of the outData argument passed to the function
Output	*outData*	Data returned to the application by the command executed. Its content depends on the command used
Return Value	CORSTATUS_OK if successful, otherwise a CORSTATUS error depending on the function executed.	

```
typedef struct
{
UINT32 deviceIndex; // Device to access
  UINT32 prmIndex;    // one of CORHW_DEVICE_PRM_xxx
  UINT32 prmSize;     // Always 0.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORCMD_USER_DEVICE_PARAMETER_READ_EX		
Description	This command permits reading variable sized data from a device while also passing variable extra input data.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ_EX
	inData	CORCMD_DEVICE_PARAMETER.
	inDataSize	Size in bytes of the inData argument passed to the function
	outDataSize	Size in bytes of the outData argument passed to the function
Output	*outData*	Data returned to the application by the command executed. Its content depends on the command used
Return Value	CORSTATUS_OK if successful, otherwise a CORSTATUS error depending on the function executed.	

```
typedef struct
{
UINT32 deviceIndex; // Device to access
  UINT32 prmIndex;    // one of CORHW_DEVICE_PRM_xxx
  UINT32 prmSize;     // size in bytes of the variable sized input data that follows this structure;
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORCMD_USER_DEVICE_PARAMETER_WRITE		
Description	This command permits writing variable sized data to a device.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_WRITE CORCMD_DEVICE_PARAMETER followed by custom input data depending on function selected. Size in bytes of the inData argument passed to the function 0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful, otherwise a CORSTATUS error depending on the function executed.	

```
typedef struct
{
UINT32 deviceIndex; // Device to access
  UINT32 prmIndex;    // one of CORHW_DEVICE_PRM_xxx
  UINT32 prmSize;     // size in bytes of the variable sized input data following;
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

User Functions

- Data structures and definitions of User Functions can be found in header *corhw_prm_user.h*
- Macro definitions can be found in header *aq2_prm_user.h*

These functions provide access to information and statistics about the frame grabber; these details can then be included directly in user applications for informational or troubleshooting purposes, as required. In general, these details are available through the frame grabber utility applications, such as the Device Manager and Diagnostic Tool.

Function	Description
CORHW_USER_DEVICE_PRM_GET_BANDWIDTH	Returns the bandwidth achieved by the device.
CORHW_USER_DEVICE_PRM_GET_CAMERA_PORT_READ	Reads variable sized data from a device.
CORHW_USER_DEVICE_PRM_GET_LANES_STATS	Returns front-end lane statistics from a device.
CORHW_USER_DEVICE_PRM_GET_NB_LANES	Returns the number of physical front-end lanes of the device.
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_BIT_TRANSFER_RATE	Returns the PCI transfer rate the device is running at in the system.
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NB_LANES	Returns the number of lanes the device uses on the PCI bus.
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NUMBER	Returns the PCI bus number of the device in the system.
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_PAYLOAD_SIZE	Returns the payload size the device uses on the PCI bus.
CORHW_USER_DEVICE_PRM_GET_PCI_BUS_REQUEST_SIZE	Returns the request size the device uses on the PCI bus.
CORHW_USER_DEVICE_PRM_GET_PCI_FUNCTION_NUMBER	Returns the PCI function number of the device in the system.
CORHW_USER_DEVICE_PRM_GET_PCI_SLOT_NUMBER	Returns the PCI slot number of the device in the system.
CORHW_USER_DEVICE_PRM_GET_TEMPERATURE	Returns the current temperature of a component on the device.
CORHW_USER_DEVICE_PRM_GET_THEORETICAL_BANDWIDTH	Returns the theoretical bandwidth that can be achieved by the device.
CORHW_USER_DEVICE_PRM_GET_TRANSMISSION_ENCODING	Returns the current transmission encoding (8B/10B or 64B/66B) of the device.
CORHW_USER_DEVICE_PRM_GET_USER_DEVICE_INFO_VALUE	Returns the user-defined information value stored in non-volatile memory of the device.
CORHW_USER_DEVICE_PRM_RESET_LANES_STATS	This function resets all lane statistics.
CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_CONNECT	Connects to a specified camera.
CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_DISCONNECT	Donnects from a specified camera.
CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_WRITE	Writes variable sized data to the specified camera.
CORHW_USER_DEVICE_PRM_SET_DIAGNOSTIC_MODE	Sets the diagnostic mode of the device.
CORHW_USER_DEVICE_PRM_SET_USER_DEVICE_INFO_VALUE	Writes the user-defined information value to the non-volatile memory of the device.

CORHW_USER_DEVICE_PRM_GET_BANDWIDTH		
Description	Returns the bandwidth achieved by the device by doing a data transfer from the device to the host memory by use of a DMA engine.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_READ CORCMD_DEVICE_PARAMETER sizeof(CORCMD_DEVICE_PARAMETER) UINT32
Output	*outData*	Pointer to a UINT32 to store the bandwidth achieved in MB/s.
Return Value	CORSTATUS_OK if successful	
Note	Device must be put in Diagnostic Mode prior to getting the bandwidth.	

```
{
•       UINT32 deviceIndex; // Set to 0. Not Used.
•       UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_BANDWIDTH
•       UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_CAMERA_PORT_READ		
Description	Reads variable sized data from a device.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_READ_EX CORCMD_DEVICE_PARAMETER_CAMERA_PORT_READ sizeof(CORCMD_DEVICE_PARAMETER_CAMERA_PORT_READ) Size in bytes of the outData argument passed to the function
Output	*outData*	Pointer to store the data read from the camera
Return Value	CORSTATUS_OK if successful CORSTATUS_DATA_INVALID CORSTATUS_RESOURCE_INVALID	 Camera Index is not valid Camera input is connected to a slave port, not a camera.

```
typedef struct
{
CORCMD_DEVICE_PARAMETER prm;                        // Generic parameters for the function
CORHW_DEVICE_PARAMETER_GET_CAMERA_PORT_READ data;  // Specific parameters for the function
} CORCMD_DEVICE_PARAMETER_CAMERA_PORT_READ, *PCORCMD_DEVICE_PARAMETER_CAMERA_PORT_READ;
typedef struct
{
UINT32 deviceIndex; // Camera Index to access
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_CAMERA_PORT_READ
  UINT32 prmSize;     // sizeof(CORHW_DEVICE_PARAMETER_GET_CAMERA_PORT_READ)
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
typedef struct
{
UINT64 regAddr;   // Register address to read from
UINT32 valueSize; // Size in bytes of the data to read
UINT32 bigEndian; // Endianness of the register to access.
} CORHW_DEVICE_PARAMETER_GET_CAMERA_PORT_READ, *PCORHW_DEVICE_PARAMETER_GET_CAMERA_PORT_READ;
```

CORHW_USER_DEVICE_PRM_GET_LANES_STATS		
Description	Returns the specified front-end lane statistics from a device.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_READ_EX CORCMD_DEVICE_PARAMETER_GET_LANES_STATS sizeof(CORCMD_DEVICE_PARAMETER_GET_LANES_STATS) UINT32
Output	*outData*	Pointer to a UINT32 to store the statistic.
Return Value	CORSTATUS_OK	
	CORSTATUS_DATA_INVALID	Statistic type is not valid
Remark	Possible statistic type values are: CORHW_DEVICE_PARAMETER_LANES_CRC_ERROR_COUNT Number of CRC errors detected in the packets received. CRC error occurs when the device's calculate CRC is different than the CRC embedded in the packet. For CLHS, there is a count per lane. For CXP, count is on master lane only. CORHW_DEVICE_PARAMETER_LANES_VIDEO_MSG_COUNT Number of video message packets received. For CLHS, there is count per lane. For CXP, count is on master lane only. CORHW_DEVICE_PARAMETER_LANES_PACKET_BUFFER_OVERFLOW_COUNT Number of Packet Buffer Overflow errors detected. Usually this error is related to device not being able to process the received packets in a timely manner. For CLHS, there is a count per lane. For CXP, count is on master lane only. CORHW_DEVICE_PARAMETER_LANES_PACKET_SIZE_ERROR_COUNT Number of invalid packet size detected by the device Only available on CXP Count is on master lane only. CORHW_DEVICE_PARAMETER_LANES_RESEND_FLAG_COUNT Number of resend request done by the device Only available on CLHS CORHW_DEVICE_PARAMETER_LANES_ENCODING_ERROR_COUNT Number of encoding error (ie. bit errors) detected. CORHW_DEVICE_PARAMETER_LANES_ENCODING_CORRECTED_ERROR_COUNT Number of encoding error (ie. bit errors) that was corrected by FEC (Forward Error Correction) engine. Only available with 64B/66B encoding. typedef struct	

```
{
  CORCMD_DEVICE_PARAMETER prm;   // Generic parameters for the function
  UINT32 statIndex;              // Specifies the type of statistic to retrieve
} CORCMD_DEVICE_PARAMETER_GET_LANES_STATS, *PCORCMD_DEVICE_PARAMETER_GET_LANES_STATS;
typedef struct
{
UINT32 deviceIndex; // Lane index from which to retrieve the statistic
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_LANES_STATS
  UINT32 prmSize;     // sizeof(UINT32)
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_NB_LANES		
Description	Returns the number of physical front-end lanes of the device.	
Input	*hServer*	Board server handle
	Cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the number of lanes.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_NB_LANES
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_BUS_BIT_TRANSFER_RATE		
Description	Returns the PCI transfer rate the device is running at in the system.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus bit transfer rate.
Return Value	CORSTATUS_OK if successful	
Note	Return value represents the Gen'X' of the device. For example, a value of 3 means Gen3.	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_BUS_BIT_TRANSFER_RATE
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NB_LANES		
Description	Returns the number of lanes the device uses on the PCI bus.	
Input -	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus number of lanes.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NB_LANES
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NUMBER		
Description	Returns the PCI bus number of the device in the system.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus number.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_BUS_NUMBER
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_BUS_PAYLOAD_SIZE		
Description	Returns the payload size the device uses on the PCI bus.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus payload size in bytes.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_BUS_PAYLOAD_SIZE
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_BUS_REQUEST_SIZE		
Description	Returns the request size the device uses on the PCI bus.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus request size in bytes.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_BUS_REQUEST_SIZE
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_FUNCTION_NUMBER		
Description	Returns the PCI function number of the device in the system.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus function number.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_FUNCTION_NUMBER
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_PCI_SLOT_NUMBER		
Description	Returns the PCI slot number of the device in the system.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	CORCMD_DEVICE_PARAMETER
	inDataSize	sizeof(CORCMD_DEVICE_PARAMETER)
	outDataSize	UINT32
Output	*outData*	Pointer to a UINT32 to store the bus slot number.
Return Value	CORSTATUS_OK if successful	

```
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_PCI_SLOT_NUMBER
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_TEMPERATURE		
Description	Returns the current temperature of a component on the device.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_READ
	inData	'*prm*' member of CORCMD_DEVICE_PARAMETER_GET_TEMPERATURE
	inDataSize	Size of '*prm*' member of CORCMD_DEVICE_PARAMETER_GET_TEMPERATURE
	outDataSize	Size of '*data*' member of CORCMD_DEVICE_PARAMETER_GET_TEMPERATURE
Output	*outData*	'*data*' member of CORCMD_DEVICE_PARAMETER_GET_TEMPERATURE
Return Value	CORSTATUS_OK if successful	

```
{
CORCMD_DEVICE_PARAMETER prm;                        // Generic parameters for the function
CORHW_DEVICE_PARAMETER_GET_TEMPERATURE_DATA data;  // Storage for the temperature data
} CORCMD_DEVICE_PARAMETER_GET_TEMPERATURE, *PCORCMD_DEVICE_PARAMETER_GET_TEMPERATURE

{
UINT32 deviceIndex; // Index of the component. For Xtium3 series, 0 is the main FPGA component.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_TEMPERATURE
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
{
  DOWRD min;   // Minimum temperature allowed for the component in milli C
  DWORD max;   // Maximum temperature allowed for the component in milli C
DWORD value; // Current temperature of the component in milli C
} CORHW_DEVICE_PARAMETER_GET_TEMPERATURE_DATA, *PCORHW_DEVICE_PARAMETER_GET_TEMPERATURE_DATA;
```

CORHW_USER_DEVICE_PRM_GET_THEORETICAL_BANDWIDTH		
Description	Returns the theoretical bandwidth that can be achieved by the device when doing a data transfer from the device to the host memory by use of a DMA engine.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_READ CORCMD_DEVICE_PARAMETER sizeof(CORCMD_DEVICE_PARAMETER) UINT32
Output	*outData*	Pointer to a UINT32 to store the theoretical bandwidth achieved in MB/s.
Return Value	CORSTATUS_OK if successful	

```
{
    UINT32 deviceIndex; // Set to 0. Not Used.
    UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_THEORECTICAL_BANDWIDTH
    UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_TRANSMISSION_ENCODING		
Description	Returns the current transmission encoding (8B/10B or 64B/66B) of the device.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_READ CORCMD_DEVICE_PARAMETER sizeof(CORCMD_DEVICE_PARAMETER) UINT32
Output	*outData*	Pointer to a UINT32 to store the transmission encoding of the device.
Return Value	CORSTATUS_OK if successful	
Remark	Possible transmission encoding values are: CORHW_DEVICE_PARAMETER_TRANSMISSION_ENCODING_NONE No Encoding CORHW_DEVICE_PARAMETER_TRANSMISSION_ENCODING_8B_10B 8B/10B Encoding CORHW_DEVICE_PARAMETER_TRANSMISSION_ENCODING_64B_66B 64B/66B Encoding	

```
typedef struct
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_TRANSMISSION_ENCODING
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_GET_USER_DEVICE_INFO_VALUE		
Description	Returns the user-defined information value stored in non-volatile memory of the device.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_READ_EX CORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN Size of 'CORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN Size of CORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT
Output	*outData*	Ponter to a CORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT
Return Value	CORSTATUS_OK if successful	

```
{
CORCMD_DEVICE_PARAMETER prm;                                  // Generic parameters for the function
CORHW_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN data;  // Information to read
} CORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN,
*PCORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN

{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_GET_USER_DEVICE_INFO_VALUE
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
{
CHAR field[CORHW_PRM_STRING_LENGTH];  // String representation of the information value to read.
} CORHW_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN,
*PCORHW_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_IN;

{
CORHW_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT data;  // Storage for the information value read
} CORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT,
*PCORCMD_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT

{
CHAR field[CORHW_PRM_STRING_LENGTH];  // String representation of the information value to read.
} CORHW_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT,
*PCORHW_DEVICE_PARAMETER_GET_USER_DEVICE_INFO_VALUE_OUT;
```

CORHW_USER_DEVICE_PRM_RESET_LANES_STATS		
Description	This function resets all lane statistics.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_WRITE CORCMD_DEVICE_PARAMETER sizeof(CORCMD_DEVICE_PARAMETER) 0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful	

```
typedef struct
{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_RESET_LANES_STATS
  UINT32 prmSize;     // Set to 0: Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_CONNECT		
Description	Makes a connection with the specified camera by first doing a discovery of the camera (if not already detected) and enabling communication interrupts to send/receive data to/from the camera.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_WRITE CORCMD_DEVICE_PARAMETER sizeof(CORCMD_DEVICE_PARAMETER) 0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful CORSTATUS_DATA_INVALID Camera Index is not valid. CORSTATUS_RESOURCE_INVALID Camera input is connected to a slave port, not a camera.	

```
typedef struct
{
UINT32 deviceIndex; // Camera Index to access
  UINT32 prmIndex;    // CORHW_DEVICE_PRM_SET_CAMERA_PORT_CONNECT
  UINT32 prmSize;     // Set to 0: Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_DISCONNECT		
Description	Disconnects from the specified camera by disabling communication interrupts to send/receive data to/from the camera.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_WRITE CORCMD_DEVICE_PARAMETER sizeof(CORCMD_DEVICE_PARAMETER) 0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful CORSTATUS_DATA_INVALID Camera Index is not valid. CORSTATUS_RESOURCE_INVALID Camera input is connected to a slave port, not a camera.	

```
typedef struct
{
UINT32 deviceIndex; // Camera Index to access
  UINT32 prmIndex;    // CORHW_DEVICE_PRM_SET_CAMERA_PORT_DISCONNECT
  UINT32 prmSize;     // Set to 0: Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
```

CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_WRITE		
Description	Writes variable sized data to the specified camera.	
Input	*hServer* *cmd* *inData* *inDataSize* *outDataSize*	Board server handle CORCMD_USER_DEVICE_PARAMETER_WRITE CORCMD_DEVICE_PARAMETER_CAMERA_PORT_WRITE sizeof(CORCMD_DEVICE_PARAMETER_SET_CAMERA_PORT_WRITE) 0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful CORSTATUS_DATA_INVALID Camera Index is not valid. CORSTATUS_RESOURCE_INVALID Camera input is connected to a slave port, not a camera.	
Remark	For CXP cameras, the access to the registers is always big endian.	

```
typedef struct
{
CORCMD_DEVICE_PARAMETER prm;                          // Generic parameters for the function
CORHW_DEVICE_PARAMETER_SET_CAMERA_PORT_WRITE data;  // Specific parameters for the function
} CORCMD_DEVICE_PARAMETER_CAMERA_PORT_WRITE, *PCORCMD_DEVICE_PARAMETER_CAMERA_PORT_WRITE;
typedef struct
{
UINT32 deviceIndex; // Camera Index to access
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_SET_CAMERA_PORT_WRITE
  UINT32 prmSize;     // sizeof(CORHW_DEVICE_PARAMETER_SET_CAMERA_PORT_WRITE)
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
typedef struct
{
  UINT64 regAddr;   // Register address to write to
  void *pValue;     // Pointer to the data to write to the register
UINT32 valueSize; // Size in bytes of the data to write to the specified camera
UINT32 bigEndian; // Endianness of the register to access.
} CORHW_DEVICE_PARAMETER_SET_CAMERA_PORT_WRITE, *PCORHW_DEVICE_PARAMETER_SET_CAMERA_PORT_WRITE;
```

CORHW_USER_DEVICE_PRM_SET_DIAGNOSTIC_MODE		
Description	Sets the diagnostic mode of the device.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_WRITE
	inData	CORCMD_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE
	inDataSize	sizeof(CORHW_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE_DATA)
	outDataSize	0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful CORSTATUS_RESOURCE_IN_USE if a required resource is not available.	
Note	When a User Command makes use of resources that cannot be shared with a SaperaLT application, one must set the device in Diagnostic Mode before executing the User Command. For example, CORHW_USER_DEVICE_PRM_GET_BANDWIDTH cannot be used if Sapera LT has a transfer connected. Otherwise, CORSTATUS_RESOURCE_IN_USE is returned.	

```
typedef struct
{
CORCMD_DEVICE_PARAMETER prm;                          // Generic parameters for the function
CORHW_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE data;  // Specific parameters for the function
} CORCMD_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE, *PCORCMD_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE;
typedef struct
{
UINT32 deviceIndex; // Set to 0. Not Used
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_SET_DIAGNOSTIC_MODE
  UINT32 prmSize;     // sizeof(CORHW_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE_DATA)
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
typedef struct
{
UINT32 mode;   // 0 = Disable, 1 = Enable
} CORHW_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE_DATA, *PCORHW_DEVICE_PARAMETER_SET_DIAGNOSTIC_MODE_DATA;
```

CORHW_USER_DEVICE_PRM_SET_USER_DEVICE_INFO_VALUE		
Description	Writes the user-defined information value to the non-volatile memory of the device.	
Input	*hServer*	Board server handle
	cmd	CORCMD_USER_DEVICE_PARAMETER_WRITE
	inData	CORCMD_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE
	inDataSize	Size of 'CORCMD_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE
	outDataSize	0
Output	*outData*	NULL
Return Value	CORSTATUS_OK if successful	

```
{
CORCMD_DEVICE_PARAMETER prm;                                // Generic parameters for the function
CORHW_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE data;  // Storage for the information to write
} CORCMD_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE,
*PCORCMD_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE

{
UINT32 deviceIndex; // Set to 0. Not Used.
  UINT32 prmIndex;    // CORHW_USER_DEVICE_PRM_SET_USER_DEVICE_INFO_VALUE
  UINT32 prmSize;     // Set to 0. Not Used.
} CORCMD_DEVICE_PARAMETER, *PCORCMD_DEVICE_PARAMETER;
{
CHAR field[CORHW_PRM_STRING_LENGTH];  // String representation of the information value to write.
CHAR value[CORHW_PRM_STRING_LENGTH];  // String representation of the value to write.
} CORHW_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE,
*PCORHW_DEVICE_PARAMETER_SET_USER_DEVICE_INFO_VALUE;
```

Appendix E: Advanced Processing Licences

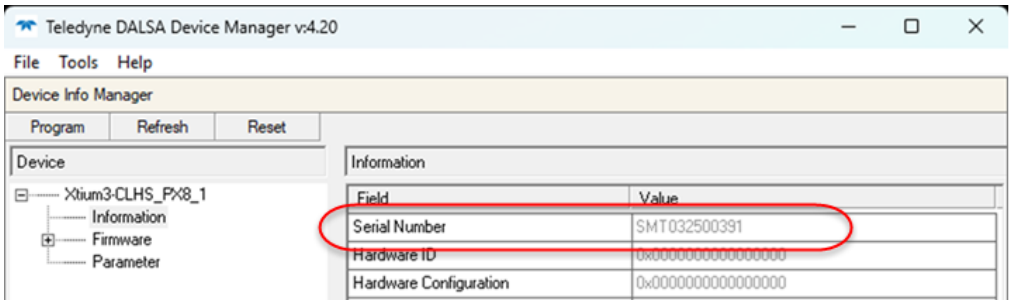
Advanced Processing features, such as 4-plane processing require a valid license, which can be enabled using the License Manager included with Sapera LT.

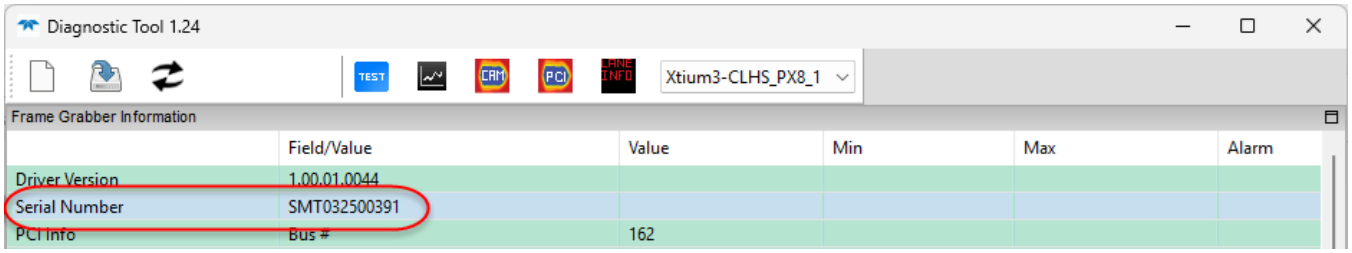
When multiple Xtium3 boards are installed in the same computer only one license is required to enable advanced processing on all boards.

NOTE
Advanced Processing features require Sapera LT Version 9.10. or greater.

*Licenses are distributed as *.lic files and can be loaded using the Sapera License Manager utility.*

To obtain a license, the serial number of the board is required. This can be retreived using the Device Manager utility or Diagnostic Tool.



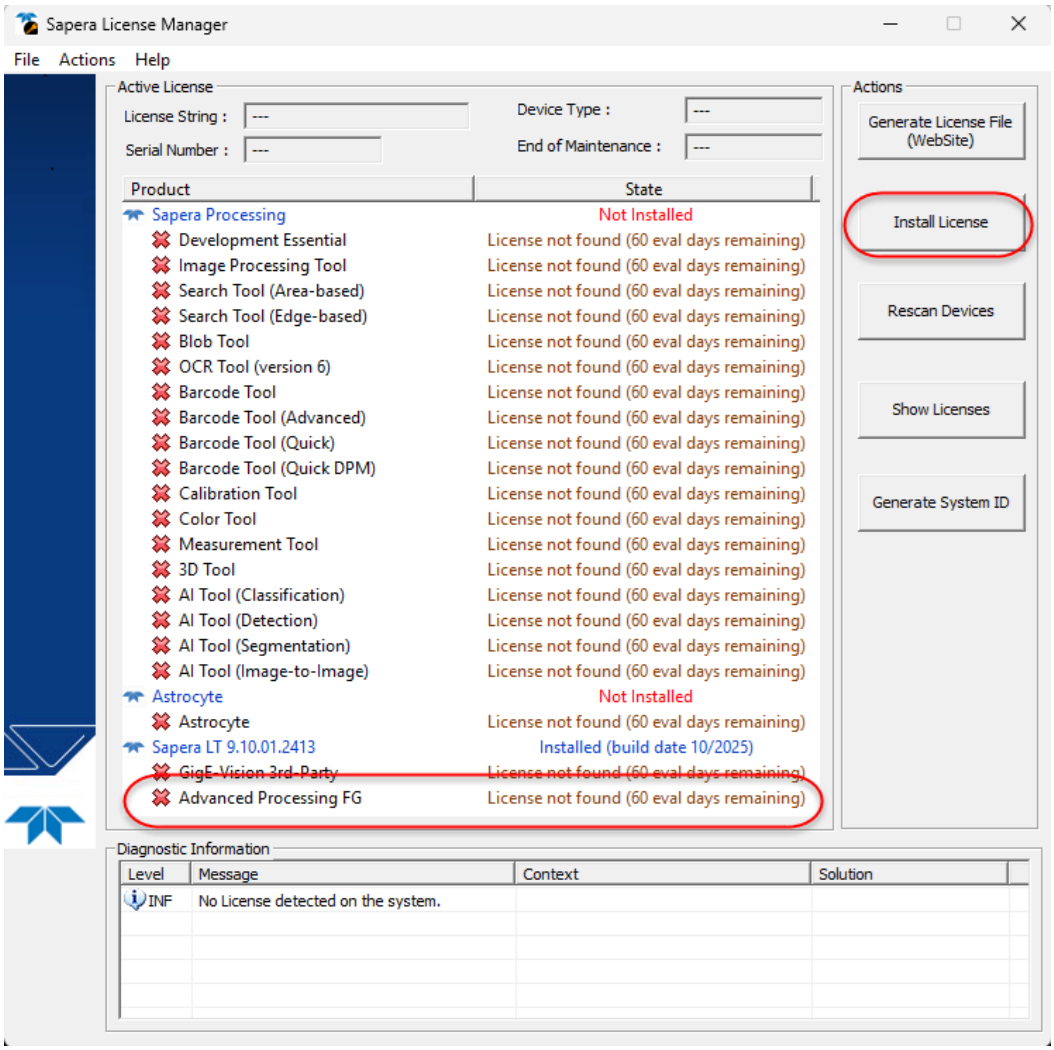


For information on purchasing Advanced Processing licenses contact Teledyne sales.

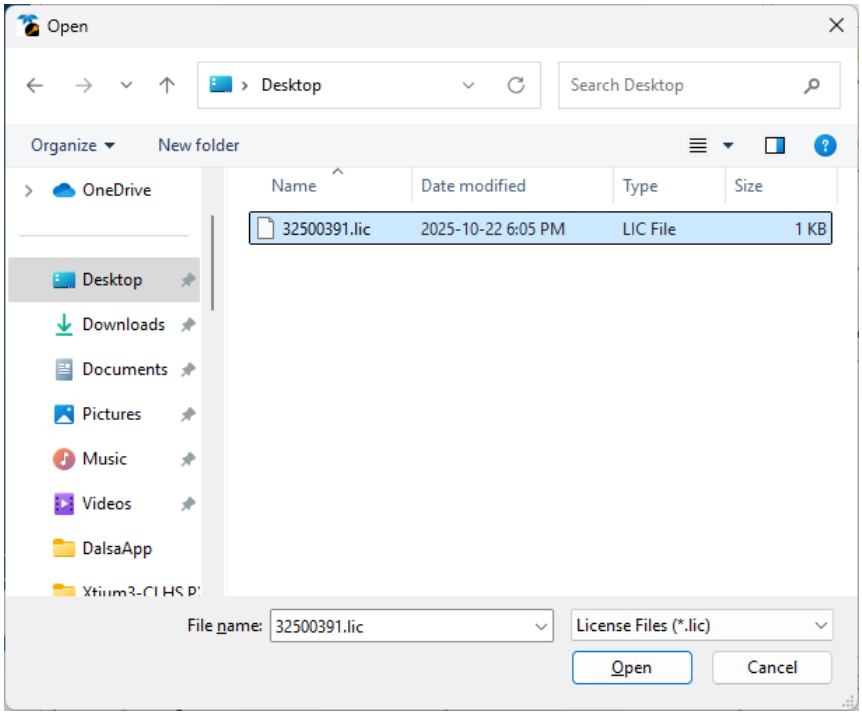
Installing an Advanced Processing License

To install an Advanced Processing FG license on a machine with an Xtium3 frame grabber installed, launch the Sapera License Manager (available…).

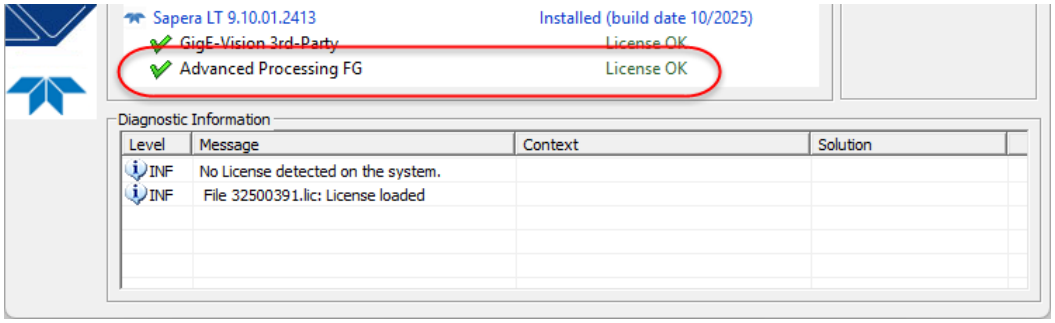
The Sapera License Manager will show that an Advanced Processing FG license is not found.



Click **Install License** and select the **.lic* license file generated for the specific Xtium3 frame grabber (by serial number) in the system.



The Advanced Processing FG field will be displayed in green and read “License OK”.



Contact Information

Sales Information

Visit our web site:	https://www.teledynevisionsolutions.com/
Email:	info@teledynevisionsolutions.com

Canadian Sales	**Canadian Sales**
Teledyne DALSA — Head office 605 McMurray Road Waterloo, Ontario N2V 2E9 Canada Tel: +1 519-886-6000 Fax: +1 519-886-8023	Teledyne DALSA — Montreal office 880 Rue McCaffrey Saint-Laurent, Quebec H4T 2C7 Canada Tel: +1 514-333-1301 Fax: +1 514-333-1388
USA Sales	**European Sales**
Teledyne DALSA — Billerica office 700 Technology Park Drive Billerica, MA 01821 USA Tel: +1 978-670-2000 Fax: +1 978-670-2010 sales.americas@teledynedalsa.com	Teledyne DALSA GMBH Lise-Meitner-Str. 7 82152 Krailling (Munich) Germany Tel: +49 89-89545730 Fax: +49 89-895457346 sales.europe@teledynedalsa.com
Asia Pacific Sales	**China Sales**
Teledyne DALSA Asia Pacific Ikebukuro East 6F 3-4-3 Higashi Ikebukuro, Toshima-ku Tokyo, 170-0013 Japan Tel: +81 3-5960-6353 Fax: +81 3-5960-6354 sales.asia@teledynedalsa.com	Teledyne DALSA China Room 904, Block C, Poly West Bund Center 275 Rui Ping Road Shanghai 200032 China Tel: +86 21-60131571 sales.asia@teledynedalsa.com

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**	https://www.teledynevisionsolutions.com/support/support-center/
Camera/sensor support information	
Product literature and driver updates	