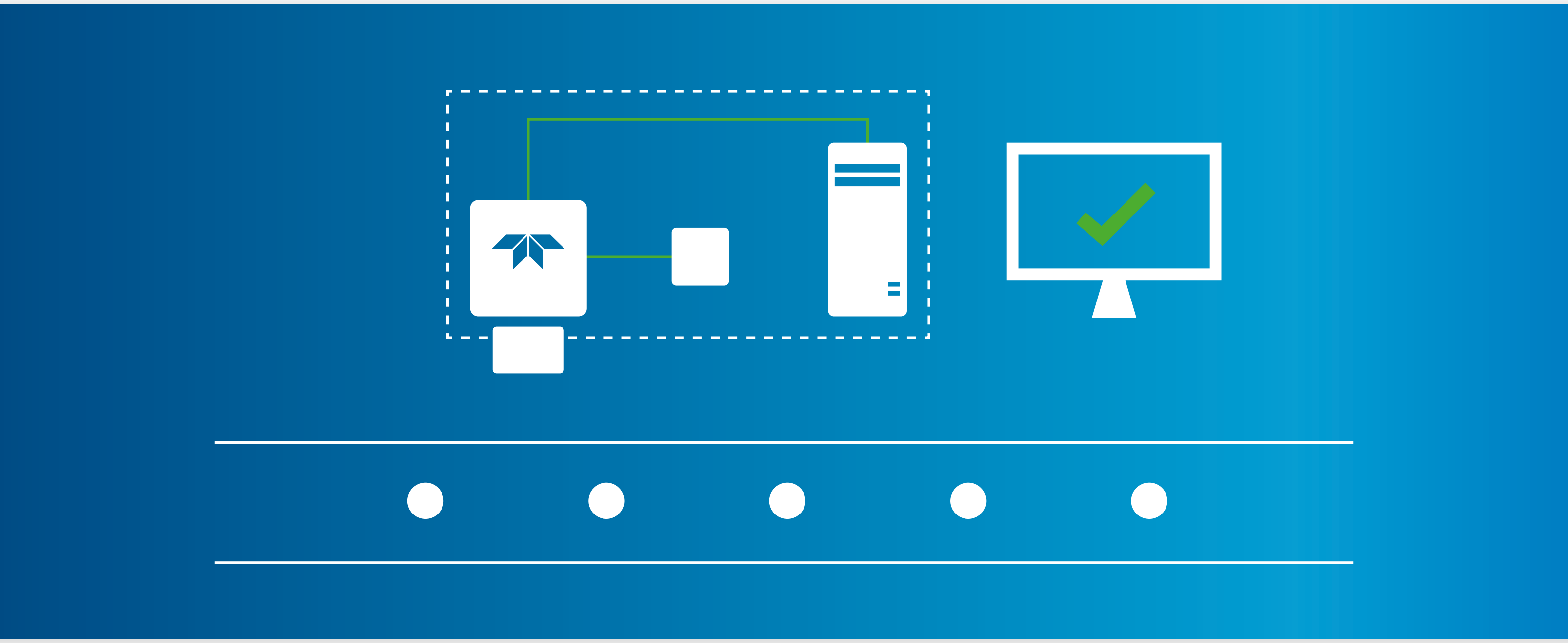
TELEDYNE DALSA
Everywhereyoulook™

TRIGGER-TO-IMAGE
RELIABILITY
BY TELEDYNE DALSA

Application Note | Case Study | Technology Primer | White Paper

Trigger-To-Image Reliability Framework

*"Without **reliability**, a machine vision system is just **liability**. T2IR provides tools to control, monitor, audit, and debug the flow of data in machine vision systems, bringing visibility to what are all too often black boxes. The ultimate results? More reliable operation, more uptime, more productivity, and lower costs."*



CONTENTS

The Challenges of Reliability in Machine Vision

Machine vision systems are used for a wide range of inspection, recognition and guidance applications in different types of manufacturing and process industries. Systems use a variety of components including lighting, triggers, cameras (area and line scan, color and monochrome), frame grabbers, and software. While the systems and inspection subjects vary, the goal is the same: to to perform autonomous inspection tasks as quickly and comprehensively as possible, with the minimum of disruptions and errors. Manufacturers depend on their vision systems to produce reliable results. They need to believe what their systems show them.

But the truth is that a lot can go wrong in a complex vision system. One of the great challenges of building a reliable system is in dealing with errors--finding out not only what went wrong, but when, how, and most importantly, why. Without careful forethought, machine vision systems can end up resembling black boxes—light goes in and data comes out, but in between, it is hard to tell what is happening.

So what are system designers to do about the black box effect? With effort, they could craft their own customized flags and indicators to assist the debugging that every system needs…or, with efficiency in mind, they could leverage Teledyne DALSA's Trigger to Image Reliability framework, a robust collection of hardware and software features created specifically to open up the black box and give system builders the tools they need to control, monitor, audit, and debug the flow of data all the way from image capture to host memory.

What is Trigger-to-Image Reliability (T2IR)?

T2IR is a combination of hardware and software features that work together to improve the reliability of your vision system. T2IR delivers full system level monitoring, control, and diagnostics capability. It lets you reach inside your vision system to audit and debug image flow. It can even store images temporarily in its components' onboard memory to overcome unexpected transfer bottlenecks and avoid data loss. That means no lost data, no false data and a clear source to identify and track any errors. T2IR ensures robust and reliable operations to produce repeatable results, and it does so in a non-intrusive manner that does not interfere with other system applications.

The Elements of the Trigger-to-Image Reliability Framework

T2IR framework capabilities are available in three principal ways:

- **Visual indicators:** As part of T2IR all Teledyne DALSA hardware incorporates LEDs to indicate device status from the time the system powers up through all operating and error modes.
- **Standalone GUI based tools:** These easy-to-use applications enable advanced diagnostics that can run concurrently with and independently of other host system applications without performance impact.
- **Sapera API programming functions:** Available for integration in user applications, these functions allow for dynamic and programmatic inline tracking, tracing and control. All the functionality of the T2IR GUI tools is available as part of the Sapera API. Users access this functionality directly from their own application using the Sapera SDK.

The following table summarizes the key benefits offered by various functional elements of T2IR framework:

T2IR ELEMENTS	BENEFIT
Acquiring the Right Target	Ensures synchronization between image acquisition and object motion to acquire the images necessary to make correct decisions.
Managing External Triggers	Handles expected and unexpected external events and provides control responses to reduced image artifacts due to motion
Tracking and Tracing Images	Provides continuous coverage of the entire image flow to deliver confidence; greatly aids debugging, reducing waste and improving up time.
Monitoring the Acquisition and Transfer Process	Enables preventive action if resource usage exceeds a predetermined threshold, selectively keeping or discarding images to sustain processing speed.
Overcoming Too Much Data	Handles peak loads to avoid data loss, ensure smooth operations.
Ensuring Data Quality	Helps increase uptime and reduce waste.
Advanced Diagnostics	Rapid pinpointing of errors for speedy diagnostic and preventive actions.

Acquiring the Right Target

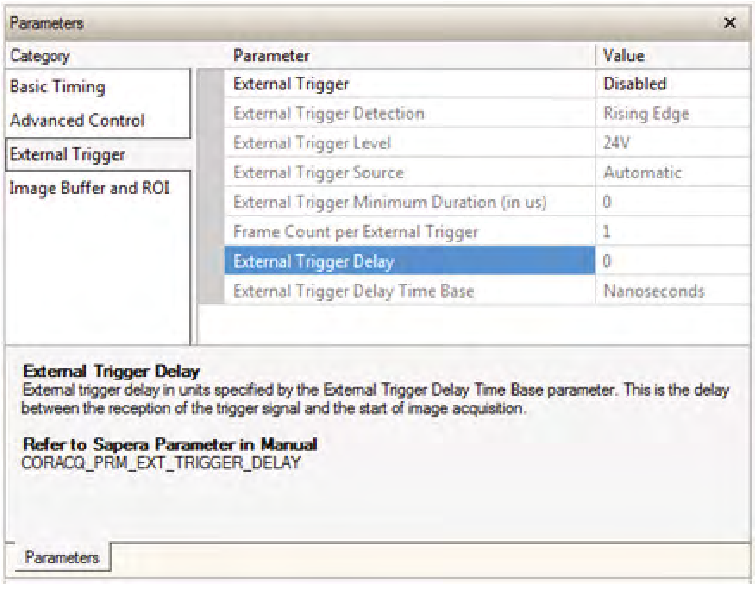
Image Acquisition Control with support for Programmable Delay Timers on Strobe and Trigger signals

The process of image acquisition for machine vision applications begins by sending a trigger signal to the camera to start generating images. A trigger represents a crucial starting point of the image acquisition sequence to enable image processing and analysis for decisions further down the inspection process. In concept this is simple enough: a trigger input generates a strobe output for lighting control and camera exposure. However, there are circumstances in which a delay between the trigger input and the strobe output is required; for example, if the camera and lighting units are not in the same position on a conveyor as the trigger sensor. Coordinating these two events through software is almost impossible and certainly not reliable (especially given the variations in command execution of the Windows operating system). To solve this problem Teledyne DALSA integrated trigger and strobe control functions into onboard hardware and incorporated programmable delay timers between these two signals.

The delay timers give developers a mechanism for establishing a precise delay between the trigger input and firing of the lighting and camera exposure. However, this amount of programmed delay is calculated based on the theoretical speed of the production line. If the actual speed is not constant (a common occurrence), the position of the object in the resulting image may not be suitable for analysis. Therefore, for reliable image acquisition the delay has to be linked to the speed of the object. This is done using the pulse output from an encoder attached to a rotating part of the conveyor system. Expressing the delay in terms of encoder ticks synchronizes it with the actual speed of the production line. As a result, the object is always at the same location in the image regardless of the speed of production line.

The easiest way to program trigger parameters is to use Sapera CamExpert. Sapera CamExpert is a camera configuration tool that offers intuitive graphical user interface and live image display for faster camera setup. CamExpert works with all Teledyne DALSA frame grabbers, GigE Vision and GenCP compliant cameras.

For example, the External Trigger parameters are all grouped in one category in the Parameters panel (shown here for the Xtium-CL PX4 frame grabber):



Parameter	Value
External Trigger	Disabled
External Trigger Detection	Rising Edge
External Trigger Level	24V
External Trigger Source	Automatic
External Trigger Minimum Duration (in us)	0
Frame Count per External Trigger	1
External Trigger Delay	0
External Trigger Delay Time Base	Nanoseconds

External Trigger Delay
External trigger delay in units specified by the External Trigger Delay Time Base parameter. This is the delay between the reception of the trigger signal and the start of image acquisition.

Refer to Sapera Parameter in Manual
CORACQ_PRM_EXT_TRIGGER_DELAY

Controlling external triggers with Sapera CamExpert

When you are satisfied with all the parameters settings these parameters can be saved in a configuration file and later retrieved by the application at run time.

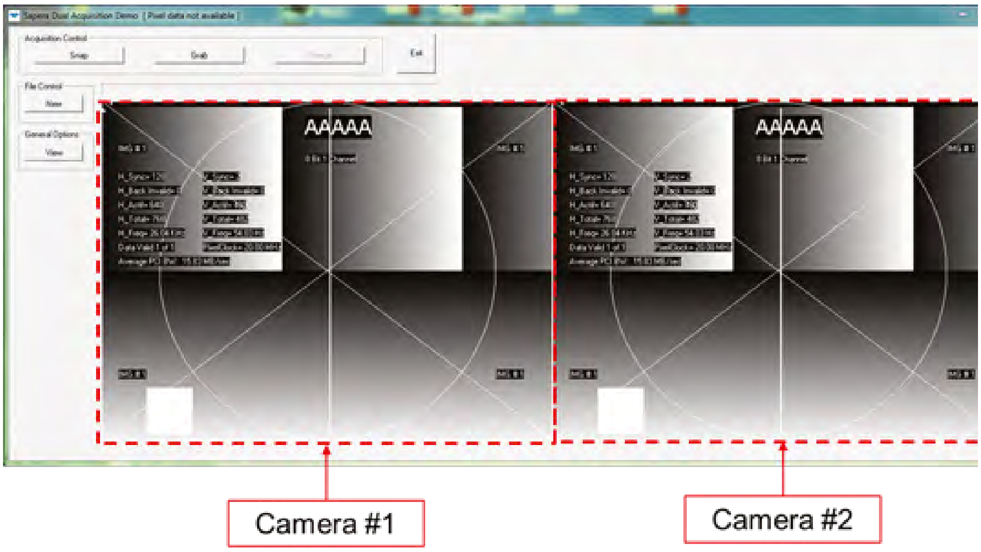
Proper synchronization achieves the first goal of Trigger-to-Image Reliability: the camera is properly controlled to capture the image of the target being inspected. These hardware features are under software control, but once initialized they act independently of any software execution, leading to predictable results.

Synchronizing Multiple Cameras or Computers

System designers often want to build systems that offer scalable performance while minimizing costs. In some cases it might be more economical to combine multiple lower resolution cameras and optics to construct higher resolution images while in some other it might be necessary to distribute very high speed images across multiple computers to minimize image processing and analysis.

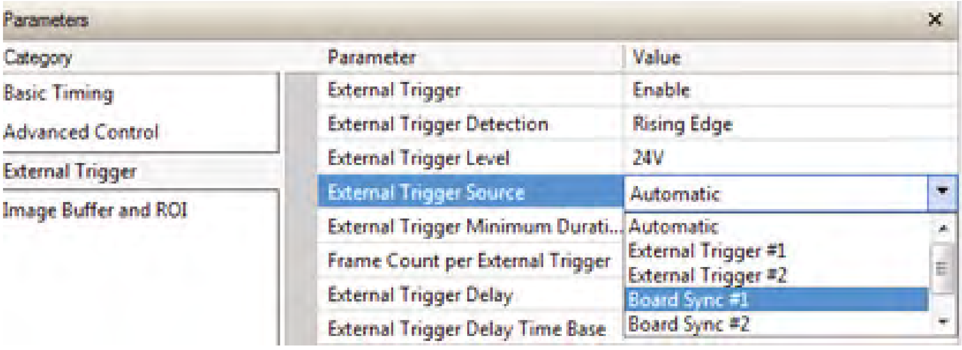
In all cases when multiple acquisition devices are used, it is important that all devices operate synchronously to produce images that are error free and ready to use. T2IR framework capabilities permit this by incorporating critical features to achieve image acquisition synchronization in hardware and software, without the need for external synchronization and data replicating devices. This T2IR synchronization feature also permits implementation of different image processing setups to achieve a target processing time.
Let us closely look at some of the commonly used system configurations.

For example, one application can combine images from two cameras in one buffer (fig A – "Multi-Board Sync." demo screenshot) or split the image from one camera across multiple frame grabbers (fig B) to overcome processing bandwidth limitations.



Multi-Board Sync

In cases where images from different devices must be combined in one buffer, Teledyne DALSA GigE Vision cameras, (such as Genie Nano and Linea GigE) and frame grabbers (such as Xtium-CL MX4) incorporate the necessary hardware to work under Sapera LT to capture images in one seamless Sapera buffer. Teledyne DALSA's Xcelera and Xtium series frame grabbers, for example, offer dedicated hardware signals to synchronize multiple boards and cameras together. The trigger source can be easily set using the CamExpert GUI. Sapera LT SDK also provides dedicated demo applications with source code to jump start the development efforts.



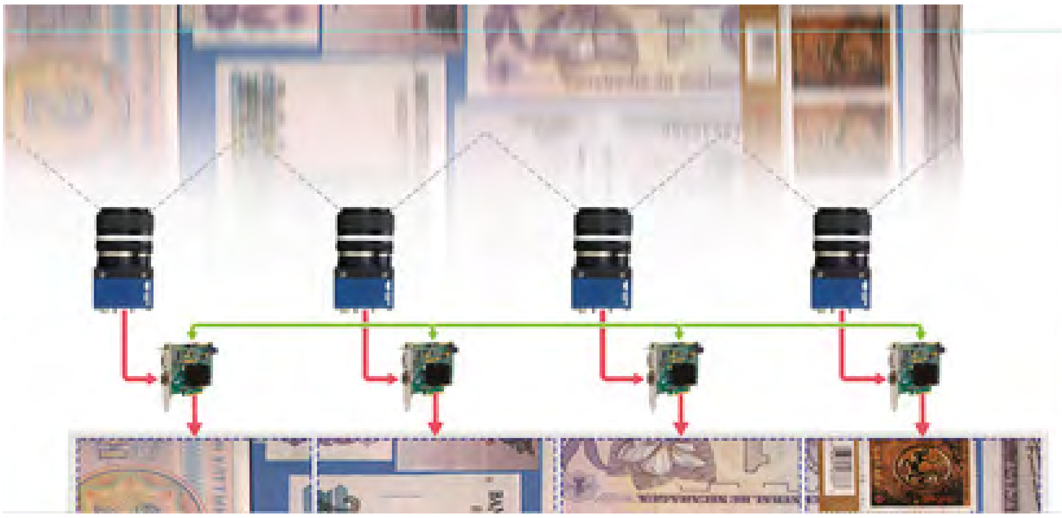
Parameters

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

Parameter	Value
External Trigger	Enable
External Trigger Detection	Rising Edge
External Trigger Level	24V
External Trigger Source	Automatic
External Trigger Minimum Durati...	Automatic
Frame Count per External Trigger	External Trigger #1
External Trigger Delay	External Trigger #2 Board Sync #1
External Trigger Delay Time Base	Board Sync #2

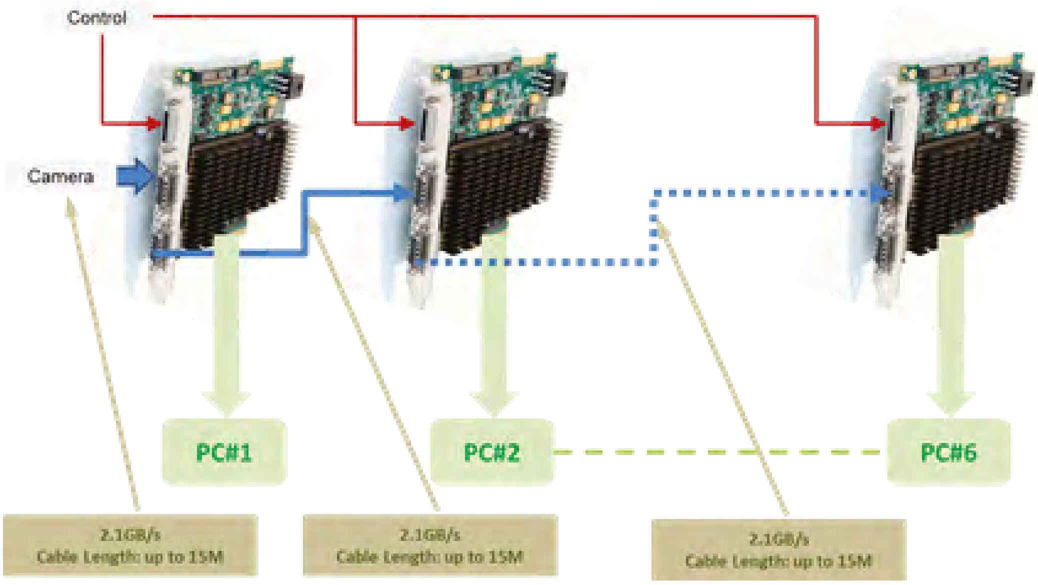
Setting External Trigger Source

Similarly, Genie Nano and Linea GigE cameras series are also capable of accepting external input signals that can be distributed to other cameras for synchronization.



Multi-Camera Synchronization

The Xtium-CLHS series also includes a dedicated image data forwarding port. This allows image processing to be distributed across multiple computers or capture images in the same PC using two frame grabbers when camera bandwidth exceeds the 2.1GB/s limit of the CLHS cable.



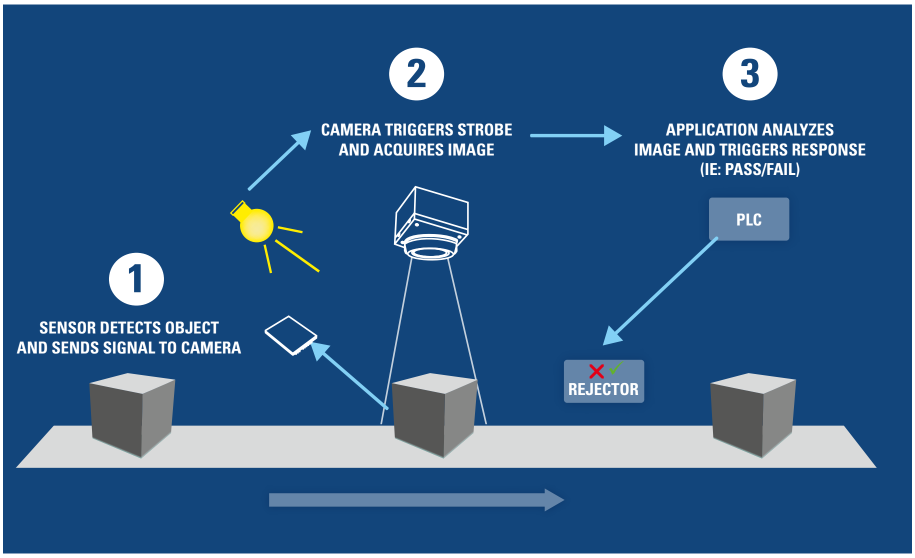
Data forwarding to distribute image processing across multiple boards

Managing External Triggers

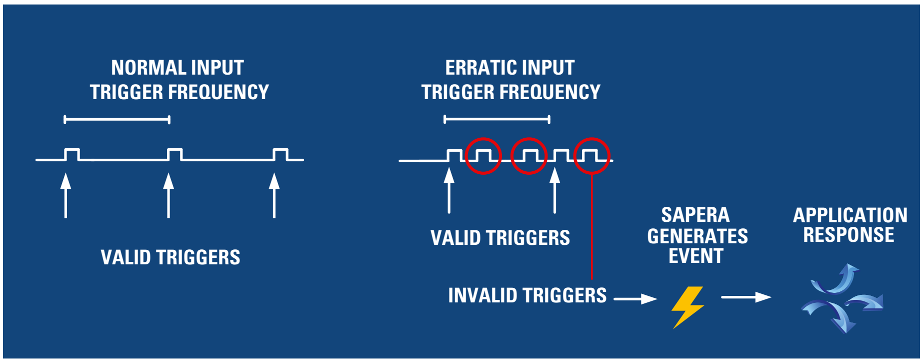
Detecting Valid as well as Invalid Triggers

External trigger management involves functions and capabilities that are essential to ensure reliability of the trigger signals seen by the vision system. It involves managing situations when the system receives too many triggers for it to properly handle and situations when the signal does not truly represent a trigger event. Let us see how T2IR handles both predictable and unpredictable triggers to ensure reliability of a vision system.

A first criterion for a valid trigger is that a trigger has to represent an actual "part-in place" for inspection. A false trigger is a signal that is not associated with a part in place. False triggers can be caused by jitter resulting from electrical noise or glitches associated with mechanical actuators and motors. T2IR capabilities offer an effective way to reduce faulty triggers by ensuring that the signal remains active for a minimum duration before it can be considered as valid for the acquisition. For added flexibility Teledyne DALSA products offer this T2IR feature as a user programmable parameter.


2
CAMERA TRIGGERS STROBE AND ACQUIRES IMAGE
3
APPLICATION ANALYZES IMAGE AND TRIGGERS RESPONSE (IE: PASS/FAIL)
PLC
1
SENSOR DETECTS OBJECT AND SENDS SIGNAL TO CAMERA
REJECTOR

Typical Imaging Application



Valid and Invalid Triggers

After the probability of spurious triggers is minimized, user applications can be programmed to handle the other extreme, appropriately called "over trigger" conditions. An over-trigger condition occurs when the camera receives a trigger but is busy acquiring previous image. Care must be given to the fact that, in some cases, sending a trigger while grabbing the previous line or frame is desirable to minimize the dead time between frames or lines (in case of line scan cameras).

Typical causes for an over-trigger state can be that the image generated from the previous trigger is still being processed, or the sensor is currently being read out or exposed for the next image (some cameras support exposing the sensor during readout, which allows for a higher frame rate than otherwise possible).

The T2IR capabilities allow applications to tolerate over-trigger situations and track them if a system starts to lose images. When frames are lost, T2IR capabilities notify Sapera based user applications with event messages for remedial actions. T2IR framework helps applications to maintain control despite timing fluctuations in trigger generation.

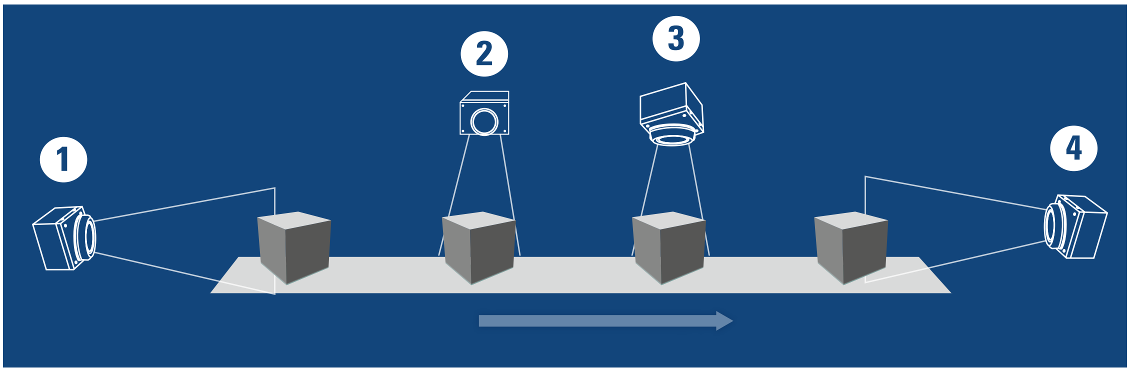
Tracking and Tracing Images

T2IR framework supports Image Tagging

While we have progressed in our discussion from the point of detecting our targets to triggering strobe lights and camera acquisition when the target is in the right location and reading the correct image data from the sensor, this is only the start of designing a reliable machine vision system. Another major issue is coordinating the collection of image data and correlating these images with physical objects moving through a material-handling system.

Trigger-to-Image Reliability uses an important design concept to assist engineers in creating reliable and repeatable systems: image tagging or timestamps. To illustrate, let us use an example of a material-handling unit processing up to 3,600 parts per minute (ppm). For factory production lines to work at maximum speed and each image must be tagged such that the downstream decision to keep, discard or re-inspect is carried out on the correct object.

More advanced applications may require inspection from multiple views. Continuing our previous example, let us assume the object has to be inspected on each side, each with different lighting, at the same frame rate. Now the constraints evolve from inspecting 3600 parts per minute to handling 14,400 images per minute. In this scenario, the imaging system must correlate four different acquisitions before making the final decision to accept, reject or re-inspect the object.



With synchronized acquisition timestamps, the 4 images for each item are:

Image 1 timestamp = ①

Image 2 timestamp = ① + X ticks = ②

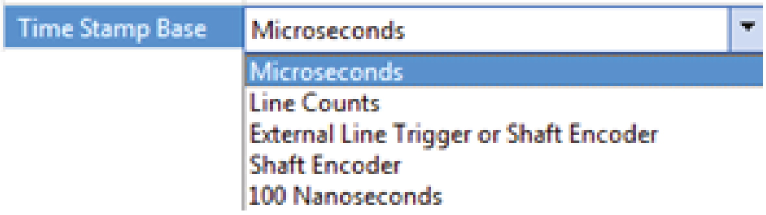
Image 3 timestamp = ① + y ticks = ③

Image 4 timestamp = ① + z ticks = ④

Where x, y and z are the expected intervals between acquisitions.

Tracking and Tracing

The image tags (timestamps) are generated either from an onboard hardware clock, the PC clock or increments using an external signal, be it a trigger, encoder tick or another pulse input at the time of image acquisition and/or image transfer to the host. For example, the Xtium-CL MX4 provides the following hardware timestamps:



Time Stamp Base: Microseconds

T2IR allows images to be timestamped from various different sources

The acquisition frame start timestamps from the device and host (issued when it starts to receive the frame) are saved in Sapera buffers with the images. These timestamps can be retrieved by the host applications using Sapera functions for analysis.

Since there is a time lag between image capture and analysis, the image timestamps can be used to ensure that the system acts on the correct object. Timestamps can also be used to precisely measure the acquisition or processing rates. It can also be used to determine if any loss of data has occurred by comparing the time lapse between successive frames.

In C++, callback functions are used to access the timestamps that T2IR creates for events and images; whenever registered events occur, the associated callback function is executed.

```
        //Register acquisition events
        success = pAcqDevice->RegisterCallback("FrameStart",
MyAcquisitionCallback, pBuffer)
        …
        //Callback function for events
        void MyAcquisitionCallback(SapAcqDeviceCallbackInfo
        *pInfo)
        {
        …
        pInfo->GetAuxiliaryTimeStamp(&myAuxTimestampValue);
        pInfo->GetHostTimeStamp(&myHostTimestampValue);
        …
        }
```

For .NET, a similar mechanism uses the EnableEvent method and

AcqDeviceNotify event to call the associated event handler.

```
        //Enable acquisition event
        device.EnableEvent("FrameStart");
        …
        //Create event handler to execute callback for enabled
events
        device.AcqDeviceNotify += new SapAcqDeviceNotifyHandl
er(AcqDeviceCallback);
        …
        //Callback function for events
        static void AcqDeviceCallback(Object sender,
SapAcqDeviceNotifyEventArgs args)
        {
        myVariableA = args.AuxTimeStamp);
        myVariableB =  args.HostTimeStamp;
```

Accessing timestamps in C++ and .Net

Monitoring the Acquisition Process

Sapera Acquisition and Transfer Events, Windows Callback function and Status Monitoring of Key Acquisition Signals

When the machine vision system is capturing the right data and tracking objects throughout the cycle for acceptance or rejection, it is now time to transfer the image data from the acquisition device onboard memory to system host memory. Trigger-to-Image Reliability framework includes a set of software tools to ensure that all required images were captured accurately into acquisition device onboard memory. While it is possible to continuously check the status to monitor system operations, in practice it comes at the expense of system performance. T2IR uses the concept of events that are issued by the acquisition devices to notify the application if certain status flags have changed. This allows applications to operate more optimally as it gets interrupted from its main processing task only when an event has occurred. Since these notifications are handled at the user application level, the applications have complete freedom to decide how best to handle them. The table below summarizes the Sapera events associated with image capture and transfer sequences into the host memory.

EVENT	DESCRIPTION
EndOfEven	End of even field
EndOfField	End of field (odd or even)
EndOfFrame	End of frame
EndOfLine	After a specific line number eventType = EndOfLine | lineNum
EndOfNLines	After a specific line number (linescan cameras only) eventType = EndOfNLines | numLines
EndOfOdd	End of odd field
EndOfTransfer	End of transfer, that is, after all frames have been transferred following calls to SapTransfer.Snap or SapTransfer.Grab/SapTransfer.Freeze.
FieldUnderrun	The number of active lines per field received from a video source is less than it should be.
LineUnderrun	The number of active pixels per line received from a video source is less than it should be.
StartOfEven	Start of even field
StartOfField	Start of field (odd or even)
StartOfFrame	Start of frame
StartOfOdd	Start of odd field

In addition to these events the status of the following acquisition signals can be monitored in the host application.
Note that the availability of status signals varies with the hardware used and this availability can be verified programmatically.
The SapAcquisition::GetSignalStatus(…..) function can be used to monitor these signals.

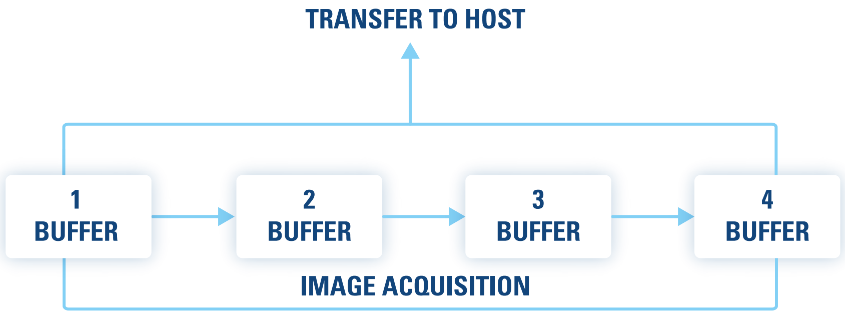
STATUS SIGNAL TO INQUIRE

SapAcquisition::SignalHSyncPresent	No signal
SapAcquisition::SignalVSyncPresent	Horizontal sync signal (analog video source) or line valid (digital video source)
SapAcquisition::SignalPixelClkPresent /	Vertical sync signal (analog video source) or frame valid (digital video source)
SapAcquisition::SignalPixelClk1Present	Pixel clock signal. For CameraLink devices, this status returns true if a clock signal is detected on the base cable.
SapAcquisition::SignalPixelClk2Present	Pixel clock signal. For CameraLink devices, this status returns true if a clock signal is detected on the medium cable.
SapAcquisition::SignalPixelClk3Present	Pixel clock signal. For CameraLink devices, this status returns true if a clock signal is detected on the full cable.
SapAcquisition::SignalPixelClkAllPresent	Pixel clock signal. For Camera Link devices, true if all required pixel clock signals have been detected by the acquisition device based on the CameraLink configuration selected.
SapAcquisition::SignalChromaPresent	Color burst signal (valid for NTSC and PAL)
SapAcquisition::SignalHSyncLock	Successful lock to an horizontal sync signal, for an analog video source
SapAcquisition::SignalVSyncLock	Successful lock to a vertical sync signal, for an analog video source

Overcoming Too Much Data

Tracks Occurrences of Trashed Frames

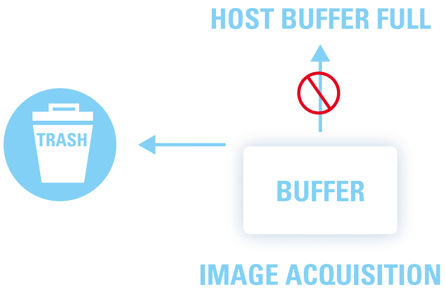
Let us build on our previous example: the system processing 3600 parts per minute that involves image acquisition from 4 sides simultaneously results in a machine vision system acquiring, processing and analyzing 14,400 images per minute. Proper system design dictates that a certain amount of over-capacity be built into the system to handle peak loads. Trigger-to-Image Reliability framework delivers peak load capacity through the concept of circular buffers. It also combines this with user notifications for continuous tracking. While handling peak-load, it is important to monitor the image queue to ensure that various parts of the system stay in-sync and if any variation occurs it is identified and promptly communicated to the user application.



Circular Buffers

The scalable nature of T2IR framework has allowed Teledyne DALSA to add sophisticated parameter switching capability in its hardware products that are well suited for use with circular buffers. Teledyne DALSA Genie cameras, for example, allow users to change trigger delay, strobe outputs, exposure delay and duration, gain, LUTs and FFC (flat field coefficients) on a frame by frame basis. Similarly, the Xtium-CL MX4 frame grabber allows users to switch flat-field and LUTs on a frame by frame basis. When activated, these advanced switching features operate entirely in the acquisition device without using the host CPU resources. Furthermore, the images generated while switching parameters can be saved as a sequence of images.

T2IR provides a broad range of options to handle situations involving too much data. It provides users with necessary information to discard images safely while preserving the accuracy of results from images that were processed. When every image counts, discarding images inevitably leads to reduced throughput. Thus, even when discarding images care must be given to minimize the impact on throughput. The T2IR framework allows applications to discard images early in the acquisition pipeline if it is determined that the system won't be able to handle the images subsequently. The T2IR framework uses a concept of "trash" buffers to discard incoming images efficiently. When a system is not able to handle the incoming data, the acquired images are transferred into the "trash buffer".



Trash Buffer

When this occurs, the user application is notified through a data-overflow event. For example, when transferring images to the host buffers, the Xtium family of frame grabbers monitors downstream bottlenecks and immediately discards images at the acquisition source instead of transferring all the way in the host memory and then discarding it. This allows the acquisition section to return to a ready state immediately to capture next image.

Ensuring Data Quality

Generally machine vision cameras are responsible for handling over-trigger situations—preferably in a predictable fashion. The over-trigger situation for a camera occurs when the rate of triggers for a camera exceeds its maximum frame rate or line rate capability. In cases where the camera does not respond properly and stops sending images at all, T2IR function provide means to recover from this situation and generates notifications to the user application. This is a standard functionality on all Teledyne DALSA frame grabbers. Trapping and handling lost line or a frame is important factor to determine the reliability of the acquisition system and has a direct impact on the accuracy of results. For example, for a line scan camera, a missing line alters the aspect ratio of the object in the image, causing the processing algorithm to produce incorrect results. For area-scan cameras, similarly, it could imply missing objects.

Advanced Diagnostics

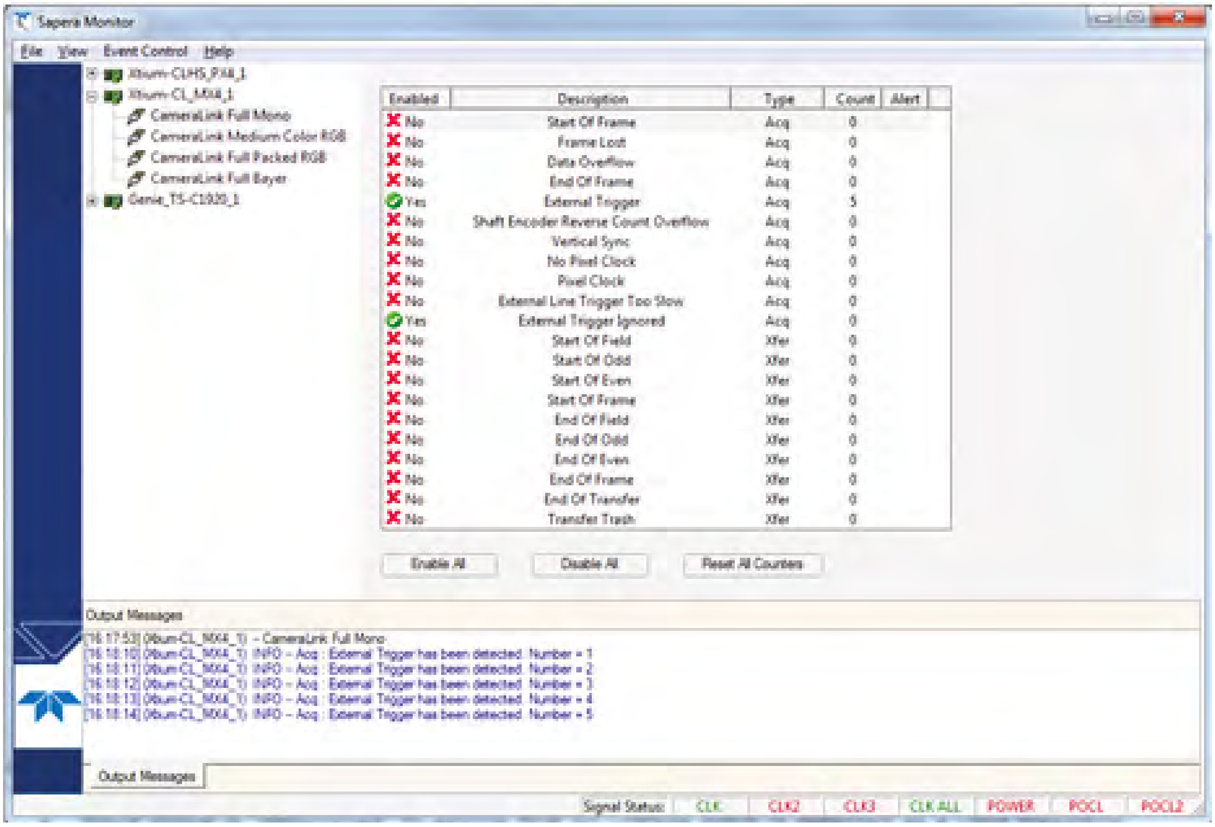
Continuous system monitoring and deep debugging tools help reduce downtime

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

- Sapera Monitor
- External LEDs
- Sapera LogViewer
- Sapera PCI Diagnostic Tool
- Sapera Networking Tool
- Sapera Configuration
- Xtium Diagnostic Tool

Sapera Monitor

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 is based on the Sapera LT T2IR functions. It allows users to see how their application is reacting to various events pertaining to the acquisition system and helps identify and debug problems without having to modify their application.



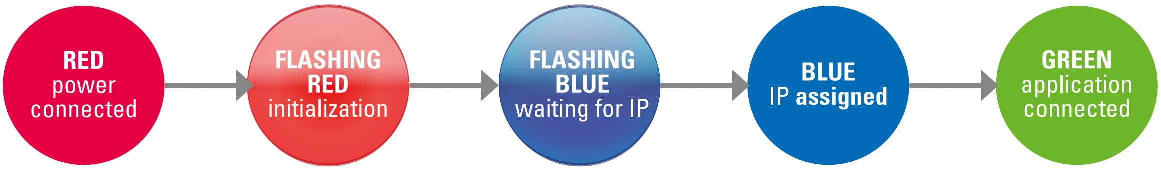
Enabled	Description	Type	Count	Alert
✗ No	Start Of Frame	Acq	0	
✗ No	Frame Lost	Acq	0	
✗ No	Data Overflow	Acq	0	
✗ No	End Of Frame	Acq	0	
✓ Yes	External Trigger	Acq	5	
✗ No	Shaft Encoder Reverse Count Overflow	Acq	0	
✗ No	Vertical Sync	Acq	0	
✗ No	No Pixel Clock	Acq	0	
✗ No	Pixel Clock	Acq	0	
✗ No	External Line Trigger Too Slow	Acq	0	
✓ Yes	External Trigger Ignored	Acq	0	
✗ No	Start Of Field	Xfer	0	
✗ No	Start Of Odd	Xfer	0	
✗ No	Start Of Even	Xfer	0	
✗ No	Start Of Frame	Xfer	0	
✗ No	End Of Field	Xfer	0	
✗ No	End Of Odd	Xfer	0	
✗ No	End Of Even	Xfer	0	
✗ No	End Of Frame	Xfer	0	
✗ No	End Of Transfer	Xfer	0	
✗ No	Transfer Trash	Xfer	0	

```
[16:17:53] (Xtium-CL_MX4_1) - CameraLink Full Mono
[16:18:10] (Xtium-CL_MX4_1) INFO - Acq : External Trigger has been detected. Number = 1
[16:18:11] (Xtium-CL_MX4_1) INFO - Acq : External Trigger has been detected. Number = 2
[16:18:12] (Xtium-CL_MX4_1) INFO - Acq : External Trigger has been detected. Number = 3
[16:18:13] (Xtium-CL_MX4_1) INFO - Acq : External Trigger has been detected. Number = 4
[16:18:14] (Xtium-CL_MX4_1) INFO - Acq : External Trigger has been detected. Number = 5
```

Sapera Monitor

External LEDs

Visual indicators are indispensable features that permit continuous system monitoring right from the power up to full operations. External LEDs are available on the frame grabber bracket and camera back panel. Xtium series frame grabbers, for example, indicate crucial information during boot-up to indicate the board's detection status, PCIe version, lane configuration, and, during operation, presence of camera and acquisition status. For cameras, the status LED indicates boot-up and connection information. For example, the following LED sequence occurs when the Genie is powered up connected to a network with installed Genie Framework software.

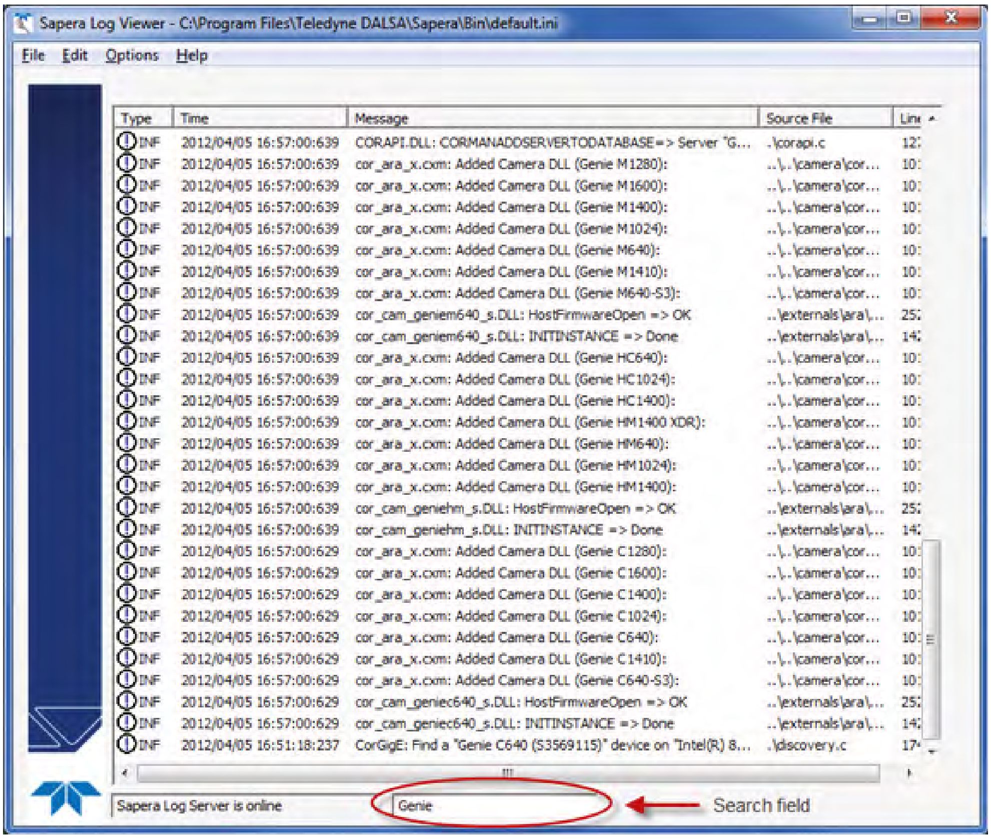

RED power connected
FLASHING RED initialization
FLASHING BLUE waiting for IP
BLUE IP assigned
GREEN application connected

Device status indicated by external LED

Sapera LogViewer

The Teledyne DALSA Sapera Log Viewer utility bundled with Sapera LT installations provides an easy way to view the Sapera messages sent to the Teledyne DALSA acquisition devices and operating system. 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 a valuable tool for pinpointing hard to isolate, infrequent errors.

Sapera LogViewer 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. LogViewer configuration options allow users to set the type of results that are logged. For example, users can choose only to log "Error" messages and ignore "Warnings" or "Info" messages to conserve space. The resulting logs can be dynamically filtered and/or searched for key terms to pinpoint the messages resulting in errors, for example.


Sapera Log Viewer - C:\Program Files\Teledyne DALSA\Sapera\Bin\default.ini
File
Edit
Options
Help

Type	Time	Message	Source File	Line
INF	2012/04/05 16:57:00:639	CORAPI.DLL: CORMANADDSERVERTODATABASE=> Server "G...	..\corapi.c	12:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M1280):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M1600):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M1400):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M1024):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M640):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M1410):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie M640-S3):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_cam_geniem640_s.DLL: HostFirmwareOpen => OK	..\externals\ara\...	25:
INF	2012/04/05 16:57:00:639	cor_cam_geniem640_s.DLL: INITINSTANCE => Done	..\externals\ara\...	14:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HC640):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HC1024):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HC1400):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HM1400 XDR):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HM640):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HM1024):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_ara_x.cxm: Added Camera DLL (Genie HM1400):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:639	cor_cam_geniehm_s.DLL: HostFirmwareOpen => OK	..\externals\ara\...	25:
INF	2012/04/05 16:57:00:639	cor_cam_geniehm_s.DLL: INITINSTANCE => Done	..\externals\ara\...	14:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C1280):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C1600):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C1400):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C1024):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C640):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C1410):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_ara_x.cxm: Added Camera DLL (Genie C640-S3):	..\..\camera\cor...	10:
INF	2012/04/05 16:57:00:629	cor_cam_geniec640_s.DLL: HostFirmwareOpen => OK	..\externals\ara\...	25:
INF	2012/04/05 16:57:00:629	cor_cam_geniec640_s.DLL: INITINSTANCE => Done	..\externals\ara\...	14:
INF	2012/04/05 16:51:18:237	CorGigE: Find a "Genie C640 (S3569115)" device on "Intel(R) 8...	..\discovery.c	17

Sapera Log Server is online

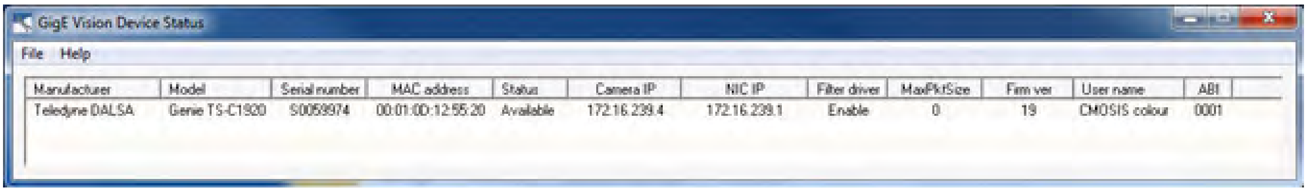
Genie

Vision System events in the Sapera 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 DALSA acquisition devices, only view the messages of interest in each instance.

Sapera GigE Vision Device Status

The GigE Vision Device Status application provides a quick method to view all the Teledyne DALSA GigE devices on your system.

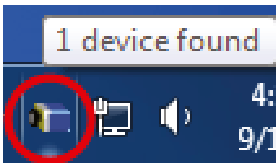


GigE Vision Device Status

Manufacturer	Model	Serial number	MAC address	Status	Camera IP	NIC IP	Filter driver	MaxPktSize	Firm ver	User name	ABI
Teledyne DALSA	Genie TS-C1920	S0059974	00:01:0D:12:55:20	Available	172.16.239.4	172.16.239.1	Enable	0	19	CMOSIS colour	0001

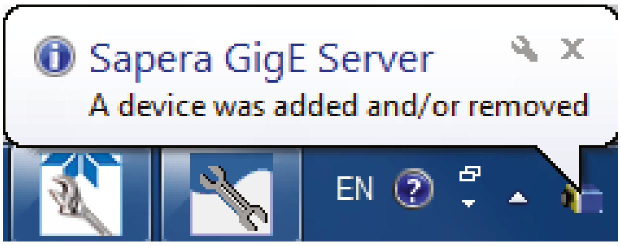
GigE Vision Device Status

It is available directly from the taskbar.



1 device found

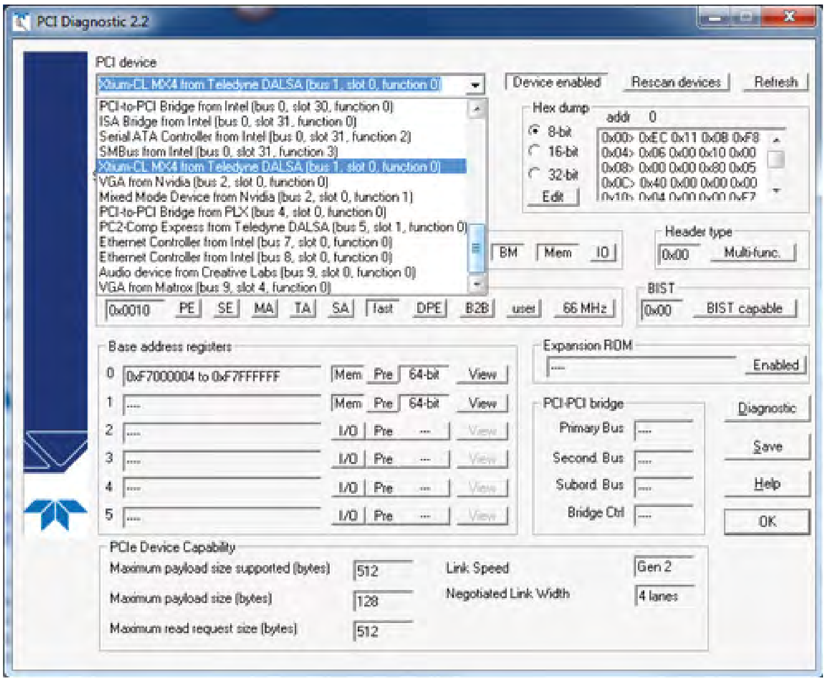
It continuously monitors the system and when a device is added or removed, a message box is displayed:



Sapera GigE Server
A device was added and/or removed

Sapera PCI Diagnostic Tool

The PCI Diagnostic tool allows you to view the low-level hardware resources allocated to the PCI devices on the system. For frame grabbers, you can quickly verify the device capabilities such as the bus and slot utilized, device ID, link speed (for example, Gen 1 or Gen 2) and payload size.



PCI Diagnostic 2.2

PCI device: Xtium-CL MX4 from Teledyne DALSA (bus 1, slot 0, function 0)

```
PCI-to-PCI Bridge from Intel (bus 0, slot 30, function 0)
ISA Bridge from Intel (bus 0, slot 31, function 0)
Serial ATA Controller from Intel (bus 0, slot 31, function 2)
SMBus from Intel (bus 0, slot 31, function 3)
Xtium-CL MX4 from Teledyne DALSA (bus 1, slot 0, function 0)
VGA from Nvidia (bus 2, slot 0, function 0)
Mixed Mode Device from Nvidia (bus 2, slot 0, function 1)
PCI-to-PCI Bridge from PLX (bus 4, slot 0, function 0)
PC2-Comp Express from Teledyne DALSA (bus 5, slot 1, function 0)
Ethernet Controller from Intel (bus 7, slot 0, function 0)
Ethernet Controller from Intel (bus 8, slot 0, function 0)
Audio device from Creative Labs (bus 9, slot 0, function 0)
VGA from Matrox (bus 9, slot 4, function 0)
```

Hex dump: ● 8-bit ○ 16-bit ○ 32-bit; addr 0

```
0x00> 0xEC 0x11 0x08 0xF8
0x04> 0x06 0x00 0x10 0x00
0x08> 0x00 0x00 0x80 0x05
0x0C> 0x40 0x00 0x00 0x00
0x10> 0x04 0x00 0x00 0xF7
```

0x0010

Header type: 0x00

BIST: 0x00

Expansion ROM: ----

Base address registers

```
0  0xF7000004 to 0xF7FFFFFF   Mem  Pre  64-bit
1  ----                       Mem  Pre  64-bit
2  ----                       I/O  Pre  ---
3  ----                       I/O  Pre  ---
4  ----                       I/O  Pre  ---
5  ----                       I/O  Pre  ---
```

PCI-PCI bridge

Primary Bus: ----
Second. Bus: ----
Subord. Bus: ----
Bridge Ctrl: ----

PCIe Device Capability

Maximum payload size supported (bytes): 512
Maximum payload size (bytes): 128
Maximum read request size (bytes): 512
Link Speed: Gen 2
Negotiated Link Width: 4 lanes

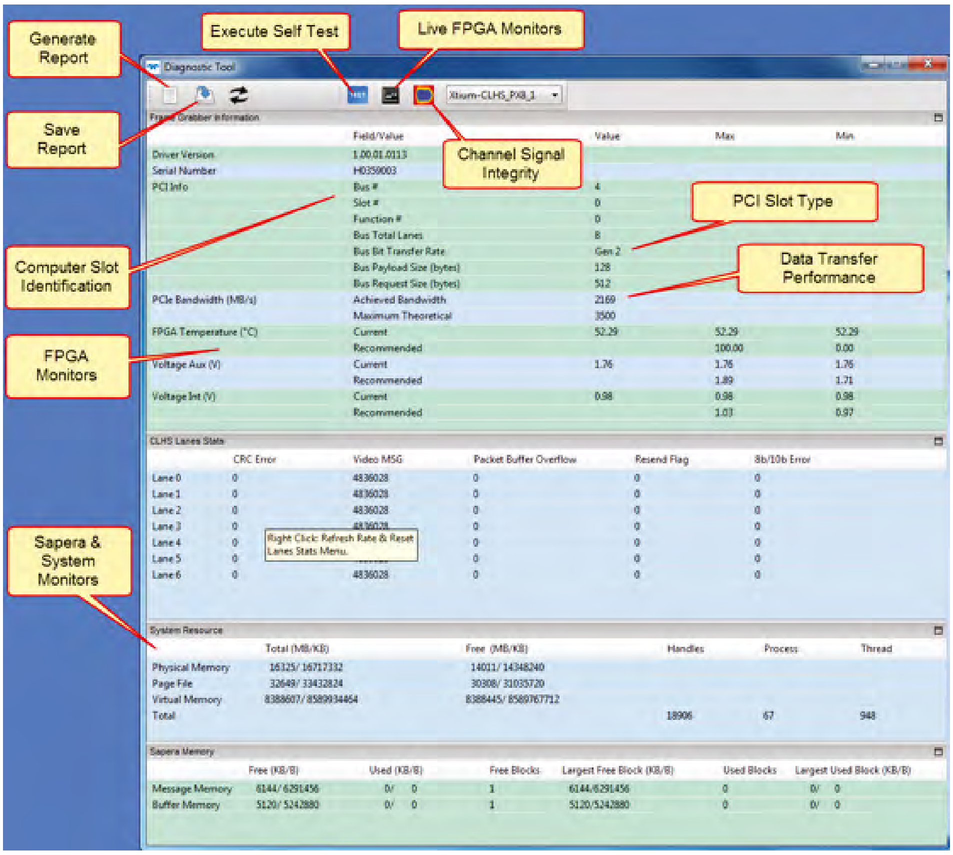
Viewing device resources with Sapera PCI Diagnostic Tool

Xtium Frame Grabber Diagnostic Tool

The Xtium Board Diagnostic Tool provides a quick method to see board status and health of Xtium family of frame grabbers. Additionally, it 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 Xtium board. Toolbar buttons execute the board self-test function and open a FPGA live status window.



Frame Grabber information

	Field/Value	Value	Max	Min
Driver Version	1.00.01.0113			
Serial Number	H0359003			
PCI Info	Bus #	4		
	Slot #	0		
	Function #	0		
	Bus Total Lanes	8		
	Bus Bit Transfer Rate	Gen 2		
	Bus Payload Size (bytes)	128		
	Bus Request Size (bytes)	512		
PCIe Bandwidth (MB/s)	Achieved Bandwidth	2169		
	Maximum Theoretical	3500		
FPGA Temperature (°C)	Current	52.29	52.29	52.29
	Recommended		100.00	0.00
Voltage Aux (V)	Current	1.76	1.76	1.76
	Recommended		1.89	1.71
Voltage Int (V)	Current	0.98	0.98	0.98
	Recommended		1.03	0.97

CLHS Lanes Stats

	CRC Error	Video MSG	Packet Buffer Overflow	Resend Flag	8b/10b Error
Lane 0	0	4836028	0	0	0
Lane 1	0	4836028	0	0	0
Lane 2	0	4836028	0	0	0
Lane 3	0	4836028	0	0	0
Lane 4	0	[illegible]	0	0	0
Lane 5	0	[illegible]	0	0	0
Lane 6	0	4836028	0	0	0

Right Click: Refresh Rate & Reset Lanes Stats Menu.

System Resource

	Total (MB/KB)	Free (MB/KB)	Handles	Process	Thread
Physical Memory	16325/ 16717332	14011/ 14348240			
Page File	32649/ 33432824	30308/ 31035720			
Virtual Memory	8388607/ 8589934464	8388445/ 8589767712			
Total			18906	67	948

Sapera Memory

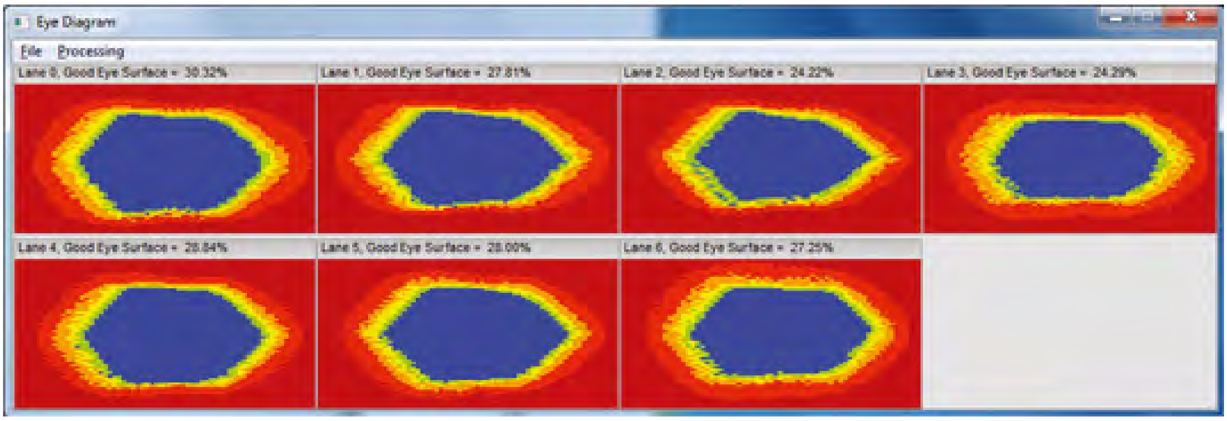
	Free (KB/B)	Used (KB/B)	Free Blocks	Largest Free Block (KB/B)	Used Blocks	Largest Used Block (KB/B)
Message Memory	6144/ 6291456	0/ 0	1	6144/6291456	0	0/ 0
Buffer Memory	5120/ 5242880	0/ 0	1	5120/5242880	0	0/ 0

Diagnostic Tool main window

Important parameters include the PCI Express bus transfer supported by the host computer and the internal Xtium FPGA temperature. The bus transfer defines the maximum data rate possible in the computer, while an excessive FPGA temperature may explain erratic acquisitions due to poor computer ventilation.

Camera Input Eye Diagram Monitor

An Eye diagram is a graphical representation of signal between camera and frame grabber data lanes. This tool can be used to determine if the cable performance starts to degrade over a long period of use. The screen capture below shows a camera with 7 data lanes, where each digital signal is repetitively sampled and overlaid over itself, showing relative low-high transitions of the differential signal. Interpreting the results is easy, the bigger the blue area (eye surface) the better the signal integrity. When all the blue areas are similar in size and shape, it indicates that each wire pair has similar performance.


Eye Diagram
File Processing
Lane 0, Good Eye Surface = 30.32%
Lane 1, Good Eye Surface = 27.81%
Lane 2, Good Eye Surface = 24.22%
Lane 3, Good Eye Surface = 24.29%
Lane 4, Good Eye Surface = 28.84%
Lane 5, Good Eye Surface = 28.00%
Lane 6, Good Eye Surface = 27.25%

Camera Input Eye Diagram

The closure (collapse or horizontal shortening) of the eye surface would indicate problems such as poor signal to noise, high cable capacitance, multipath interference, among many possible digital transmission faults.

Conclusion

Trigger-to-Image Reliability benefits OEMs and end users in the development of high speed, high-volume applications in which machine vision systems need to be able to detect defective objects and reject them from the production line quickly and reliably. OEMs have the peace of mind of knowing that the frame grabbers/GigE cameras used in their application will contribute to making their systems more robust and reliable. In addition, development is greatly simplified, since OEMs don't have to invest time and effort into trying to understand why a system failed or will fail, as the frame grabber/GigE camera itself will detect problems, provide timely notifications and take corrective actions when possible.

Trigger-to-Image Reliability also reduces maintenance and support once the equipment is in the field, leading to lower total system costs. Trigger-to-Image Reliability saves end users extensive time and money by increasing uptime and productivity. As many industries, such as semiconductor, move towards 100% inspection, Trigger-to-Image Reliability helps these manufacturers achieve better product yields at the lowest possible production costs.

Supported Devices

Linea Series – Cost effective line scan camera for GigE and CameraLink interfaces
Genie Nano Series – low cost/ high performance effective area GigE Vision cameras
Xtium Series – PCIe Gen2 frame grabbers for CameraLink and CLHS cameras
Xcelera Series – PCIe Gen1 frame grabbers for Camera Link, HSLink, LVDS cameras