

# Phoenix 8.9 MP Technical Reference Manual



## Welcome! #

---

The PHX089S is a micro compact camera with a versatile board design, making it the world's smallest GigE PoE camera. The PHX089S camera features multiple lens mount and Ethernet connector port options to provide an even smaller footprint. The PHX089S camera is equipped with low noise, high dynamic range image sensing technology suitable for a wide range of applications such as factory automation, medical, scientific and logistics.

PHX089S is targeting machine vision system developers who are looking for:

- Great image quality
- Flexible form factor (change form factor on your own or combine with 90° and 180° Transform Kit)
- Ultra lightweight and compact size
- Large GigE Vision and GenICam feature set



## Safety and Precautions #



Follow these guidelines carefully before using your PHX089S camera.

## Definitions and Symbols #

Below are some warning, safety, and/or tips icons used in this document.



The **Warning** icon indicates a potentially hazardous situation. If not avoided, the situation can result in damage to the product.



The **ESD** icon indicates a situation involving electrostatic discharge. If not avoided, the situation can result in damage to the product.



The **Help** icon indicates important instructions and steps to follow.



The **Light Bulb** icon indicates useful hints for understanding the operation of the camera.



The **Computer** icon represents useful resources found outside of this documentation.

## General Safety Notices #



## Powering the Camera



- The camera may not work, may be damaged, or may exhibit unintended behavior if powered outside of the specified power range.
- When using Power over Ethernet, the power supply must comply with the IEEE 802.3af standard.
- When using the GPIO, the supplied power must be within the stated voltage range.
- See the Power section for further information.

## Operating Temperature



- The camera may not work, may be damaged, or may exhibit unintended behavior if operated outside of the specified temperature range.
- See the Temperature section for further information.



## Electrostatic Discharge

- Ensure proper precautions are implemented to prevent damage from an electrostatic discharge.



## Image Quality

- Dust or fingerprints on the sensor may result in a loss of image quality.
- Work in a clean and dust-free environment.
- Attach the dust cap to the camera when a lens is not mounted.
- Use only compressed ionized air or an optics cleaner to clean the surface of the sensor window.



# Contact Us #



**Think LUCID. Go LUCID.**

Contact Sales

<https://thinklucid.com/contact-us/> (<https://thinklucid.com/contact-us/>)

Contact Support

<https://thinklucid.com/support/> (<https://thinklucid.com/support/>)



## Installing the Camera Hardware #

---

### Mounting #

The Phoenix is equipped with four M3 mounting holes and five M2 mounting holes.

One pair of M2 and one pair of M3 holes are present on opposite sides of the lens mount surface. An additional M2 hole is located on the bottom side of the third spacer.

The camera in the unfolded configuration has 4 mounting points on each of the main PCBs. Each core PCB is equipped with four M1.7 holes in the corner which can be used as mounting points.

### Connector #

The PHX089S camera comes in 3 different interface connector options as described below.

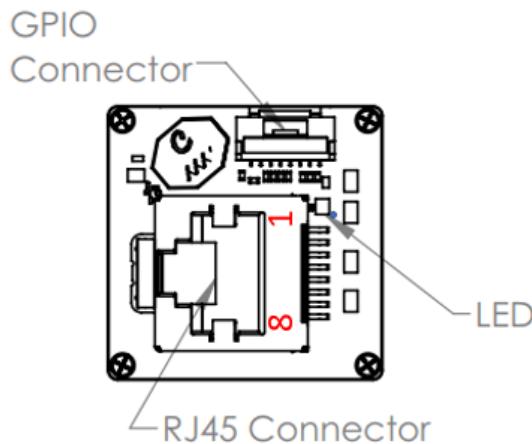


## RJ45 Connector #

The 8-pin RJ-45 jack connector is available as an option for PHX089S. The connector supports 100/1000 Mbit/s Ethernet connection to the camera. If power is not supplied via the GPIO connector, Power over Ethernet (PoE) is needed. Recommended mating connector is 8-pin RJ-45 plug (snap-in).

### Pin 1 Position

As shown below, pin 1 of the RJ-45 connector is indicated by a dot on the PCB.



### Pin Numbering and Assignments

Pin Number (Camera Side)	Pin Name	Pin Description
1	BI_DA+	Bi-directional pair A+
2	BI_DA-	Bi-directional pair A-
3	BI_DB+	Bi-directional pair B+
4	BI_DC+	Bi-directional pair C+
5	BI_DC-	Bi-directional pair C-
6	BI_DB-	Bi-directional pair B-
7	BI_DD+	Bi-directional pair D+



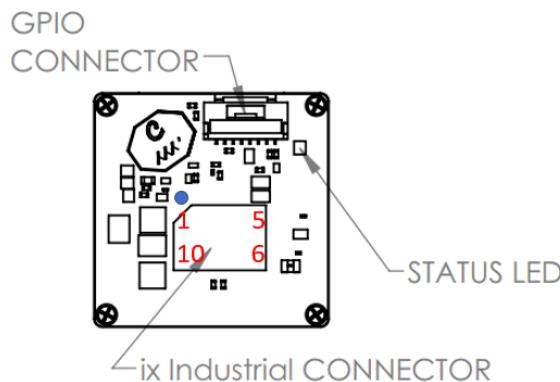
<b>Pin Number (Camera Side)</b>	<b>Pin Name</b>	<b>Pin Description</b>
8	BI_DD-	Bi-directional pair D-

## IX Connector #

The 10-pin ix connector is available as an option for PHX089S. The ix connector is supplied by Hirose Electric Co. (<https://www.hirose.com/product/series/IX?lang=en#>). The ix Industrial connector uses 70% less space compared to RJ45 sockets. It provides high EMC immunity, active metal snap-in hooks, a locking force greater than 98 n, guarantee of 5,000 mating cycles, and railway EN 50155 specification compliance for shock and vibration resistance.

### Pin 1 Position

As shown below, pin 1 of the ix connector is indicated by a dot on the PCB.



### Pin Numbering and Assignments

<b>Pin Number (Camera Side)</b>	<b>Pin Name</b>	<b>Pin Description</b>
1	BI_DA+	Bi-directional pair A+



<b>Pin Number (Camera Side)</b>	<b>Pin Name</b>	<b>Pin Description</b>
2	BI_DA-	Bi-directional pair A-
4	BI_DC+	Bi-directional pair C+
5	BI_DC-	Bi-directional pair C-
6	BI_DB+	Bi-directional pair B+
7	BI_DB-	Bi-directional pair B-
9	BI_DD+	Bi-directional pair D+
10	BI_DD-	Bi-directional pair D-

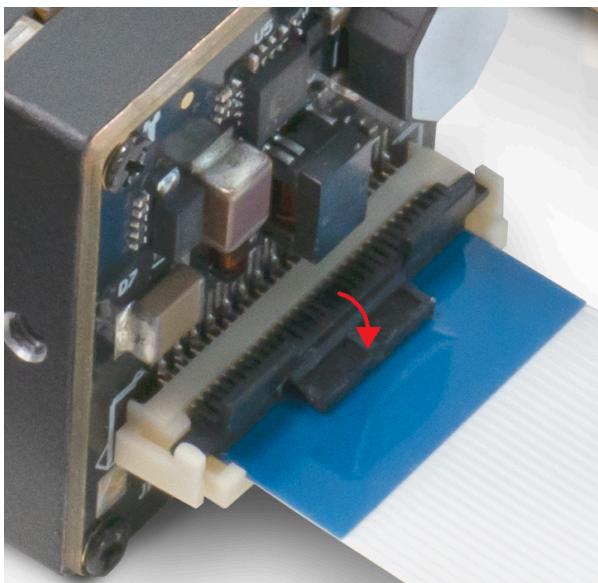
## FFC Connector #

The 30-pin flexible flat cable (FFC) connector is also available as an option for PHX089S. The FFC connector is supplied by Kyocera Electronic Components (<https://global.kyocera.com/prdct/electro/>). The connector provides connections for image transfer, input and output signals, and power supply.

### Attaching the connector

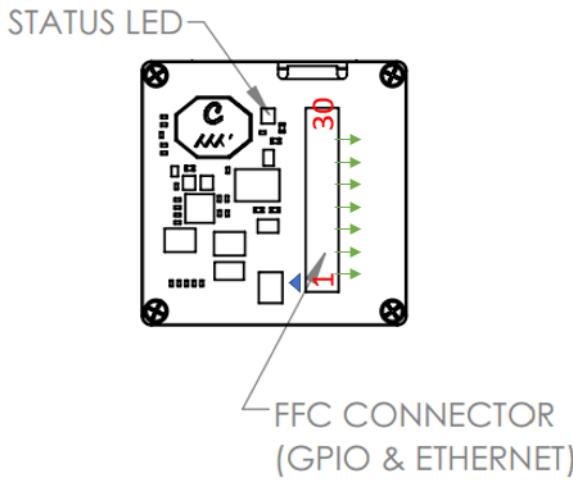
When using an FFC cable from LUCID, make sure the blue side is facing toward the locking tab on the camera-side connector. Engage this locking tab to its horizontal position to lock it.





## Pin 1 Position

As shown below, pin 1 of the FFC connector is indicated by a triangle on the PCB. Contact of the FFC connector is indicated below with green arrows.



## Pin Numbering and Assignments

Pin Number (Camera Side)	Pin Name	Pin Description
1	BI_DA+	Bi-directional pair A+
2	BI_DA-	Bi-directional pair A-
5	BI_DB+	Bi-directional pair B+
6	BI_DB-	Bi-directional pair B-



<b>Pin Number (Camera Side)</b>	<b>Pin Name</b>	<b>Pin Description</b>
8	BI_DC+	Bi-directional pair C+
9	BI_DC-	Bi-directional pair C-
11	BI_DD+	Bi-directional pair D+
12	BI_DD-	Bi-directional pair D-
14	OPTOGND	Opto-Isolated Ground Reference for LINE1 and LINE0
15	LINE1	Opto-Isolated Output
16	LINE0	Opto-Isolated Input
23	VAUX	Power Input, 12-24V DC
24	VAUX	Power Input, 12-24V DC
25	GND	Ground Reference for VAUX, VOUT, LINE2, and LINE3
26	GND	Ground Reference for VAUX, VOUT, LINE2, and LINE3
28	LINE3	Non-Isolated, High-Speed, Bi-Directional IO
29	LINE2	Non-Isolated, High-Speed, Bi-Directional IO
30	VOUT	Power Output, 2.5V DC

LUCID provides FFC connector board accessory parts for customers who do not wish to make their own FFC termination. These accessory parts can be found our accessories page (<https://thinklucid.com/category/phoenix-transform-kits/>).

The maximum recommended length of the flex cable is 15cm – based on LUCID's internal testing this does not impact the GbE performance significantly. Longer cable lengths are possible, but impact on EMI/EMC and GbE reach should be considered (has not been tested by LUCID). While there is a definite benefit in using shielded/controlled impedance FFC for longer flex length,



current FFC connector is not compatible with such FFC types. LUCID's internal testing shows that with 15cm FFC, camera EMI/EMC performance is similar to RJ45 version (within FCC Part 15 class A / EN55022 class A).

The connector order code is Kyocera Manufacturer Part Number:  
046251030010846+.

---

## GigE Cable #

- For the best performance, a shielded Ethernet Cat5e or higher grade should be used. STP shielding is recommended for minimal electromagnetic interference in environments with harsh EMI conditions.
  - An unshielded or lower grade/quality Ethernet cable may result in loss of camera connection and/or lost and inconsistent image data.
  - The maximum cable length from camera to host with no switch or repeater in between is 100 meters.
  - LUCID Vision Labs recommends using qualified Ethernet cables from our web store.
- 

## GPIO Cable #

The PHX089S camera is equipped with an 8-pin General Purpose Input/Output (GPIO) connector at the back.

- Poor quality unshielded GPIO cable may result in false triggers or missing triggers.
- The GPIO connector part number is BM08B-NSHSS-TBT. The mating connector is NSHR-08V-S.
- Recommended wire thickness is AWG26 or AWG28.
- LUCID Vision Labs recommends using qualified GPIO cables from our web store.



## Lens #

---

### C Mount Lens #

C mount lenses can be used on a C mount camera. According to standard, the C mount flange back distance is 17.53 mm.

---

### Note about using a heavy lens with the camera #

Mounting a heavy and long lens may cause damage to the camera board. If the lens is considerably heavier than the camera, the lens' weight may exert significant force on the lens mount attached to the camera's board causing unexpected damage to the board and soldered components. If a heavy lens is necessary for the production environment, it is recommended to use the lens as a mounting point rather than the camera to avoid damage to the camera.



## Configuring the Camera and Host System #

---

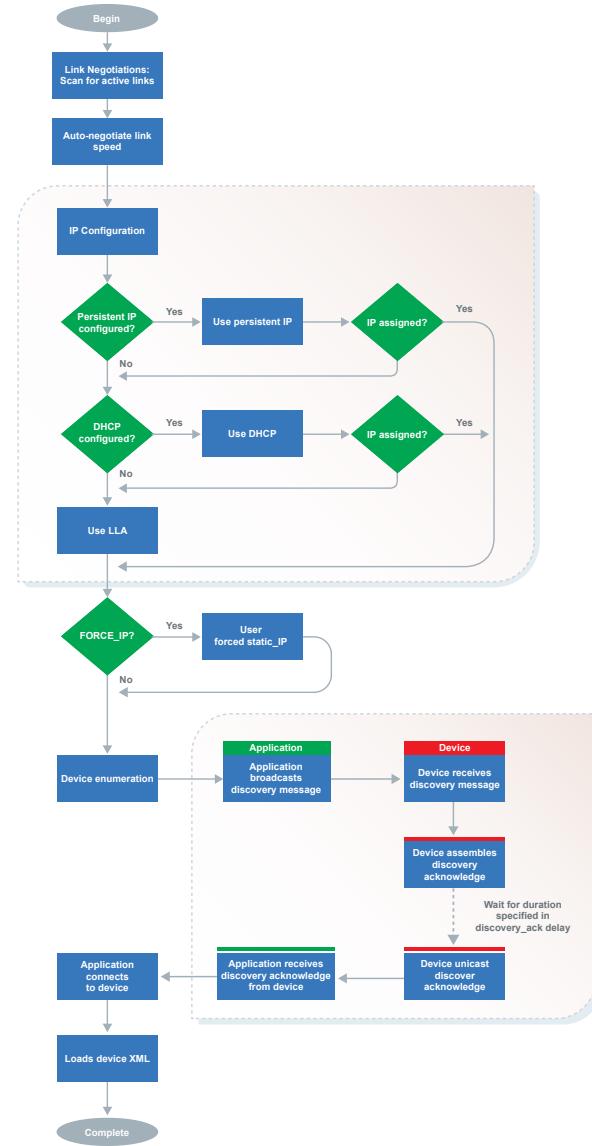
### Installing the Ethernet Driver #



LUCID Vision Labs recommends updating to the latest version of your Ethernet adapter's driver before connecting your camera. You may need to navigate to the manufacturer's website to find the latest version of the driver.

# Device Discovery and Enumeration #

Lucid cameras are discovered and enumerated with the following process:



## Camera IP Setup #



There are three methods used by LUCID cameras to obtain an IP address:

- Persistent – The camera uses a fixed IP address
- Dynamic Host Configuration Protocol (DHCP) – The camera is assigned an address from a DHCP server
- Link-Local Address (LLA) – The camera obtains an address in the Link-Local Address range from 169.254.1.0 to 169.254.254.255



A Link-Local Address is selected with a pseudo-random number generator, so an LLA address cannot be reliably predicted. If you require a consistent IP address, you should set up a persistent IP on the camera. For more information on Link-Local Address selection, refer to Section 2.1 in RFC 3927 (<https://tools.ietf.org/html/rfc3927#section-2.1>).



The camera must be on the same subnet as the Ethernet adapter and have a valid IP address before use.

Persistent IP and DHCP configurations can be disabled on the camera. The default camera configuration is as follows:

<b>Persistent IP</b>	Disabled
<b>DHCP</b>	Enabled
<b>LLA</b>	Enabled (always enabled)

Out of the box, the camera first attempts to connect using DHCP. If the camera is unable to connect using DHCP, it will use Link-Local Addressing.

## Setting Up Persistent IP #



The following pseudocode demonstrates setting up persistent IP:

```
1 // Connect to camera  
2 // Get device node map  
3  
4 GevCurrentIPConfigurationPersistentIP = true;  
5 GevPersistentIPAddress = 192.168.0.10; //Enter persistent IP address for  
6 GevPersistentSubnetMask = 255.255.255.0;  
7 GevPersistentDefaultGateway = 192.168.0.1;
```



## Bandwidth Management #

### Jumbo Frames #

LUCID Vision Labs recommends enabling jumbo frames on your Ethernet adapter. A jumbo frame is an Ethernet frame that is larger than 1500 bytes. Most Ethernet adapters support jumbo frames, however it is usually turned off by default.

Enabling jumbo frames on the Ethernet adapter allows a packet size of up to 9000 bytes to be set on the Triton. The larger packet size will enable optimal performance on high-bandwidth cameras, and it usually reduces CPU load on the host system. Please note in order to set a 9000 byte packet size on the camera, the Ethernet adapter must support a jumbo frame size of 9000 bytes or higher.

The following table are some of the Ethernet adapters that LUCID Vision Labs has tested:

Product Name	Maximum Jumbo Frame Size
ADLINK PCIe-GIE64+	9000



Product Name	Maximum Jumbo Frame Size
Neousys PCIe-PoE354at	9500
Intel EXPI9301CT (non-POE)	9000

If you still experience issues such as lost packets or dropped frames, you can also try:

- Updating the Ethernet adapter driver (you may need to enable jumbo frames again after updating).
- Increasing the receive buffer size in your Ethernet adapter properties.
- Reducing the DeviceLinkThroughputLimit value (this may reduce maximum frame rate).

---

## Receive Buffers #

A receive buffer is the size of system memory that can be used by the Ethernet adapter to receive packets. Some Ethernet adapter drivers or the operating system itself may set the receive buffer value to a low value by default, which may result in decreased performance. Increasing the receive buffer size, however, will also result in increased system memory usage.

---

## Device Link Throughput Limit #

The Device Link Throughput Limit is the maximum available bandwidth for transmission of data represented in bytes per second. This can be used to control the amount of bandwidth used by the camera. The maximum available frame rate may decrease when this value is lowered since less bandwidth is available for transmission.



# User Sets, Streamables, and File Access #

## User Sets #

The PHX089S features two customizable user sets to load or save user-defined settings on the camera. Accessing the user set named Default will allow loading or saving of factory default settings. The Triton will load the user set selected in UserSetDefault when powering up or when reset.



If the camera is acquiring images, AcquisitionStop must be called before loading or saving a user set.

## Streamables #

A camera feature marked as Streamable allows the feature's current value to be stored to and loaded from a file.

## File Access #

The Triton features persistent storage for generic file access on the camera. This feature allows users to save and load a custom file up to 16 megabytes with UserFile. Users can also save and load User Set contents to and from a file.



 Loading new firmware onto the camera may overwrite existing UserFile contents.

The following pseudocode demonstrates reading from the camera using file access:

```
1 // Connect to camera
2 // Get device node map
3 FileSelector = UserFile1;
4 FileMode = Read;
5 FileOperationSelector = Open;
6 FileOperationExecute();
7 // Device sets FileOperationSelector = Close
8 // Read custom file from camera
9 FileOperationExecute();
10 // Device sets FileOperationSelector = Open
```

The following pseudocode demonstrates writing to the camera using file access:

```
1 // Connect to camera
2 // Get device node map
3 FileSelector = UserFile1;
4 FileMode = Write;
5 FileOperationSelector = Open;
6 FileOperationExecute();
7 // Device sets FileOperationSelector = Close
8 // Write custom file to camera
9 FileOperationExecute();
10 // Device sets FileOperationSelector = Open
```

 User Sets, Streamables, and File Access functions do not load or save camera IP configuration settings.



# Using the Camera #



## Using Arena SDK #

The Arena Software Development Kit (SDK) is designed from the ground up to provide customers with access to the latest in industry standards and computer technology. The SDK supports Lucid GigE Vision cameras on both Windows and Linux platforms.

## Using ArenaView #

The Arena SDK includes Lucid's easy to use GUI called ArenaView. Based on the GenICam standard, ArenaView allows you to access and validate camera features quickly and easily through the GenICam XML based feature tree. Optimized for today's diverse range of user preferences, our viewer improves readability on higher resolutions and includes options for different color schemes.



For more information on Arena SDK and ArenaView, please refer to our website (<https://thinklucid.com/arena-software-development-kit/>).

## Using the Camera with Third Party Software #

Lucid's cameras are compatible with many third party GigE Vision software packages including libraries from Cognex, MVtec, Matrox Imaging Library, Mathworks, and National Instruments. For more information on how to use your



PHX089S with other libraries, please refer to our 3rd Party Getting Software Guides (<https://thinklucid.com/3rd-party-software-guides/>).



Please refer to our 3rd Party Getting Software Guides (<https://thinklucid.com/3rd-party-software-guides/>) for more info.



## Camera Specifications #

---

### Power #

The PHX089S can be powered via the Ethernet cable using Power over Ethernet (PoE) or the GPIO using the pins described in the GPIO Characteristics section.

When using PoE, the power supply must comply with the IEEE 802.3af standard. You can find recommended parts to power the camera in our web store.

---

### Temperature #

The PHX089S should be kept in the following storage, operating, and humidity conditions.

Storage Temperature	-30 to 60°C
Operating Temperature	-10 to 55°C ambient
Humidity	Operating: 20% ~ 80%, relative, non-condensing



Placing the camera outside of these conditions may result in damage to the device.

The PHX089S is equipped with a built in temperature sensor that can be read by reading the DeviceTemperature property.



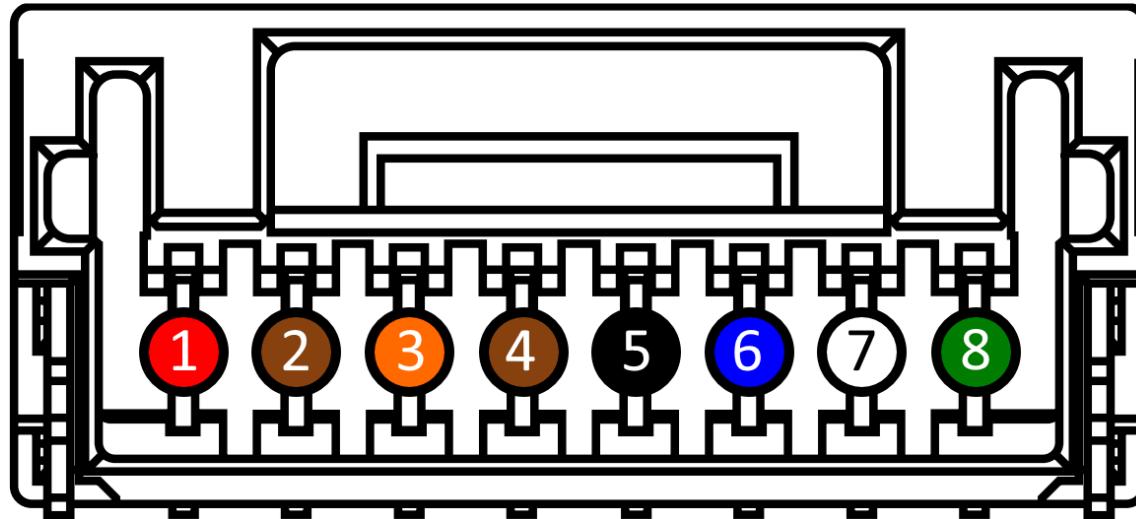
The camera can get hot to touch if it has been streaming images for an extended period of time.



DeviceTemperature can show values outside of the operating temperature range. This is generally acceptable as long as the camera is kept within the stated operating temperature range.

## GPIO Characteristics #

### GPIO Pinout Diagram



GPIO connector as seen from rear Phoenix view. Pin colors correspond to GPIO-8P20 (<https://thinklucid.com/product/gpio-8-pin-20cm-jst-cable/>) cable from LUCID Vision Labs.

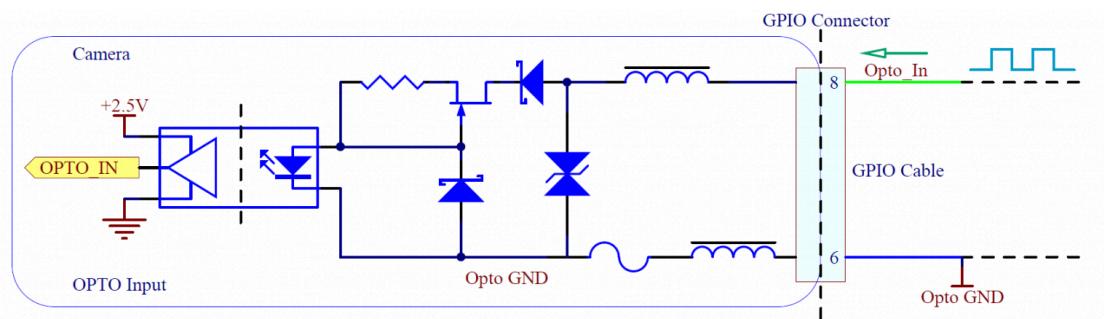


Pin Number	Pin Description
1 (Red)	V <sub>AUX</sub> (12-24V DC Power Input)
2 (Brown)	Non-isolated bi-directional GPIO channel (Line 2)
3 (Orange)	V <sub>DD</sub> GPIO (2.5V Power Output) (Line 4)
4 (Brown)	Non-isolated bi-directional GPIO channel (Line 3)
5 (Black)	GND (Camera GND)
6 (Blue)	OPTO GND (Opto-isolated Reference)
7 (White)	OPTO OUT (Opto-isolated Output) (Line 1)
8 (Green)	OPTO IN (Opto-isolated Input) (Line 0)

Consult the Turning on GPIO Voltage Output section for enabling V<sub>DD</sub>.

## GPIO Schematics #

### Opto-isolated Input – GPIO Line 0 #



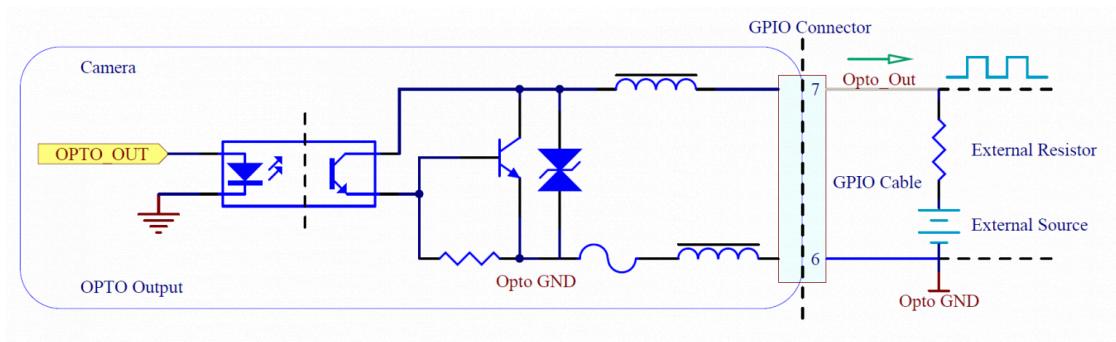
Opto-isolated Input Measurements:



Voltage (v)	Max Rise Delay (us)	Max Fall Delay (us)	Max Rise Time (us)	Max Fall Time (us)	Min Pulse Input (us)	Min Input High (v)	Min Input Low (v)
2.5	1	1	1	1	2	2.5	1.0
5	1	1	1	1	2	2.5	1.0

Sample values measured at room temperature. Results may vary over temperature and setup.

## Opto-isolated Output – GPIO Line 1 #



Opto-isolated Output Measurements:

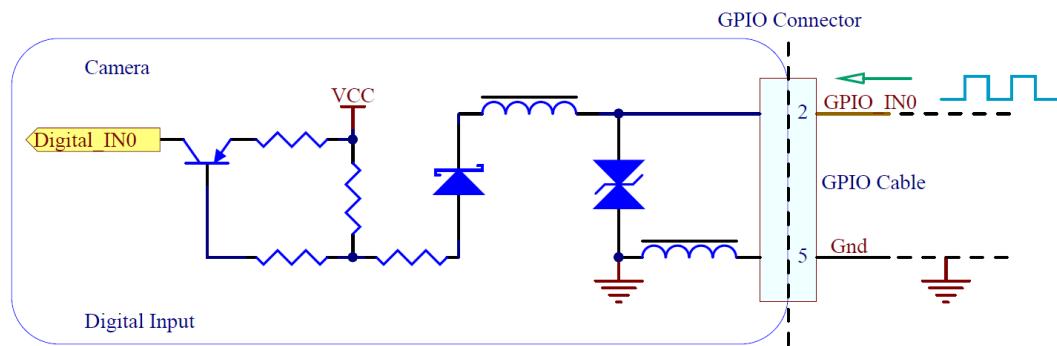
Voltage (v)	External Resistor ( $\Omega$ )	Max Rise Delay (us)	Max Fall Delay (us)	Max Rise Time (us)	Max Fall Time (us)	Current (mA)	Low Level (v)
2.5	150	50	5	40	5	5.7	0.9
2.5	330	50	5	40	5	2.9	0.8
2.5	560	50	5	40	5	1.9	0.5
2.5	1k	50	5	40	5	1.2	0.3
5	330	50	5	50	5	6.6	0.9
5	560	50	5	50	5	4	0.7



<b>Voltage (v)</b>	<b>External Resistor (Ω)</b>	<b>Max Rise Delay (us)</b>	<b>Max Fall Delay (us)</b>	<b>Max Rise Time (us)</b>	<b>Max Fall Time (us)</b>	<b>Current (mA)</b>	<b>Low Level (v)</b>
5	1k	50	5	50	5	2.4	0.5
5	1.8k	50	5	50	5	1.4	0.4
12	1k	50	5	60	5	6	0.9
12	1.8k	50	5	60	5	3.4	0.9
12	2.7k	50	5	60	5	2.4	0.7
12	4.7k	50	5	60	5	1.5	0.5
24	1.8k	60	5	60	5	7.1	0.9
24	2.7k	60	5	60	5	4.7	0.9
24	4.7k	60	5	60	5	2.8	0.7
24	6.8k	60	5	60	5	2.1	0.6

Sample values measured at room temperature. Results may vary over temperature and setup.

## Non-isolated Input - GPIO Line 2 #



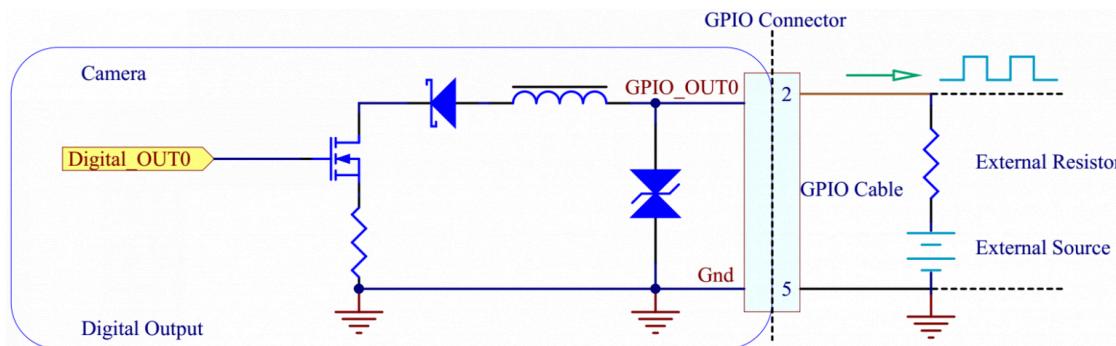
Non-isolated Input Measurements:



<b>Voltage (v)</b>	<b>Max Rise Delay (us)</b>	<b>Max Fall Delay (us)</b>	<b>Max Rise Time (us)</b>	<b>Max Fall Time (us)</b>	<b>Min Pulse Input (us)</b>	<b>Min Input High (v)</b>	<b>Min Input Low (v)</b>
2.5	1	1	1	1	2	2.5	0.5
5	1	1	1	1	2	2.5	0.5

Typical values measured at room temperature. Results may vary over temperature.

## Non-isolated Output - GPIO Line 2 #



Non-isolated Output Measurements:

<b>Voltage (v)</b>	<b>External Resistor (<math>\Omega</math>)</b>	<b>Max Rise Delay (us)</b>	<b>Max Fall Delay (us)</b>	<b>Max Rise Time (us)</b>	<b>Max Fall Time (us)</b>	<b>Current (mA)</b>	<b>Low Level (v)</b>
2.5	150	0.5	0.5	1	0.5	4.3	1.3
2.5	330	0.5	0.5	1	0.5	2.6	1
2.5	560	0.5	0.5	1	0.5	1.8	0.8
2.5	1k	0.5	0.5	1	0.5	1.1	0.6
5	330	0.5	0.5	1	0.5	5.6	1.4
5	560	0.5	0.5	1	0.5	3.7	1.1



<b>Voltage (v)</b>	<b>External Resistor (Ω)</b>	<b>Max Rise Delay (us)</b>	<b>Max Fall Delay (us)</b>	<b>Max Time (us)</b>	<b>Max Fall Time (us)</b>	<b>Current (mA)</b>	<b>Low Level (v)</b>
5	1k	0.5	0.5	1	0.5	2.3	0.9
5	1.8k	0.5	0.5	1	0.5	1.4	0.7
12	1k	0.5	0.5	1	0.5	5.5	1.4
12	1.8k	0.5	0.5	1	0.5	3.2	0.9
12	2.7k	0.5	0.5	1	0.5	2.3	0.9
12	4.7k	0.5	0.5	1	0.5	1.5	0.7
24	1.8k	0.5	0.5	2	0.5	6.5	1.6
24	2.7k	0.5	0.5	2	0.5	4.5	1.3
24	4.7k	0.5	0.5	2	0.5	2.6	0.9
24	6.8k	0.5	0.5	2	0.5	1.8	0.8

Typical values measured at room temperature. Results may vary over temperature.

## Non-isolated Input - GPIO Line 3 #

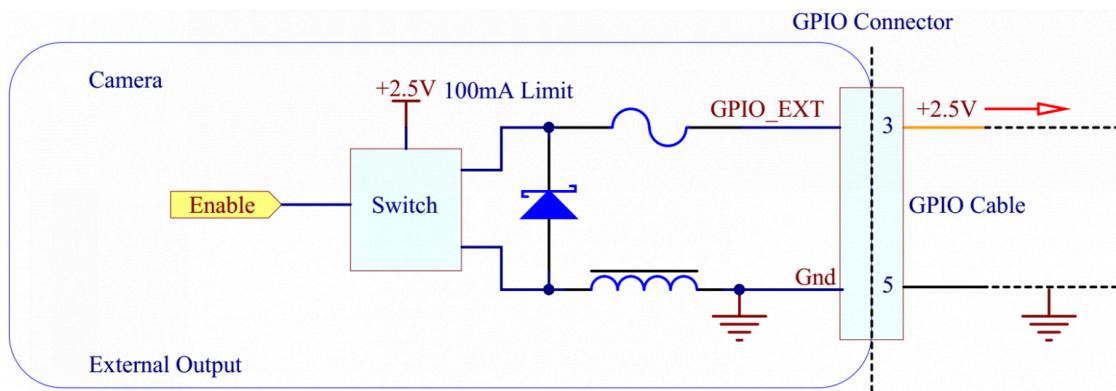
Same as Non-isolated Input – GPIO Line 2 (GPIO\_IN is Pin 4, GND is Pin 5)

## Non-isolated Output - GPIO Line 3 #

Same as Non-isolated Output – GPIO Line 2 (GPIO\_OUT is Pin 4, GND is Pin 5)



## 2.5V Output #



## LED Status #

The Triton camera is equipped with an LED that identifies the current state of the camera.

LED Status	Status Information
Flashing red	Camera powered, but no Ethernet link established.
Flashing green	Camera powered, Ethernet link established, but no network traffic.
Solid green	Camera powered, Ethernet link established, and there is network traffic.
Flashing red/green	Firmware update in progress.
Solid red	Error. Firmware update failed.

The following LED sequence occurs when the camera is powered up and connected to a network:

1. LED off, plug in the Ethernet cable.



2. LED on, flashing red.
3. After link is established, LED becomes flashing green.
4. Launch application and start capturing images, LED becomes solid green.

The status LED can also be controlled using the DeviceIndicatorMode property.

Possible values are:

- Inactive: LED is off.
- Active: LED indicates camera status according to the above table.

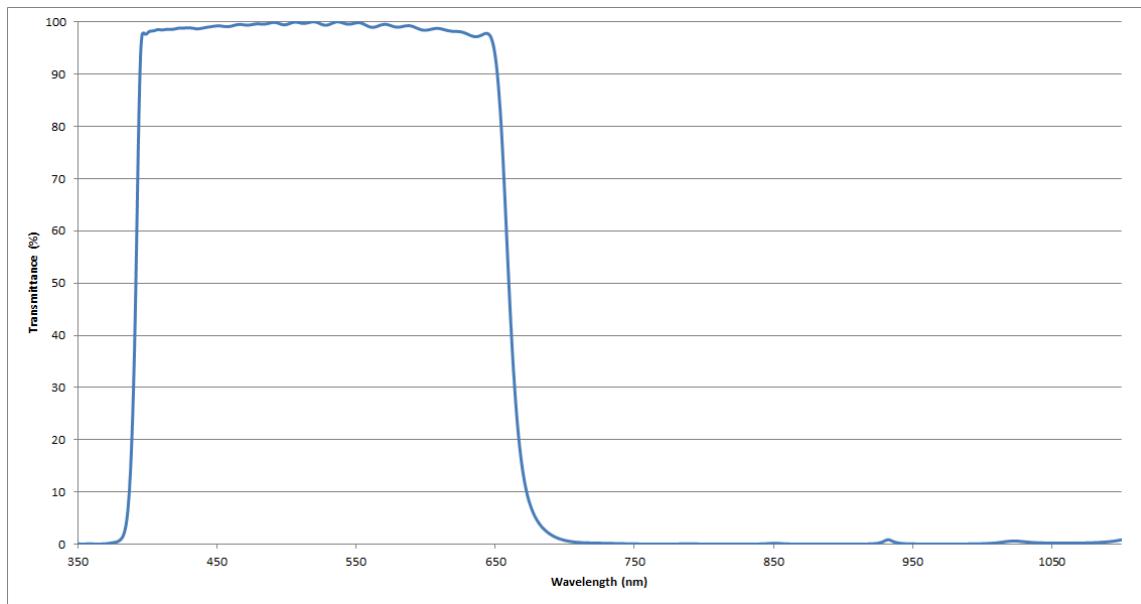
---

## IR Filter #

Color cameras from LUCID Vision Labs are equipped with an IR filter that is installed under the gasket of the mount. Mono cameras are equipped with a transparent glass window instead of an IR filter. The dimensions of the IR filter / transparent glass window are as follows:

Model	Size	Thickness
24 x 24mm Phoenix C-Mount	14 x 14mm	1mm
24 x 24mm Phoenix NF-Mount	11.7 x 10mm	1mm
28 x 28mm Phoenix C-Mount	18 x 14.5mm	1mm





The IR filter and transparent glass window have anti-reflective coating on one side.

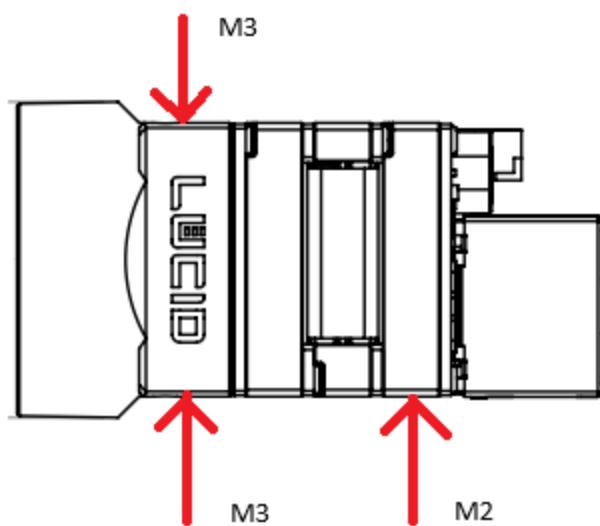
## Shock and Vibration #

To ensure the camera's stability, LUCID's PHX089S camera is tested under the following shock/vibration conditions. After the testing, the camera showed no physical damage and could produce normal images during normal operation.

Test	Standard	Parameters
Shock	DIN EN 60068-2-27	Each axis (x/y/z), 20g, 11ms, +/- 10 shocks
Bump	DIN EN 60068-2-27	Each axis (x/y/z), 20g, 11ms, +/- 100 bumps
Vibration (random)	DIN EN 60068-2-64	Each axis (x/y/z), 4.9g rms, 15-500Hz, 0.05g <sup>2</sup> /Hz acceleration, 30min per axis

Due to the nature of the stacked design, the Phoenix camera was mounted by securing the M3 mounting holes on the top and bottom of the camera lens mount as well as the rear M2 hole as illustrated below.





Due to the nature of the stacked design, the Phoenix camera was mounted by securing the M3 mounting holes on the top and bottom of the camera lens mount as well as the rear M2 hole as illustrated below.

To mimic a real camera usage scenario, a dummy lens mass is attached to the camera lens mount during the shock and vibration test. The dummy lens has the following properties.

Mass	66 grams
Center of mass from C-mount	16 mm

## Specification Tests #

### FCC #

This product has been tested and complies 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 his own expense.

Users are advised that any changes or modifications not approved by LUCID Vision Labs will void the FCC compliance. The product is intended to be used as a component of a larger system, hence users are advised that cable and other peripherals may affect overall system FCC compliance.

---

## **RoHS, REACH, and WEEE #**

LUCID Vision Labs declares the PHX089S camera is in conformity of the following directives:

- RoHS 2011/65/EC
  - REACH 1907/2006/EC
  - WEEE 2012/19/EC
- 

## **CE #**

LUCID Vision Labs declares the PHX089S meets requirements necessary for CE marking. The product complies with the requirements of the listed directives below:

- EN61000-4-2
- EN61000-4-3
- EN61000-4-4

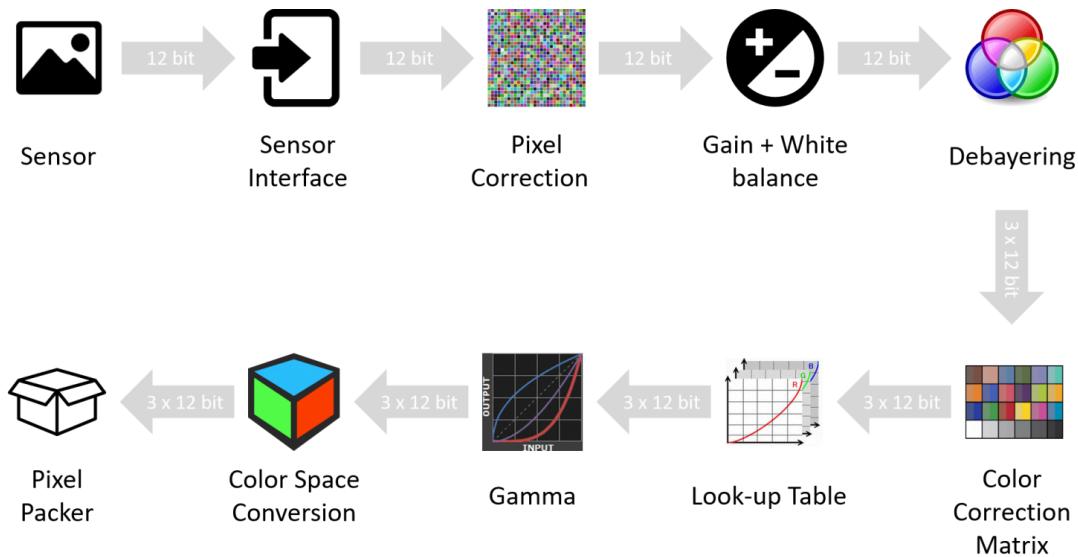




# Camera Features #

## Image Processing Controls #

The PHX089S camera is equipped with the following image processing control flow.



The details of each of the image processing controls are described in the following documentation.

## Defect Pixel Correction #

The PHX089S supports a list of pixel coordinates to be corrected via firmware. For the list of pixel coordinates, their actual pixel values are replaced by interpolation of their neighboring pixel values. The camera has a preloaded pixel correction list and these pixels are loaded during the camera manufacturing process. It is natural that sensors come with defective pixels and



they are inevitable in the semi-conductor manufacturing process. As the camera operates longer in heat or is exposed to radiation, more defective pixels may appear. Users can update the pixel correction list any time.

### Steps to add a new pixel to the correction list

1. Set OffsetX and OffsetY to zero. Set Width and Height to the maximum allowed value.
2. Set Gain to zero and note the coordinates of any bright pixels in the image. Please ensure the camera is not exposed to light by covering it with a lens cap and placed in a dark box.
3. Fire PixelCorrectionAdd command.
4. Enter the X-coordinate noted in step 2 into PixelCorrectionPositionX.
5. Enter the Y-coordinate noted in step 2 into PixelCorrectionPositionY.
6. Fire PixelCorrectionSave command.
7. Repeat steps 3-6 as needed and fire the DefectCorrectionSave when done.



Pixel correction is still applied if the image geometry changes (e.g. applying ReverseX, ReverseY, a region of interest, or binning).

The following pseudocode demonstrates adding a defective pixel to the correction list:



```
1 // Connect to camera
2 // Get device node map
3 // Set maximum width and height
4 OffsetX = 0;
5 OffsetY = 0;
6 Width = Max Width;
7 Height = Max Height;
8 // Set constant ExposureTime and Gain
9
10 DefectCorrectionGetNewDefect();
11 DefectCorrectionPositionX = 25; // The X-coordinate of the blemish pixel
12 DefectCorrectionPositionY = 150; // The Y-coordinate of the blemish pixel
13 DefectCorrectionApply();
14
15 // Repeat the above four steps as needed. When complete:
16 DefectCorrectionSave();
```



## Gain #

Gain refers to a multiplication factor applied to a signal to increase the strength of that signal. On Lucid cameras, gain can be either manually adjusted or automatically controlled.

Some cameras feature gain that is purely digital while others allow for analog gain control up to a certain value, beyond which the gain becomes digital. Depending on the camera family and sensor model, the specific gain control can vary.

### Analog Gain

Analog Gain refers to amplification of the sensor signal prior to A/D conversion.

### Digital Gain

Digital Gain refers to amplification of the signal after digitization.

Model	Analog	Digital
PHX004S	0-24dB	24-48dB
PHX016S	0-24dB	24-48dB



<b>Model</b>	<b>Analog</b>	<b>Digital</b>
PHX023S	0-24dB	24-48dB
PHX032S	0-24dB	24-48dB
PHX050S	0-24dB	24-48dB
PHX050S-(P/Q)	0-24dB	24-48dB
PHX064S	0-24dB	24-48dB
PHX089S	0-24dB	24-48dB
PHX120S	0-24dB	24-48dB
PHX122S	0-27dB	NA
PHX200S	0-27dB	NA

The following pseudocode demonstrates setting Gain to 12 dB:

```

1 // Connect to camera
2 // Get device node map
3 GainAuto = Off;
4 Gain = 12;
```

## Color Processing #

The PHX089S camera is equipped with a debayering core within the image processing pipeline. The color processing core enables the camera to output a color processed image format in addition to the unprocessed Bayer-tiled image. Currently the camera supports the RGB8 pixel format which outputs 8-bits of data per color channel for a total of 24-bits per pixel. Due to the amount of bits per pixel, the total image size for RGB8 would be 3 times larger when compared to an 8-bit image. This increase in image data size per frame would result in a reduction of average frame rate for the camera.



The following pseudocode demonstrates configuring the camera to RGB8 pixel format:

```
1 // Connect to camera  
2 // Get device node map  
3 PixelFormat = PixelFormat_RGB8;
```

## White Balance #

The white balance module aims to change the balance between the Red, Green and Blue channels such that a white object appears white in the acquired images. LUCID Vision Labs cameras allow for manual white balance adjustment by the user, or automatic white balance adjustment based on statistics of previously acquired frames. Different external illuminations and different sensors may render acquired images with color shift. The White Balance module allows the user to correct for the color shift by adjusting gain value of each color channel.

LUCID Vision Labs offers two types of white balance algorithm as described below. Both methods below allow for user controlled anchor points or reference points, from which multipliers are computed for each channel. The different anchor points are summarized below.

Anchors	Information
Min	The lowest luminance channel is used as reference while other channels are adjusted to match it. There is no chance of overflowing the pixels, however the image is darkened.
Max	The highest luminance channel is used as reference while other channels are adjusted to match it. There is a chance of overflowing the pixels.



Anchors	Information
Mean	The mean value of all channels is used as reference while all channels are adjusted to match the mean. There is a smaller chance of overflowing.
Green	Green channel is used as the reference while the Red and Blue are adjusted.

## Grey World

A grey world assumes that the average of all colors in an image is a neutral grey.

## White Patch

White patch has the same idea as Grey World, but only considering a section of the image (i.e. the section being the white patches). A simple way to determine such section(s) of the image is to indicate a pixel as white when R+G+B is greater than the threshold pixel value. Determining the threshold can be done using a 90% percentile of previous image. There is also a need for an additional threshold to exclude saturated pixels for better white balance adjustment.

## Look-Up Table (LUT) #

Look-Up Table also known as LUT is used for mapping 12-bit user-specified pixel values to replace 12-bit raw sensor pixel values. Users input values for the even indices including the last index 4095 while averaging is used to calculate rest of the odd indices. So there are in total 2049 effective input entries; 2048 even (e.g., 0, 2, 4, ..., 4092, 4094) + 1 odd (4095). Index value 0 correspond to black color while the index value 4095 correspond to white color.

To build a LUT, users input index values (e.g., 0, 2, 4, ..., 4092, 4094, 4095) that need to be replaced in the LUTIndex field and the corresponding new value in the LUTValue field. For the odd index values in the gap (e.g., 1,3,5 ...,4089, 4091,



4093), their mapped value is calculated by taking the average of their neighbor mapped values (e.g., If the input mappings are LUTIndex = 1090 → LUTValue = 10 and LUTIndex = 1092 → LUTValue = 20, then the mapped value of LUTIndex = 1091 will be LUTValue = 15)



To reset Look-Up Table, please execute **LUTReset** command.

The following pseudocode demonstrates replacing black pixel values with white pixel values:

```

1 // Connect to camera
2 // Get device node map
3
4 LUTEnable = true;
5 LUTIndex = 0; //Pixel value to be replaced, in this case black
6 LUTValue = 4095; //New pixel value, in this case white
7 LUTSave();
```

## Gamma #

The gamma control allows the optimization of brightness for display. LUCID implements Gamma using GenICam standard, that is

$$X = Y^{\text{Gamma}}$$

where,

- X = New pixel value;  $0 \leq X \leq 1$
- Y = Old pixel value;  $0 \leq Y \leq 1$
- Gamma = Pixel intensity:  $0.2 \leq \text{Gamma} \leq 2$

Y in the Gamma formula is scaled down to [0-1] from original pixel range which results in a pixel range of [0-1] for X. As an example, for 12-bit pixel formats, this would mean scaling down pixel range from [0-4095] to [0-1] and for 16-bit pixel formats, this would mean scaling down pixel range from [0-65535] to [0-1].



The camera applies gamma correction values to the intensity of each pixel. In general, gamma values can be summarized as follows:

- Gamma = 1: brightness is unchanged.
  - $1 \leq \text{Gamma} \leq 2$ : brightness decreases.
  - $0.2 \leq \text{Gamma} \leq 1$ : brightness increases.
- 

## Color Space Conversion and Correction #

The Color Space Conversion control allows the user to convert from RGB color space to another color space such as YUV. The conversion is done in a linear manner as shown in the following equation.

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} Gain_{00} & Gain_{01} & Gain_{02} & Offset_0 \\ Gain_{10} & Gain_{11} & Gain_{12} & Offset_1 \\ Gain_{20} & Gain_{21} & Gain_{22} & Offset_2 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \\ 1 \end{bmatrix}$$

The color correction function allows the user to choose between a few preset values or user-configurable matrix values. The color correction is done by allowing the multiplication of a  $3 \times 3$  matrix to the  $3 \times 1$  matrix containing R, G and B pixel values to achieve a more desirable R', G' and B' values. The specific mathematical procedure can be represented by the following.

$$\begin{bmatrix} Gain_{00} & Gain_{01} & Gain_{02} \\ Gain_{10} & Gain_{11} & Gain_{12} \\ Gain_{20} & Gain_{21} & Gain_{22} \end{bmatrix} \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix} + \begin{bmatrix} Offset_0 \\ Offset_1 \\ Offset_2 \end{bmatrix} = \begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix}$$

- `ColorTransformationEnable` is a node that indicates whether the conversion matrix of the color space conversion module is used or bypassed.
- When the pixel format is YUV or YCbCr, `ColorTransformationSelector` is displayed as `RGBtoYUV`. When the pixel format is Mono, `ColorTransformationSelector` is displayed as `RGBtoY`.



- ColorTransformationValueSelector chooses which coefficient in the conversion matrix and the coefficient value is shown in ColorTransformationValue. In Mono, YUV or YCbCr pixel formats, ColorTransformationValue is read only.

---

## Image Format Controls #

The PHX089S camera is equipped with the following image format control capabilities.

---

### Region of Interest (ROI) #

The region of interest feature allows you to specify which region of the sensor is used for image acquisition. This feature allows a custom width and height for image size and a custom X and Y offset for image position. The width and height must be a multiple of the minimum width and height values allowed by the camera.

The following pseudocode demonstrates configuring the camera to use a region of interest of 200×200 at offset (250,150):

```
1 // Connect to camera
2 // Get device node map
3 OffsetX = 250;
4 OffsetY = 150;
5 Width = 200;
6 Height = 200;
```



If the camera is acquiring images, AcquisitionStop must be called before changing region of interest settings.



## Binning #

The PHX089S camera supports binning in which columns and/or rows of pixels are combined to achieve a reduced overall image size without changing the image's field of view. This feature may result in an increase of camera's frame rate.

The binning factor indicates how many pixels in the horizontal and vertical axis are combined. For example, when applying 2×2 binning, which is 2 pixels in the horizontal axis and 2 pixels in the vertical axis, 4 pixels combine to form 1 pixel. The resultant pixel values can be summed or averaged.

When binning is used, the settings of the image width and height will be affected. For example, if you are using a camera with sensor resolution of 2448 x 2048 and apply 2×2 binning, the effective resolution of the resultant image is reduced to 1224 x 1024. This can be verified by checking the Width and Height nodes.

The following pseudocode demonstrates configuring binning on the camera:

```
1 // Connect to camera
2 // Get device node map
3
4 BinningSelector = Digital;    // Digital binning is performed by FPGA, so
5 BinningHorizontalMode = Sum; // Binned horizontal pixels will be summed
6 BinningVerticalMode = Sum;   // Binned vertical pixels will be summed
7
8 BinningHorizontal = 2; // Set Horizontal Binning by a factor of 2
9 BinningVertical = 2;   // Set Vertical Binning by a factor of 2
10
11 // Resulting image is 1/4 of original size
```



 When horizontal binning is used, horizontal decimation (if supported) is not available. When vertical binning is used, vertical decimation (if supported) is not available.

 If the camera is acquiring images, AcquisitionStop must be called before adjusting binning settings.

## Decimation #

The PHX089S camera supports decimation in which columns and/or rows of pixel are skipped to achieve reduced overall image size without changing the image's field of view. This feature is also known as "subsampling" due to the smaller sample size of pixels the camera transmits. This feature may result in an increase of camera's frame rate.

When decimation is used, the settings of the image width and height will be effected. For example, if you are using a camera with sensor resolution of 2448 x 2048. When horizontal and vertical decimation are both set to 2, the effective resolution of the resultant image is reduced to 1224 x 1024. This can be verified by checking the Width and Height nodes.

The following pseudocode demonstrates configuring decimation on the camera:

```
1 // Connect to camera
2 // Get device node map
3
4 DecimationHorizontal = 2; //Set Horizontal Decimation by factor of 2
5 DecimationVertical = 2; // Set Vertical Decimation by factor of 2
6
7 // Resulting image is 1/4 of original size
```





When horizontal decimation is used, horizontal binning (if supported) is not available. When vertical decimation is used, vertical binning (if supported) is not available.



If the camera is acquiring images, `AcquisitionStop` must be called before adjusting decimation settings.

## Horizontal and Vertical Flip #

This feature allows the camera to flip the image horizontally and vertically. The flip action occurs on the camera before transmitting the image to the host.

The following pseudocode demonstrates configuring the camera to flip both horizontal and vertical axes:

```
1 // Connect to camera  
2 // Get device node map  
3 ReverseX = True;  
4 ReverseY = True;
```

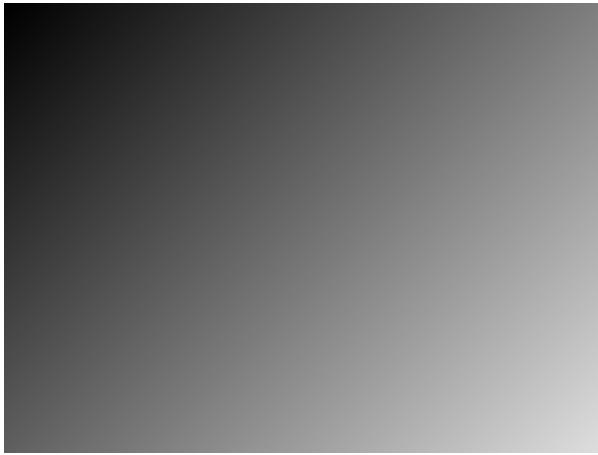


If the camera is acquiring images, `AcquisitionStop` must be called before changing horizontal or vertical flip.

## Test Pattern #

The camera outputs a FPGA-generated pattern when test pattern is enabled.





---

## Digital IO #

The Phoenix Digital IO controls input and output lines that can be utilized with external circuitry for synchronization with other devices. An example use of an input line is to allow the camera to take an image upon receipt of an internal software signal or an external pulse (rising or falling edge). An example use of an output line is to fire a pulse when the camera starts integration for the duration of the current ExposureTime value.

The Digital IO lines correspond to the Phoenix GPIO pins. Please consult the GPIO Cable section for more information on the required cable for the GPIO connector and the GPIO Characteristics section for a GPIO pinout diagram.

---

### Configuring an Input Line #

When a Digital IO line is set to Input, the line can accept external pulses. To trigger the camera upon receipt of an external pulse, the camera must also have trigger mode enabled.

The following pseudocode demonstrates enabling trigger mode and setting Line0 as the trigger input source.



```
1 // Connect to camera
2 // Get device node map
3
4 // Choose Line0 and set it to Input
5 LineSelector = Line0;
6 LineMode = Input;
7
8 TriggerMode = On;
9 TriggerSelector = FrameStart;           // Trigger signal starts a frame
10 TriggerSource = Line0;                 // External signal to be expected
11 TriggerActivation = FallingEdge;       // Camera will trigger on the falling edge
12
13 AcquisitionStart();
14
15 // Acquire images by sending pulses to Line0
```



It is also possible to set software as the input source. This will enable the camera to trigger upon a software signal. Note this mechanism may not be as accurate as using an external trigger source.

The following pseudocode demonstrates enabling trigger mode and setting a software trigger source.

```
1 // Connect to camera
2 // Get device node map
3
4 TriggerMode = On;
5 TriggerSelector = FrameStart;           // Trigger signal starts a frame
6 TriggerSource = Software;              // Software signal will trigger
7
8 AcquisitionStart();
9
10 TriggerSoftware();                   // Execute TriggerSoftware to start acquisition
```



## TriggerOverlap #

By default, the PHX089S will reject input pulses until the last triggered image has completed the readout step on the sensor. This may limit the maximum achievable trigger frequency on some cameras when compared to maximum



non-triggered FrameRate.

To address this situation, the PHX089S also supports TriggerOverlap functionality. When TriggerOverlap is enabled, this allows the camera to accept an input pulse before the readout step is complete. This allows the camera to be triggered at frequencies closer to the maximum non-triggered FrameRate.

Some Digital IO lines may be Input only (e.g. an opto-isolated input).

Consult the GPIO Characteristics



(<https://support.thinklucid.com/documentation/camera-specifications-2/gpio-characteristics/>) section for a GPIO pinout diagram.

## Configuring an Output Line #

When a Digital IO line is set to Output, the line can fire pulses.

The following pseudocode demonstrates setting Line1 as the output source. This sets up the camera to fire a pulse when the camera is acquiring an image.

```
1 // Connect to camera
2 // Get device node map
3
4 // Choose Line1 and set it to Output
5 LineSelector = Line1;
6 LineMode = Output;
7 LineSource = ExposureActive;           // The output pulse will match 1
8
9 AcquisitionStart();
10
11 // Acquire images and process signal from Line1
```





Opto-isolated outputs will require external circuitry to be properly signaled. Some Digital IO lines may be Output only (e.g. an opto-isolated output). Consult the GPIO Characteristics section for a GPIO pinout diagram.

## Turning on GPIO Voltage Output #

The PHX089S is capable of supplying external circuits with power through the  $V_{DD}$  line. By default this line is turned off. The following pseudocode demonstrates enabling the GPIO  $V_{DD}$  line to output 2.5V.

```
1 // Connect to camera
2 // Get device node map
3
4 // Choose Line4 and set it to output external voltage
5 LineSelector = Line4;
6 VoltageExternalEnable = True;
7
8 // Continue with the rest of the program...
```



Consult the GPIO Characteristics section for a GPIO pinout diagram.

## Chunk Data #

Chunk data is additional tagged data that can be used to identify individual images. The chunk data is appended after the image data.



## Extracting the Image CRC checksum with ChunkCRC #

When the chunk data property named CRC is enabled, the camera tags a cyclic-redundancy check (CRC) checksum that is calculated against the image payload.

The following pseudocode demonstrates enabling the CRC property in chunk data and extracting the ChunkCRC chunk from acquired images:

```
1 // Connect to camera
2 // Get device node map
3 ChunkModeActive = True;
4 ChunkSelector    = CRC;
5 ChunkEnable      = True;
6
7 AcquisitionStart();
8
9 // Acquire image into buffer
10 // Read chunk data from received buffer
11 bufferCRC      = ChunkCRC;           // CRC checksum for image payload
```



If the camera is acquiring images, AcquisitionStop must be called before enabling or disabling chunk data.

## Transfer Control #

Transfer Control allows the device to accumulate images on the on-camera buffer in a queue. The data stored in the queue, referred to as blocks, can be transmitted to the host application at a later time. The host application will be able to request the device transmit one or more block. By default, this control is disabled on the PHX089S and acquired images are automatically transmitted.



## Automatic Transfer Control #

When using Automatic Transfer Control mode, the transfer of blocks to the host are controlled by the device's acquisition controls.

In Automatic Transfer Control mode, the TransferOperationMode is read only and set to Continuous. Using Continuous TransferOperationMode is similar to when Transfer Control is not enabled, except the host application can stop data transmission without stopping image acquisition on the device.

The following pseudocode demonstrates enabling Automatic Transfer Control mode on the camera.

```
1 // Connect to camera
2 // Get device node map
3
4 // Set desired Pixel Format, Width, Height
5
6 // Optional: Read TransferQueueMaxBlockCount to determine the number of
7 // that can be stored in the on-camera buffer with the current device se
8
9 TransferControlMode = Automatic;
10 TransferQueueMode = FirstInFirstOut;
11
12 AcquisitionStart();           // When in Automatic Transfer Control Mo
13                         // AcquisitionStart command automatically
14
15 // Acquire images
16 // Images will be accumulating into the on-camera buffer
17 // Blocks will be transmitted to the host system automatically until Tr
18
19 // Retrieve images on host system
```



In Automatic Transfer Control mode, the following commands are available:

- **TransferPause** to pause transfer of blocks without executing TransferStop or AcquisitionStop.
- **TransferResume** to resume transfer of blocks after TransferPause has been called.
- **TransferAbort** to abort transfer of blocks.
- **TransferStop** to stop transfer of blocks.

When blocks are not transmitted from the on-camera buffer to the host quickly enough and the amount of blocks to be stored in the buffer exceeds TransferQueueMaxBlockCount, the new images will be dropped. Set TransferStatusSelector to **QueueOverflow** and read TransferStatus to determine if a block is lost.

## UserControlled Transfer Control #

When using UserControlled Transfer Control mode, the transfer of blocks to the host are controlled by the host application. Using MultiBlock TransferOperationMode with UserControlled Transfer Control allows the host application to specify when to transmit data from the device.

The following pseudocode demonstrates enabling UserControlled Transfer Control mode with MultiBlock TransferOperationMode on the camera. It also demonstrates transferring 2 blocks from the on-camera buffer for each transmission operation.



```
1 // Connect to camera
2 // Get device node map
3
4 // Set desired PixelFormat, Width, Height
5
6 // Optional: Read TransferQueueMaxBlockCount to determine the number of
7 // that can be stored in the on-camera buffer with the current device settings
8
9 TransferControlMode = UserControlled;
10 TransferQueueMode = FirstInFirstOut;
11
12 TransferOperationMode = MultiBlock;      // Transmit the number of blocks
13 TransferBlockCount = 2;                  // This should be less than or equal to TransferQueueMaxBlockCount
14
15 AcquisitionStart();
16
17 // Acquire images
18 // Images will be accumulating into the on-camera buffer
19
20 // Read TransferQueueCurrentBlockCount
21 If (TransferQueueCurrentBlockCount > 0)
22 {
23     // Transmit blocks to host system
24     TransferStart();
25
26     // Retrieve block 1 on host system
27     // Retrieve block 2 on host system
28
29     TransferStop();
30 }
```



In UserControlled Transfer Control mode, the following commands are available:

- **TransferPause** to pause transfer of blocks without executing TransferStop or AcquisitionStop.
- **TransferResume** to resume transfer of blocks after TransferPause has been called.
- **TransferAbort** to abort transfer of blocks.
- **TransferStop** to stop transfer of blocks.

 When blocks are not transmitted from the on-camera buffer to the host quickly enough and the amount of blocks to be stored in the buffer exceeds TransferQueueMaxBlockCount, the new images will be dropped. Set TransferStatusSelector to **QueueOverflow** and read TransferStatus to determine if a block is lost.



To turn off the Transfer Control mechanism, set the TransferControlMode to Basic.



If the camera is acquiring images, AcquisitionStop must be executed before changing Transfer Control properties.

## Event Control #

Events are notifications generated by the camera to inform host application about internal updates. Event Control is a mechanism that is used to synchronize the camera with host application using these events.



EventSelector can be used to select the event for turning notifications on or off. By default, EventNotification for all the events are set to Off except for Test event. This is because EventNotification for Test event is always set to On and cannot be changed.

The following pseudocode demonstrates enabling Event Control on the camera.

```
1 // Connect to camera  
2 // Get device node map  
3  
4 InitializeEvents();  
5  
6 RegisterCallback();  
7  
8 GenerateEvent();  
9  
10 // Wait on event  
11  
12 DeregisterCallback();  
13  
14 DeinitializeEvents();
```



Events available on PHX089S are as follow:

## Exposure Start #

Exposure Start event occurs when camera starts exposing the sensor to capture a frame. Following data is sent by the camera at Exposure Start event:

- **EventExposureStart:** Unique identifier of the Exposure Start type of Event.
- **EventExposureStartTimestamp:** Unique timestamp of the Exposure Start Event.
- **EventExposureStartFrameID:** Unique Frame ID related to Exposure Start Event.

## Exposure End #



Exposure End event occurs when camera has finished exposing the sensor.

Following data is sent by the camera at Exposure End event:

- **EventExposureEnd:** Unique identifier of the Exposure End type of Event.
  - **EventExposureEndTimestamp:** Unique timestamp of the Exposure End Event.
  - **EventExposureEndFrameID:** Unique Frame ID related to Exposure End Event.
- 

## Test #

Test event occurs when TestEventGenerate node is executed. Test event is mainly used to confirm that camera is generating events. Following data is sent by the camera at Test event:

- **EventTest:** Unique identifier of the Test type of Event.
  - **EventTestTimestamp:** Unique timestamp of the Test Event.
- 

## Counter and Timer Control #

### Counter #

**Counters** can be used to keep a running total of an internal event. The Counter value can be reset, read, or written at any time.

The following nodes are available to configure the Counter increment:

- **CounterSelector:** The Counter control to be used
- **CounterEventSource:** The event source that will increment the Counter
- **CounterEventActivation:** The activation mode used for **CounterEventSource** (e.g. RisingEdge)
- **CounterTriggerSource:** The source that will start the Counter



The following nodes are available to configure the Counter reset:

- **CounterResetSource**: The signal that will reset the Counter
- **CounterResetActivation**: The signal used to activate **CounterResetSource**
- **CounterReset**: The command will reset CounterValue and set the last value to **CounterValueAtReset**

The following nodes are available to read or control the Counter:

- **CounterValue**: The current value of the Counter
- **CounterValueAtReset**: The value of the Counter before it was reset
- **CounterDuration**: The value to count before setting **CounterStatus = CounterCompleted**
- **CounterStatus**: The status of the Counter

The following **CounterStatus** modes are available:

- **CounterIdle**: Counter control is not enabled
- **CounterTriggerWait**: Counter is waiting to be activated
- **CounterActive**: Counter is counting until the value specified in **CounterDuration**
- **CounterCompleted**: Counter has incremented to value set in **CounterDuration**
- **CounterOverflow**: **CounterValue** has reached its maximum possible value



When **CounterValue = CounterDuration**, the counter stops counting until a **CounterReset** or a new **CounterTriggerSource** occurs.



Changing **CounterDuration** will issue a **CounterReset** to **CounterValue**.

## Use Counter to Count Rising Edges #



The following pseudocode demonstrates using a Rising Edge counter event to track the number of rising edges signals received on Line 0. A second counter is enabled to track the number of Exposure Start events that occurred to indicate the number of images captured.

```
1 // Connect to camera
2 // Get device node map
3
4 // Use Counter0 to count rising edges received on Line0
5 CounterSelector = Counter0;
6 CounterEventSource = Line0;
7 CounterEventActivation = RisingEdge;
8 CounterTriggerSource = AcquisitionStart;      // Start this counter at t1
9 CounterDuration = 1;
10 CounterValue = 1;                           // Start at CounterValue =
11
12 // Use Counter1 to count ExposureStart events occurring on camera
13 CounterSelector = Counter1;
14 CounterEventSource = ExposureStart;
15 CounterEventActivation = RisingEdge;
16 CounterTriggerSource = AcquisitionStart;      // Start this counter at t1
17 CounterDuration = 1;
18 CounterValue = 1;                           // Start at CounterValue =
19
20 // Enable Rising Edge trigger on Line0
21 TriggerSelector = FrameStart;
22 TriggerSource = Line0;
23 TriggerActivation = RisingEdge;
24 TriggerMode = On;
25
26 AcquisitionStart();
27
28 // Acquire images by sending pulses to Line0
29
30 AcquisitionStop();
31
32 // To read rising edge pulses received:
33 // CounterSelector = Counter0
34 // Read CounterValue
35
36 // To read images captured:
37 // CounterSelector = Counter1
38 // Read CounterValue
39
40 // CounterValue for Counter0 should equal CounterValue for Counter1. Ij
41
42
```



## Timer #

**Timers** can be used to measure the duration of internal or external signals.

The following nodes are available to configure the Timer:

- **TimerSelector**: The Timer control to be used
- **TimerTriggerSource**: The source that will trigger the Timer
- **TimerTriggerActivation**: The activation mode used for **TimerTriggerSource** (e.g. RisingEdge)
- **TimerDuration**: The duration of the Timer pulse in microseconds
- **TimerDelay**: The duration of delay to apply before starting the Timer upon receipt of a TimerTriggerActivation
- **TimerReset**: Resets the Timer control
- **TimerValue**: Reads or writes the current value of the Timer
- **TimerStatus**: The status of the Timer

The following TimerStatus modes are available:

- **TimerIdle**: Timer control is not enabled
- **TimerTriggerWait**: Timer is waiting to be activated
- **TimerActive**: Timer is counting for the specified duration in **TimerDuration**
- **TimerCompleted**: Timer has reached the duration specified in **TimerDuration**



When **TimerStatus = TimerCompleted**, the Timer signal becomes active low until a **TimerReset** or a new **TimerTriggerSource** occurs.

## Use Timer to Delay a Line Output Pulse #



The following pseudocode demonstrates using a Timer to delay a Line output pulse relative to the ExposureStart event of the image.

```
1 // Connect to camera
2 // Get device node map
3
4 // Set up Line2 for Timer pulse output:
5 LineSelector = Line2;
6 LineMode = Output;
7 LineSource = Timer0Active;           // The Line output will be the
8
9 // Use Timer0 to configure your Line2 pulse:
10 TimerSelector = Timer0;
11 TimerTriggerSource = ExposureStart;
12 TimerTriggerActivation = RisingEdge;
13 TimerDelay = 2000;                  // The Line output delay will be
14 TimerDuration = 8000;              // The Line output duration will be
15
16 AcquisitionStart();
17 // Acquire images
18 // Monitor Line2 output
```



## Use Timer to Output a Custom Pulse Width #

The following pseudocode demonstrates using a Timer output a pulse of a custom width. This can be used for a single pulse width or pulse width modulation.



```

1 // Connect to camera
2 // Get device node map
3
4 // Set up Line2 for Timer pulse output:
5 LineSelector = Line2;
6 LineMode = Output;
7 LineSource = Timer0Active; // The Line output will be the
8
9 // Use Timer0 to configure your Line2 pulse:
10 TimerSelector = Timer0;
11 TimerTriggerSource = SoftwareSignal0;
12 TimerTriggerActivation = RisingEdge;
13 TimerDelay = 0;
14 TimerDuration = 2000; // The Line output duration
15
16 // Use SoftwareSignal0 to trigger the Timer pulse
17 SoftwareSignalSelector = SoftwareSignal0;
18 SoftwareSignalPulse();

```



## Firmware Update #

Use the camera's Firmware Update page to update your camera's firmware.

Updating your camera's firmware is easy with our firmware fwa file. The firmware fwa files for each camera are located on our Downloads page (<https://thinklucid.com/downloads-hub/>).

Every Lucid camera comes with an on-board firmware update page that can be directly accessed in a web browser. If you already know your camera's IP address you can access it at <http://{your-camera-ip-address}/firmware-update.html>.



## Updating your Camera Firmware #

1. Navigate to your camera's firmware update page in your web browser (<http://{your-camera-ip-address}/firmware-update.html>).
2. Click **Choose a file** on the Firmware Update page and select your fwa file.
3. Click **Submit** on the Firmware Update page. This will start the firmware update process.

The Firmware Update page will show the progress of the update. During the update, the camera will not be accessible for control or image capture. The update may take a few minutes to complete. Please refrain from removing power from the camera during the update.

When the firmware update is complete, the camera will reboot. Once the update is finished, you can close the Firmware Update page and access the camera again.

## Device Nodes #

---

## Acquisition Control #

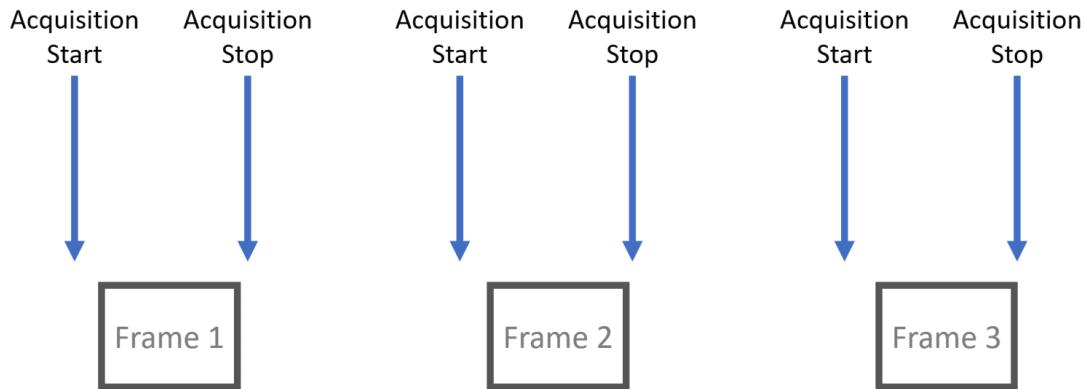
Acquisition Control contains features related to image acquisition. Triggering and exposure control functionalities are included in this section.

### Acquisition Modes #



There are 3 main types of acquisition modes – SingleFrame acquisition, MultiFrame acquisition, and Continuous acquisition.

## SingleFrame Acquisition



Under SingleFrame acquisition mode, one frame is acquired after AcquisitionStart is called. AcquisitionStop is an optional call as the acquisition process automatically stops after the single frame is acquired. During the acquisition process, all Transport Layer parameters are locked and cannot be modified.



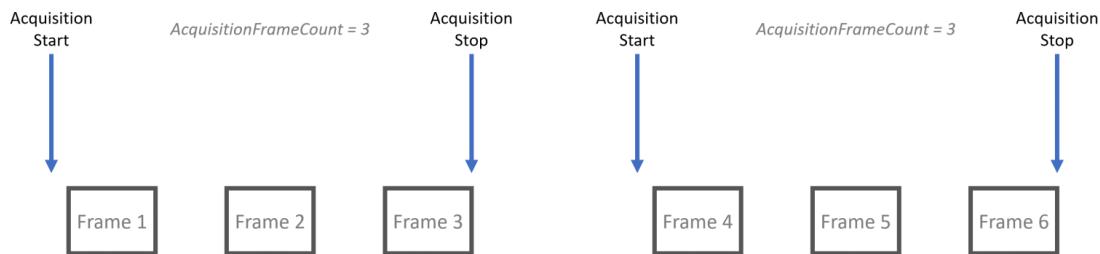
Note that if “Acquisition Stop” is executed after “Acquisition Start” but prior to a frame is available, it is possible that no frame is acquired.

The following code block demonstrates configuring camera to single frame acquisition mode.

```
1 // Connect to camera  
2 // Get device node map  
3 AcquisitionMode = SingleFrame;
```



## MultiFrame Acquisition



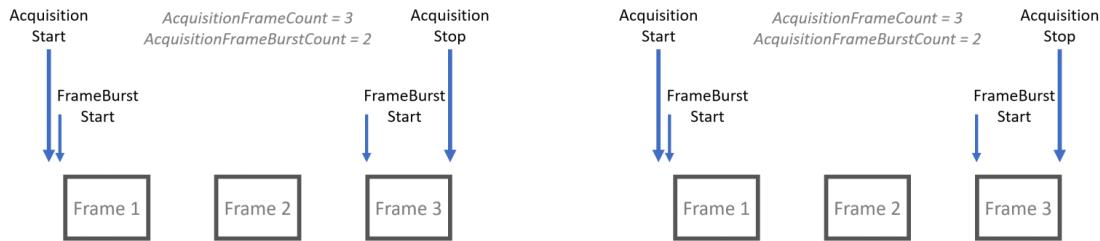
Under MultiFrame acquisition mode, frames are acquired once `AcquisitionStart` is called. The number of frames acquired is dictated by the parameter `AcquisitionFrameCount`. During the acquisition process, all Transport Layer parameters are locked and cannot be modified.



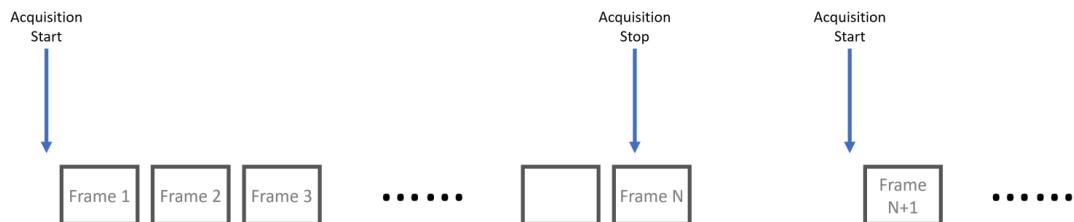
Note that `AcquisitionStop` is optional under this acquisition mode.

## MultiFrame with FrameBurstStart

A burst of frames is defined as a capture of a group of one or many frames within an acquisition. This can be achieved with MultiFrame acquisition mode as demonstrated in the diagram below. Note in the diagram, the second `FrameBurstStart` in each acquisition sequence only results in 1 frame acquired since `AcquisitionFrameCount` is set to a value of 3.



## Continuous Acquisition



Under Continuous acquisition mode, frames are acquired once `AcquisitionStart` is called. Frames are acquired until `AcquisitionStop` is called. During the acquisition process, all Transport Layer parameters are locked and cannot be modified.



Note that if `AcquisitionStop` is called during the last frame then the acquisition sequence will stop after the current frame finishes.

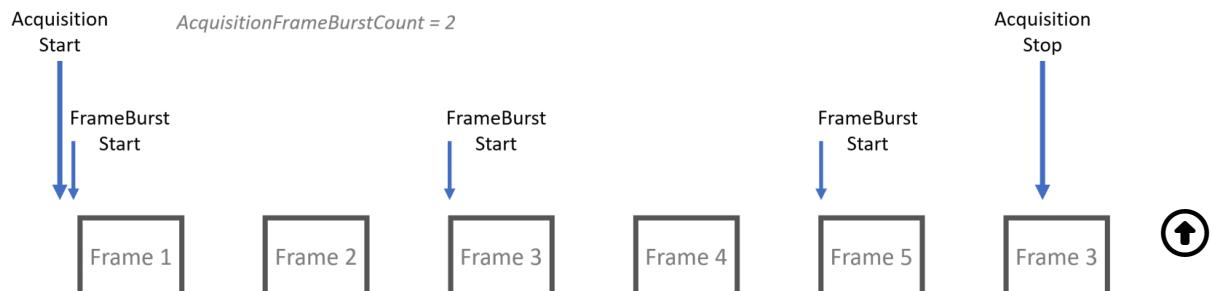
The following pseudocode demonstrates configuring the camera to Continuous acquisition mode:

```

1 // Connect to camera
2 // Get device node map
3 AcquisitionMode = Continuous;
  
```

## Continuous Acquisition with FrameBurstStart

A burst of frames is defined as a capture of a group of one or many frames within an acquisition. This can be achieved with Continuous acquisition mode as below. Note that when `AcquisitionStop` is called, the current frame will need to be finished.



<b>Node Name</b>	<b>Description</b>
AcquisitionMode	Specifies the acquisition mode of the current device. It helps determine the number of frames to acquire during each acquisition sequence.
AcquisitionStart	Start the acquisition sequence for the current device.
AcquisitionStop	Stop the acquisition sequence for the current device.
AcquisitionFrameCount	This node specifies the number of frames to be acquired under MultiFrame AcquisitionMode.
AcquisitionBurstFrameCount	This parameter is ignored if AcquisitionMode is set to SingleFrame. This feature is also constrained by AcquisitionFrameCount if AcquisitionMode is MultiFrame.
AcquisitionFrameRate	Specifies the frequency in which frames are acquired. Note that TriggerMode must be off for this parameter to take effect.
AcquisitionFrameRateEnable	Controls if the AcquisitionFrameRate feature is writable and used to control the acquisition rate.
AcquisitionLineRate	Controls the rate (in Hertz) at which the Lines in a Frame are captured.
TriggerSelector	This node selects the specific trigger type to configure.
TriggerMode	Controls the On/Off status of the current trigger.
TriggerSoftware	Executing this will generate a software trigger signal. Note that current TriggerSource must be set to Software.



<b>Node Name</b>	<b>Description</b>
TriggerSource	This node specifies the source of the trigger. It can be a software internal signal or a physical input hardware signal.
TriggerActivation	This node specifies the state in which trigger is activated.
TriggerOverlap	Specifies the type of trigger overlap permitted with the previous frame or line. This defines when a valid trigger will be accepted (or latched) for a new frame or a new line.
TriggerLatency	Enables low latency trigger mode.
TriggerDelay	Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.
TriggerArmed	Specifies whether the trigger is armed. If the trigger is not armed, triggers will be ignored.
ExposureTime	Controls the device exposure time in microseconds (us).
ExposureTimeRaw	Reports the device raw exposure time value.
ShortExposureEnable	Sets to Short Exposure Mode.
ExposureAuto	Sets the automatic exposure mode.
TargetBrightness	Sets the target brightness in 8-bit.
ExposureAutoAlgorithm	Controls the auto exposure algorithm.
ExposureAutoDamping	Controls the auto exposure damping factor in percent. Bigger values converge faster but have higher chance of oscillating.
ExposureAutoDampingRaw	Controls the auto exposure damping factor raw value. Bigger values converge faster but have higher chance of oscillating.



<b>Node Name</b>	<b>Description</b>
CalculatedMedian	Reports the current image exposure median value.
CalculatedMean	Reports the current image exposure average value.
AutoExposureAOI	Category for auto exposure AOI features.

## Auto Exposure AOI #

<b>Node Name</b>	<b>Description</b>
AutoExposureAOIEnable	Controls auto exposure AOI enable (1) or disable (0).
AutoExposureAOIWidth	Controls auto exposure AOI width relative to user AOI.
AutoExposureAOIHeight	Controls auto exposure AOI height relative to user AOI.
AutoExposureAOIOffsetX	Controls auto exposure AOI offset X relative to user AOI.
AutoExposureAOIOffsetY	Controls auto exposure AOI offset Y relative to user AOI.

## Action Control #



<b>Node Name</b>	<b>Description</b>
ActionUnconditionalMode	Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.
ActionQueueSize	Indicates the size of the scheduled action commands queue.
ActionDeviceKey	Provides the device key that allows the device to check the validity of action commands.
ActionSelector	Selects to which Action Signal further Action settings apply.
ActionGroupKey	Provides the key that the device will use to validate the action on reception of the action protocol message.
ActionGroupMask	Provides the mask that the device will use to validate the action on reception of the action protocol message.

## Analog Control #

Analog Control contains features that describe how to influence the analog features of an image, such as gain, black level, and gamma. Features related to white balance are not available on monochrome cameras.

<b>Node Name</b>	<b>Description</b>
GainSelector	Selects which Gain is controlled by the various Gain features.
Gain	Controls the selected Gain as an absolute physical value.
GainRaw	Reports the gain raw value.



<b>Node Name</b>	<b>Description</b>
GainAuto	Sets the automatic gain control mode.
BlackLevelSelector	Selects which BlackLevel is controlled by the various Black Level features.
BlackLevel	Controls the BlackLevel as an absolute physical value.
BlackLevelRaw	Controls the raw BlackLevel value.
BalanceRatioSelector	Selects which BalanceRatio is controlled by the various Balance Ratio features.
BalanceRatio	Controls the selected BalanceRatio as an absolute physical value. This is an amplification factor applied to the video signal.
BalanceWhiteAuto	Controls the mode for automatic white balancing between the color channels. The white balancing ratios are automatically adjusted.
BalanceWhiteEnable	Activates balance white features.
BalanceWhiteAutoAnchorSelector	Controls which type of statistics are used for BalanceWhiteAuto.
AwbWhitePatchEnable	Controls if the white patch algorithm is used for BalanceWhiteAuto.
AwbStatsFrameCount	Controls how many frames are used for collecting statistics for BalanceWhiteAuto.
GammaEnable	Controls the selected balance ratio as an absolute physical value. This is an amplification factor applied to the video signal.
Gamma	Controls the gamma correction of pixel intensity.



# Chunk Data Control #

<b>Node Name</b>	<b>Description</b>
ChunkModeActive	Activates the inclusion of Chunk data in the payload of the image.
ChunkSelector	Selects which Chunk to enable or control.
ChunkEnable	Enables the inclusion of the selected Chunk data in the payload of the image.
ChunkCRC	CRC chunk data.
ChunkPixelFormat	Chunk Pixel Format chunk data.
ChunkWidth	Chunk Width chunk data.
ChunkHeight	Chunk Height chunk data.
ChunkOffsetX	Chunk Offset X chunk data.
ChunkOffsetY	Chunk Offset Y chunk data.
ChunkPixelDynamicRangeMin	Chunk Pixel Dynamic Range Min chunk data.
ChunkPixelDynamicRangeMax	Chunk Pixel Dynamic Range Max chunk data.
ChunkGain	Chunk Gain chunk data.
ChunkBlackLevel	Chunk Black Level chunk data.
ChunkExposureTime	Chunk Exposure Time chunk data.

# Counter and Timer Control #

<b>Node Name</b>	<b>Description</b>
CounterSelector	Selects which Counter to configure.
CounterEventSource	Select the events that will be the source to increment the Counter.



<b>Node Name</b>	<b>Description</b>
CounterEventActivation	Selects the Activation mode Event Source signal.
CounterTriggerSource	Selects the source to start the Counter.
CounterTriggerActivation	Selects the Activation mode of the trigger to start the Counter.
CounterResetSource	Select the signals that will be the source to reset the Counter.
CounterResetActivation	Selects the Activation mode of the Counter Reset Source signal.
CounterReset	Does a software reset of the selected Counter and starts it.
CounterValue	Reads or writes the current value of the selected Counter.
CounterValueAtReset	Reads value of the selected Counter when it was reset by a trigger or by an explicit CounterReset command.
CounterDuration	Sets the duration (or number of events) before the Counter End event is generated.
CounterStatus	Returns the current status of the Counter.
TimerSelector	Selects which Timer to configure.
TimerTriggerSource	Selects the source to start the Timer.
TimerTriggerActivation	Selects the Activation mode of the trigger to start the Timer.
TimerDuration	Sets the duration (in microseconds) of the Timer pulse.
TimerDelay	Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before stating the Timer.



<b>Node Name</b>	<b>Description</b>
TimerReset	Does a software reset of the selected Timer and starts it.
TimerValue	Reads or writes the current value of the selected Timer.
TimerStatus	Returns the current status of the Timer.

## Defect Correction Control #

<b>Node Name</b>	<b>Description</b>
DefectCorrectionEnable	Activates the defect correction feature.
DefectCorrectionCount	The number of defect corrected pixels.
DefectCorrectionIndex	The index of the defect corrected pixel to access.
DefectCorrectionPositionX	The column index of the defect corrected pixel selected by DefectCorrectionIndex.
DefectCorrectionPositionY	The row index of the defect corrected pixel selected by DefectCorrectionIndex.
DefectCorrectionGetNewDefect	Get a new defect corrected pixel to be entered.
DefectCorrectionApply	Apply the new defect corrected pixel.
DefectCorrectionRemove	Remove the defect corrected pixel selected by DefectCorrectionIndex.
DefectCorrectionSave	Save the defect corrected pixels to non-volatile memory.
DefectCorrectionRestoreDefault	Reset the defect correction to factory default.



# Device Control #

Device Control contains features that provide information of the capabilities of the device. There are also features that describe the particular device in detail. These features can be used in applications to query the capabilities of the device and report them to the end user if needed.

Below is a list of the available features that are contained in the specific category.

<b>Node Name</b>	<b>Description</b>
DeviceType	Returns the device type.
DeviceScanType	Scan type of the sensor of the device.
DeviceVendorName	Name of the manufacturer of the device.
DeviceModelName	Model of the device.
DeviceFamilyName	Identifier of the product family of the device.
DeviceManufacturerInfo	Provides additional information from the vendor about the device.
DeviceVersion	Version of the device.
DeviceFirmwareVersion	Version of the firmware in the device.
DeviceSerialNumber	Device's unique serial number.
DeviceUserID	A device ID string that is user-programmable.
DeviceSFNCVersionMajor	Major version of the Standard Features Naming Convention (SFNC) that was used to create the device's GenICam XML.
DeviceSFNCVersionMinor	Minor version of the Standard Features Naming Convention (SFNC) that was used to create the device's GenICam XML.



<b>Node Name</b>	<b>Description</b>
DeviceSFNCVersionSubMinor	Sub minor version of the Standard Features Naming Convention (SFNC) that was used to create the device's GenICam XML.
DeviceManifestEntrySelector	Manifest entry selector.
DeviceManifestXMLMajorVersion	Indicates the major version number of the GenICam XML file of the selected manifest entry.
DeviceManifestXMLMinorVersion	Indicates the minor version number of the GenICam XML file of the selected manifest entry.
DeviceManifestXMLSubMinorVersion	Indicates the sub minor version number of the GenICam XML file of the selected manifest entry.
DeviceManifestSchemaMajorVersion	Indicates the major version number of the schema file of the selected manifest entry.
DeviceManifestSchemaMinorVersion	Indicates the minor version number of the schema file of the selected manifest entry.
DeviceManifestPrimaryURL	Indicates the first URL to the GenICam XML device description file of the selected manifest entry.
DeviceManifestSecondaryURL	Indicates the second URL to the GenICam XML device description file of the selected manifest entry.
DeviceTLType	Transport Layer type of the device.
DeviceTLVersionMajor	Major version of the Transport Layer of the device.



<b>Node Name</b>	<b>Description</b>
DeviceTLVersionMinor	Minor version of the Transport Layer of the device.
DeviceTLVersionSubMinor	Sub minor version of the Transport Layer of the device.
DeviceMaxThroughput	Maximum bandwidth of the data that can be streamed out of the device.
DeviceLinkSelector	Selects which Link of the device to control. In general, the device only has one link.
DeviceLinkSpeed	Indicates the speed of transmission negotiated on the specified Link selected by DeviceLinkSelector.
DeviceLinkThroughputLimitMode	Controls if the DeviceLinkThroughputLimit is active.
DeviceLinkThroughputLimit	Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.
DeviceLinkThroughputReserve	Allocates the maximum percentage of bandwidth reserved for re-transmissions.
DeviceLinkHeartbeatMode	Activate or deactivate the selected Link's heartbeat.
DeviceLinkHeartbeatTimeout	Controls the current heartbeat timeout of the specific Link in microseconds.
DeviceLinkCommandTimeout	Indicates the command timeout of the specified Link in microseconds.
DeviceStreamChannelCount	Indicates the number of stream channels supported by the device.
DeviceStreamChannelSelector	Selects the stream channel to control.
DeviceStreamChannelType	Reports the type of the stream channel.



<b>Node Name</b>	<b>Description</b>
DeviceStreamChannelEndianness	Endianness of multi-byte pixel data for this stream.
DeviceStreamChannelPacketSize	Specifies the stream packet size, in bytes, to send on the selected channel for the device.
DeviceEventChannelCount	Indicates the number of event channels supported by the device.
DeviceCharacterSet	Character set used by all the strings of the device.
DeviceReset	Resets the device to its power up state.
DeviceFactoryReset	Resets device to factory defaults.
FirmwareUpdate	Starts a firmware update.
DeviceFeaturePersistenceStart	Indicate to the device and GenICam XML to get ready for persisting of all streamable features.
DeviceFeaturePersistenceEnd	Indicate to the device the end of feature persistence.
DeviceRegistersStreamingStart	Prepare the device for registers streaming without checking for consistency.
DeviceRegistersStreamingEnd	Announce the end of registers streaming.
DeviceIndicatorMode	Controls the behavior of the indicator LED showing the status of the Device.
DeviceTemperatureSelector	Selects the temperature sensor to read from.
DeviceTemperature	Device temperature in degrees Celsius.
DevicePressure	The internal device pressure in kilopascals.



<b>Node Name</b>	<b>Description</b>
DevicePower	Device power in Watts.
DeviceClockSelector	Selects the clock frequency to access from the device.
DeviceClockFrequency	Returns the frequency of the selected Clock.
Timestamp	Reports the current value of the device timestamp counter.
TimestampReset	Resets the current value of the device timestamp counter. Executing this command causes the timestamp counter to restart automatically.
TimestampLatch	Latches the current timestamp counter into TimestampLatchValue.
TimestampLatchValue	Returns the latched value of the timestamp counter.
DeviceUpTime	Time the device has been powered in seconds.
LinkUpTime	Time the device link has been established in seconds.

## Digital IO Control #

<b>Node Name</b>	<b>Description</b>
LineSelector	Selects the physical line (or pin) of the external device connector to configure.
LineMode	Controls if the physical Line is used to Input or Output a signal.



<b>Node Name</b>	<b>Description</b>
LineInverter	Controls the inversion of the signal of the selected input or output Line.
LineStatus	Returns the current status of the selected input or output Line.
LineStatusAll	Returns the current status of all available Line signals at time of polling.
LineSource	Selects which internal acquisition or I/O source signal to output on the selected Line. LineMode must be set to Output.
LineFormat	Controls the current electrical format of the selected physical input or output Line.
LineFilterSelector	Selects the input filter to configure. LineMode must be Input.
LineFilterWidth	Width of the filter in microseconds.
VoltageExternalEnable	Controls if the external voltage is enabled.
UserOutputSelector	Selects which bit of the User Output register will be set by UserOutputValue.
UserOutputValue	Sets the value of the bit selected by UserOutputSelector.
UserOutputValueAll	Sets the value of all the bits of the User Output register. It is subject to the UserOutputValueAllMask.
UserOutputValueAllMask	Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register. If the UserOutputValueAllMask feature is present, setting the user Output register using UserOutputValueAll will only change the bits that have a corresponding bit in the mask set to one.



# Event Control #

Event Control has the device inform the host application that an event has occurred. Below is a list of the available features that are contained in the specific category.

<b>Node Name</b>	<b>Description</b>
EventSelector	Selects which Event to signal to the host application.
EventNotification	Activate or deactivate the notification to the host application of the occurrence of the selected Event.
EventDestinationAddress	Controls the destination IP address for events.
EventExposureStartData	Category that contains all the data features related to the ExposureStart Event.
EventExposureEndData	Category that contains all the data features related to the ExposureEnd Event.
EventTestData	Category that contains all the data features related to the Event Test generated using the TestEventGenerate command.

---

## Event Exposure End Data #

<b>Node Name</b>	<b>Description</b>
EventExposureEnd	Returns the unique identifier of the ExposureEnd type of Event.
EventExposureEndTimestamp	Returns the Timestamp of the ExposureEnd Event.



<b>Node Name</b>	<b>Description</b>
EventExposureEndFrameID	Returns the unique Identifier of the Frame (or image) that generated the ExposureEnd Event.

## Event Exposure Start Data #

<b>Node Name</b>	<b>Description</b>
EventExposureStart	Returns the unique identifier of the ExposureStart type of Event.
EventExposureStartTimestamp	Returns the Timestamp of the ExposureStart Event.
EventExposureStartFrameID	Returns the unique Identifier of the Frame (or image) that generated the ExposureStart Event.

## Event Test Data #

<b>Node Name</b>	<b>Description</b>
EventTest	Returns the unique identifier of the Event Test type of event generated using the TestEventGenerate command.
EventTestTimestamp	Returns the Timestamp of the Event Test event.



# File Access Control #

<b>Node Name</b>	<b>Description</b>
FileSelector	Selects the target file in the device.
FileOperationSelector	Selects the target operation for the selected file in the device. This operation is executed when the FileOperationExecute feature is called.
FileOperationExecute	Executes the operation selected by FileOperationSelector on the selected file.
FileOpenMode	Selects the access mode in which a file is opened in the device.
FileAccessBuffer	Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
FileAccessOffset	Controls the offset of the mapping between the device file storage and the FileAccessBuffer.
FileAccessLength	Controls the length of the mapping between the device file storage and the FileAccessBuffer.
FileOperationStatus	Represents the FileOperationExecute status.
FileOperationResult	Represents the file operation result. For Read or Write operations, the number of successfully read/written bytes is returned.
FileSize	Represents the size of the selected file in bytes.
FileStorageSize	Represents the total size of the file storage selected by FileSelector in bytes.
FileStorageUsedSize	Represents the used size of the file storage selected by FileSelector in bytes.



<b>Node Name</b>	<b>Description</b>
FileStorageFreeSize	Represents the free size of the file storage selected by FileSelector in bytes.

## I2C Access Control #

<b>Node Name</b>	<b>Description</b>
I2COperationSelector	Selects the target operation for the I2C communication on the device. This operation is executed when the I2COperationExecute feature is called.
I2COperationExecute	Executes the operations selected by I2COperationSelector on the I2C communication.
I2CAccessBuffer	Defines the intermediate access buffer that allows the exchange of data between the device's I2C communication and the application.
I2CRegisterAddress	Controls the register address for the operation selected by the I2COperationSelector.
I2CRegisterAddressLength	Controls the register address length used for the I2C communication.
I2CAccessLength	Controls the number of bytes to transfer to or from the I2CAccessBuffer.
I2COperationStatus	Represents the I2COperationExecute status.
I2COperationResult	Represents the number of successfully read/written bytes for the I2C operation.
I2CSlaveDeviceAddress	Controls the device address for the slave.



<b>Node Name</b>	<b>Description</b>
I2CSlaveDeviceAddressLength	Controls the slave device address length used for the I2C communication.
I2CSclSource	Selects the physical line to use for the I2C SCL source.
I2CSdaSource	Selects the physical line to use for the I2C SDA source.

## Image Format Control #

<b>Node Name</b>	<b>Description</b>
PixelFormat	Format of the pixels provided by the device.
PixelColorFilter	Type of color filter that is applied to the image.
IspBayerPattern	Reports the image <b>Bayer</b> pattern in the ISP.
SensorWidth	Effective width of the sensor in pixels.
SensorHeight	Effective height of the sensor in pixels.
SensorShutterMode	Specifies the shutter mode.
WidthMax	Maximum width of the image in pixels.
HeightMax	Maximum height of the image in pixels.
PhysicalPixelSize	Reports the physical size of an individual pixel on the sensor.
ADCBitDepth	Reports the bit depth of the sensor's ADC.
ImagerWidth	Width of the sensor image after binning and decimation.
ImagerHeight	Height of the sensor image after binning and decimation.



<b>Node Name</b>	<b>Description</b>
ImagerOffsetX	Horizontal offset from the origin of the sensor region.
ImagerOffsetY	Vertical offset from the origin of the sensor region.
Width	Width of the image provided by the device in pixels.
Height	Height of the image provided by the device in pixels.
OffsetX	Horizontal offset from the origin to the region of interest in pixels.
OffsetY	Vertical offset from the origin to the region of interest in pixels.
PixelSize	Total size in bits of a pixel of the image.
ReverseX	Flip horizontally the image sent by the device.
ReverseY	Flip vertically the image sent by the device.
PixelDynamicRangeMin	Minimum value that can be returned during the digitization process.
PixelDynamicRangeMax	Maximum value that can be returned during the digitization process.
BinningSelector	Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.
BinningHorizontalMode	Selects how to combine the horizontal pixels together.
BinningVerticalMode	Selects how to combine the vertical pixels together.



<b>Node Name</b>	<b>Description</b>
BinningHorizontal	Number of horizontal pixels to combine together. This reduces the horizontal resolution (width) of the image. A value of 1 indicates that no horizontal binning is performed by the camera.
BinningVertical	Number of vertical pixels to combine together. This reduces the vertical resolution (height) of the image. A value of 1 indicates that no vertical binning is performed by the camera.
DecimationSelector	Selects which decimation engine is controlled by the DecimationHorizontal and DecimationVertical features.
DecimationHorizontalMode	Selects how to decimate the horizontal pixels.
DecimationVerticalMode	Selects how to decimate the vertical pixels.
DecimationHorizontal	Number of horizontal pixels to decimate. This reduces the horizontal resolution (width) of the image. A value of 1 indicates that no horizontal decimation is performed by the camera.
DecimationVertical	Number of vertical pixels to decimate. This reduces the vertical resolution (height) of the image. A value of 1 indicates that no vertical decimation is performed by the camera.
TestPattern	Selects the type of test pattern that is generated by the device as image

## LUT Control #

<b>Node Name</b>	<b>Description</b>
LUTEnable	Activates the saved LUT.



Node Name	Description
LUTSelector	Selects which LUT to control.
LUTIndex	Control the index (offset) of the coefficient to access in the selected LUT.
LUTValue	Access the value at entry LUT Index of the LUT selected by LUT Selector.
LUTValueAll	Accesses all the LUT coefficients in a single access without using individual LUT Index.
LUTSave	Save the values set through LUT Values or LUT Value All.
LUTReset	Reset the LUT Values to be linear mapping.

## Sequencer Control #

Sequencer Control allows acquisition of images with varying settings without stopping stream. A total of 8 different sets can be defined in Sequencer with each set maintaining unique values for the supported features. When Sequencer Mode is enabled, camera will cycle through these 8 sets and grab images for example with different Gain and Exposure Time. For each cycle, 8 different paths can be defined and in each path one set can point to any of other 7 sets for upcoming frame .

### Supported Features:

Current implementation supports following features in a sequencer set:

1. Gain
2. Exposure Time



# Configuring a Sequencer Set

Following instructions will explain the process of configuring sequencer with 2 sets of Exposure Time where set 0 goes to set 1 and set 1 goes back to set 0, all being triggered on Frame Start.

## **Configuring first set of sequencer:**

1. In order to start the configuration process, set SequencerConfigurationMode to "On" under Sequencer Control group
2. Ensure SequencerFeatureSelector is set to "ExposureTime"
3. Be default SequencerSetSelector is set to "0", this means that we can set a exposure time for set "0" under AcquisitionControl (the ExposureAuto node is already forced to Off at this point). Go under Acquisition Control group and set an ExposureTime of 100us for example
4. Set SequencerSetStart to "0". This means that every cycle will start from set "0".
5. SequencerPathSelector can be "0"
6. SequencerSetNext should be set to "1"
7. Set SequencerTriggerSource to "FrameStart", this will allow sequencer to change set at each Frame Start signal
8. Clicking "SequencerSetSave" will save your first set of sequencer setting with exposure time 100us

## **Now let's configure the second set of sequencer:**

1. Change SequencerSetSelector from "0" to "1" in order to start configuring set 2
2. Set SequencerSetNext to "0"
3. Go under Acquisition Control group and Set ExposureTime to another value like 50000us
4. Leave SequencerTriggerSource to "FrameStart", this will allow sequencer to change at each Frame Start signal
5. Clicking on "SequencerSetSave" will save your second set of sequencer setting with exposure time 50000us



Now turn SequencerConfigurationMode to "off", the firmware will validate your sequencer setting to ensure it is valid. SequencerMode should now be writable which allows you to set it to "On", if it's not writable then it that means sequencer set configured is invalid. After setting SequencerMode to "On" and streaming the camera, you should see flashing of images with 2 exposure times.



Note that for SingleFrame and MultiFrame acquisition modes,

AcquisitionStop must be executed before executing AcquisitionStart again.

The following pseudocode summarizes above instructions and demonstrates configuring two sets 0 and 1 with varying ExposureTime so they loop 0-1-0-1.

```
1 // Connect to camera
2 // Get device node map
3 ExposureAuto = "Off";
4 GainAuto = "Off";
5 SequencerMode = "Off";
6 SequencerConfigurationMode = "On";
7
8 // Configuring set 1
9 SequencerSetSelector = 0;
10 SequencerFeatureSelector = "ExposureTime";
11 ExposureTime = 100;
12 SequencerPathSelector = 0;
13 SequencerSetNext = 1;
14 SequencerTriggerSource = FrameStart;
15 SequencerSetSave;
16
17 // Configuring set 2
18 SequencerSetSelector = 1;
19 SequencerFeatureSelector = "ExposureTime";
20 ExposureTime = 50000;
21 SequencerPathSelector = 0;
22 SequencerSetNext = 0;
23 SequencerTriggerSource = FrameStart;
24 SequencerSetSave;
25
26 SequencerSetStart = 0
27 SequencerConfigurationMode = "Off"
28 SequencerMode = "On"
```



<b>Node Name</b>	<b>Description</b>
SequencerMode	Controls if the sequencer mechanism is active.
SequencerConfigurationMode	Controls if the sequencer configuration mode is active.
SequencerFeatureSelector	Selects which sequencer features to control.
SequencerFeatureEnable	Enables the selected feature and make it active in all the sequencer sets.
SequencerSetSelector	Selects the sequencer set to which further feature settings applies.
SequencerSetSave	Saves the current device state to the sequencer set selected by the SequencerSetSelector.
SequencerSetLoad	Loads the sequencer set selected by SequencerSetSelector in the device.
SequencerSetActive	Contains the currently active sequencer set.
SequencerSetStart	Sets the initial/start sequencer set, which is the first set used within a sequencer.
SequencerPathSelector	Selects to which branching path further path settings applies.
SequencerSetNext	Specifies the next sequencer set.
SequencerTriggerSource	Specifies the internal signal or physical input line to use as the sequencer trigger source.
SequencerTriggerActivation	Specifies the activation mode of the sequencer trigger.

## Serial Access Control #



<b>Node Name</b>	<b>Description</b>
SerialOperationSelector	Selects the target operation for the serial communication on the device. This operation is executed when the SerialOperationExecute feature is called.
SerialOperationExecute	Executes the operation selected by SerialOperationSelector on the serial communication.
SerialAccessBuffer	Defines the intermediate access buffer that allows the exchange of data between the device's serial communication and the application.
SerialAccessLength	Controls the number of bytes to transfer to or from the SerialAccessBuffer.
SerialOperationStatus	Represents the SerialOperationExecute status.
SerialOperationResult	Represents the Serial operation result. For Read or Write operations, the number of successfully read/written bytes is returned.
SerialReceivedSize	Represents the number of bytes received from or transferred to the serial communication.
SerialReceiveSource	Selects the physical line to use for the serial communication receive source.
SerialBaudRate	This feature controls the baud rate used for serial communication.
SerialParity	This feature controls the parity used for serial communication.
SerialStopBits	This feature controls the number of stop bits used for serial communication.
SerialDataBits	This feature controls the number of data bits used for serial communication.



<b>Node Name</b>	<b>Description</b>
SerialFlowControl	This feature controls the flow control used for serial communication.

## Software Signal Control #

<b>Node Name</b>	<b>Description</b>
SoftwareSignalSelector	Selects which Software Signal features to control.
SoftwareSignalPulse	Generates a pulse signal that can be used as a software trigger.

## Test Control #

<b>Node Name</b>	<b>Description</b>
TestPendingAck	Tests the device's pending acknowledge feature.
TestEventGenerate	Generates a Test Event.
TestMode	None

## Transfer Control #

Transfer Control allows the device to accumulate on the on-camera buffer in a queue. The data stored in the queue, referred to as blocks, can be transmitted to the host application at a later time. The host application will be able to request



the device transmit one or more block. By default, this control is disabled on the PHX089S and acquired images are automatically transmitted.

<b>Node Name</b>	<b>Description</b>
TransferSelector	Selects which stream transfers are currently controlled by the selected Transfer features.
TransferControlMode	Selects the control method for the transfers.
TransferOperationMode	Selects the operation mode of the transfer.
TransferBlockCount	Specifies the number of data Blocks that the device should stream before stopping.
TransferQueueMaxBlockCount	Controls the maximum number of data blocks that can be stored in the block queue of the selected stream.
TransferQueueCurrentBlockCount	Returns the number of Block(s) currently in the transfer queue.
TransferQueueMode	Specifies the operation mode of the transfer queue.
TransferStart	Starts the streaming of data blocks out of the device.
TransferStop	Stops the streaming of data Block(s).
TransferAbort	Immediately abort the streaming of data block(s).
TransferPause	Pauses the streaming of data Block(s).
TransferResume	Resumes a data Blocks streaming that was previously paused by a TransferPause command.
TransferStatusSelector	Selects which status of the transfer module to read.
TransferStatus	Reads the status of the Transfer module signal selected by TransferStatusSelector.



# Transport Layer Control #

Transport Layer Control includes a list of features related to the GigE Vision bootstrap registers and GigE Vision transport medium.

<b>Node Name</b>	<b>Description</b>
PayloadSize	Provides the number of bytes transferred for each image or chunk on the stream channel.
GigEVision	Category that contains the features pertaining to the GigE Vision transport layer of the device.
PtpControl	Category that contains the features related to the Precision Time Protocol (PTP) of the device.
ReceivedPacketResendCount	Returns the number of received packet resend commands.
ValidPacketResendCount	Returns the number of valid packet resends handled by the device.
UnavailablePacketResendCount	Returns the number of unavailable packet resends handled by the device.

---

## GigE Vision #

<b>Node Name</b>	<b>Description</b>
GevPhysicalLinkConfiguration	Controls the principal physical link configuration to use on next restart/power-up of the device.
GevCurrentPhysicalLinkConfiguration	Indicates the current physical link configuration of the device.



<b>Node Name</b>	<b>Description</b>
GevSupportedOptionSelector	Selects the GEV option to interrogate for existing support.
GevSupportedOption	Returns if the selected GEV option is supported.
GevInterfaceSelector	Selects which logical link to control.
GevMACAddress	MAC address of the logical link.
GevPAUSEFrameReception	Controls whether incoming PAUSE Frames are handled on the given logical link.
GevPAUSEFrameTransmission	Controls whether PAUSE Frames can be generated on the given logical link.
GevCurrentIPConfigurationLLA	Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.
GevCurrentIPConfigurationDHCP	Controls whether the DHCP IP configuration scheme is activated on the given logical link.
GevCurrentIPConfigurationPersistentIP	Controls whether the Persistent IP configuration scheme is activated on the given logical link.
GevCurrentIPAddress	Reports the IP address for the given logical link.
GevCurrentSubnetMask	Reports the subnet mask of the given logical link.
GevCurrentDefaultGateway	Reports the default gateway IP address to be used on the given logical link.
GevIPConfigurationStatus	Reports the current IP configuration status.



<b>Node Name</b>	<b>Description</b>
GevPersistentIPAddress	Controls the Persistent IP address for this logical link.
GevPersistentSubnetMask	Controls the persistent subnet mask associated with the Persistent IP address on this logical link.
GevPersistentDefaultGateway	Controls the persistent default gateway for this logical link.
GevDiscoveryAckDelay	Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.
GevGVCPExtendedStatusCodesSelector	Selects the GigE Vision version to control extended status codes for.
GevGVCPExtendedStatusCodes	Enables the generation of extended status codes.
GevGVCPPendingAck	Enables the generation of PENDING_ACK.
GevPrimaryApplicationSwitchoverKey	Controls the key to use to authenticate primary application switchover requests.
GevGVSPExtendedIDMode	Enables the extended IDs mode.
GevCCP	Controls the device access privilege of an application.
GevPrimaryApplicationSocket	Returns the UDP source port of the primary application.
GevPrimaryApplicationIPAddress	Returns the address of the primary application.
GevMCPHostPort	Controls the port to which the device must send messages.



<b>Node Name</b>	<b>Description</b>
GevMCDA	Controls the destination IP address for the message channel.
GevMCTT	Provides the transmission timeout value.
GevMCRC	Controls the number of retransmissions allowed when a message channel message times out.
GevMCSP	This feature indicates the source port for the message channel.
GevStreamChannelSelector	Selects the stream channel to control.
GevSCPIfaceIndex	Index of the logical link to use.
GevSCPHostPort	Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.
GevSCPSFireTestPacket	Sends a test packet. When this feature is set, the device will fire one test packet.
GevSCPSDoNotFragment	The state of this feature is copied into the 'do not fragment' bit of IP header of each stream packet.
GevSCPSPacketSize	It specifies the stream packet size, in bytes, to send on the selected channel for a GVSP transmitter or specifies the maximum packet size supported by a GVSP receiver.



<b>Node Name</b>	<b>Description</b>
GevSCPD	Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.
GevSCDA	Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.
GevSCSP	Indicates the source port of the stream channel.

## PTP Control #

<b>Node Name</b>	<b>Description</b>
PtpEnable	Enables the Precision Time Protocol (PTP).
PtpStatus	Returns the latched state of the PTP clock.
PtpSlaveOnly	Enables slave only operation of the PTP.
PtpClockAccuracy	Indicates the expected accuracy of the device PTP clock when it is the grandmaster, or in the event it becomes the grandmaster.
PtpServoStatus	Returns the latched state of the clock servo.
PtpDataSet	Reports the current value of the PTP's timestamp.
PtpDataSetLatch	Latches the current values from the device's PTP clock data set.
PtpDataSetLatchValue	Returns the latched value of the PTP clock.



<b>Node Name</b>	<b>Description</b>
PtpOffsetFromMaster	Returns the latched offset from the PTP master clock in nanoseconds. If PtpOffsetFromMaster returns -1 then the synchronization has not started.
PtpClockID	Returns the latched clock ID of the PTP device.
PtpParentClockID	Returns the latched parent clock ID of the PTP device.
PtpGrandmasterClockID	Returns the latched grandmaster clock ID of the PTP device.

## User Set Control #

<b>Node Name</b>	<b>Description</b>
UserSetSelector	Selects the feature User Set to load, save, or configure.
UserSetLoad	Loads the User Set specified by UserSetSelector to the device and makes it active.
UserSetSave	Save the User Set specified by UserSetSelector to the non-volatile memory of the device.
UserSetDefault	Selects the feature User Set to load and make active by default when the device is reset.
UserSetFeatureSelector	Selects which individual User Set feature to control.
UserSetFeatureEnable	Enables the selected feature and make it active in all the User Sets.



© 2026 LUCID Vision Labs Inc.

Looking to purchase our cameras?

Visit the LUCID Webstore at [thinklucid.com](https://thinklucid.com) (<https://thinklucid.com>)

