

INSTRUCTION MANUAL

ITALA

GigE Vision Cameras



Contents

| | | |
|-------|---------------------------------------|----|
| 1 | GENERAL INFORMATION | 5 |
| 1.1 | Disclaimer | 5 |
| 1.2 | Intended use | 5 |
| 1.3 | Forbidden use | 5 |
| 1.4 | Revisions | 7 |
| 1.5 | Ordering code | 9 |
| 2 | WARRANTY AND CERTIFICATIONS | 11 |
| 2.1 | Warranty | 11 |
| 2.2 | CE Declaration | 11 |
| 2.3 | FCC Declaration | 11 |
| 2.4 | KC Certification | 12 |
| 2.5 | Shock and vibrations | 12 |
| 2.5.1 | ITALA G - G.EL | 12 |
| 2.5.2 | ITALA G.IP | 13 |
| 2.6 | RoHS, REACH and WEEE | 13 |
| 3 | INTRODUCTION | 14 |
| 3.1 | Manual and conventions | 14 |
| 3.2 | Product identification data | 14 |
| 3.3 | Storage and use conditions | 15 |
| 3.3.1 | Storage conditions | 15 |
| 3.3.2 | Operating conditions | 15 |
| 3.4 | Cleaning and maintenance | 16 |
| 4 | GETTING STARTED | 17 |
| 4.1 | Overview | 17 |
| 4.2 | Hardware installation | 17 |
| 4.2.1 | Camera installation | 17 |

| | | |
|-------|--|----|
| 4.2.2 | Lens | 17 |
| 4.2.3 | Ethernet cable | 19 |
| 4.2.4 | GPIO cable | 20 |
| 4.2.5 | Liquid lens | 21 |
| 4.3 | System configuration | 22 |
| 4.3.1 | System requirements | 22 |
| 4.3.2 | Camera Driver | 22 |
| 4.3.3 | Install Driver on Windows | 23 |
| 4.3.4 | Install Driver on Ubuntu | 24 |
| 4.3.5 | Network and configuration | 25 |
| 4.3.6 | Bandwidth management | 25 |
| 4.4 | Itala SDK | 26 |
| 4.5 | Using the camera with Itala API | 31 |
| 4.5.1 | Itala SDK documentation | 31 |
| 4.6 | Using the camera with third party software | 31 |
| 4.7 | Using the camera with Itala View | 31 |
| 4.7.1 | Tabs and panels | 32 |
| 4.7.2 | IP configurator | 33 |
| 4.7.3 | Firmware update | 34 |
| 4.7.4 | LUT wizard | 37 |
| 4.7.5 | Defective pixels correction wizard | 38 |
| 4.7.6 | Color correction wizard | 40 |

5 TECHNICAL SPECIFICATIONS 43

| | | |
|-------|---|----|
| 5.1 | Technical specifications | 43 |
| 5.2 | Electrical specifications | 46 |
| 5.3 | Sensor optical response | 48 |
| 5.4 | Optical filters | 52 |
| 5.5 | Mechanical specifications | 54 |
| 5.5.1 | Dimensional drawings | 54 |
| 5.5.2 | GigE Vision mechanical requirements | 56 |
| 5.5.3 | Sensor centering data | 56 |
| 5.6 | Connectors and pinout | 57 |
| 5.7 | I/O circuitry | 59 |
| 5.7.1 | Opto Isolated Input | 59 |
| 5.7.2 | Opto Isolated Output | 59 |
| 5.8 | LED and indicators | 60 |

| | | |
|--------|--|-----|
| 6.1 | Device Control | 61 |
| 6.1.1 | Bandwidth limit | 63 |
| 6.2 | Image Format Control | 63 |
| 6.2.1 | Image ROI | 65 |
| 6.2.2 | Binning | 65 |
| 6.2.3 | Decimation | 66 |
| 6.2.4 | Readout direction | 67 |
| 6.2.5 | Test pattern | 68 |
| 6.3 | Acquisition Control | 69 |
| 6.3.1 | Trigger overlap | 71 |
| 6.3.2 | Dual Exposure | 73 |
| 6.3.3 | Trigger delay | 74 |
| 6.3.4 | <i>Timed vs TriggerWidth Exposure Mode</i> | 76 |
| 6.3.5 | Image Compression | 76 |
| 6.4 | Analog Control | 78 |
| 6.4.1 | Gain | 79 |
| 6.4.2 | White balance | 79 |
| 6.4.3 | Gamma correction | 81 |
| 6.4.4 | Black level | 82 |
| 6.5 | OE Auto Functions Control | 83 |
| 6.5.1 | OE Autoexposure/Autogain | 83 |
| 6.6 | LUT Control | 85 |
| 6.6.1 | LUT | 86 |
| 6.7 | Color transformation control | 87 |
| 6.7.1 | Color Correction Matrix (CCM) | 87 |
| 6.7.2 | How to perform a correct color calibration | 88 |
| 6.8 | Digital I/O Control | 94 |
| 6.8.1 | I/O stage | 95 |
| 6.9 | Counter and Timer Control | 96 |
| 6.10 | Encoder Control | 97 |
| 6.10.1 | Encoder interface | 98 |
| 6.10.2 | Encoder output mode | 98 |
| 6.10.3 | Encoder mode | 98 |
| 6.10.4 | EncoderValue wrapping management | 99 |
| 6.11 | Logic Block Control | 100 |
| 6.11.1 | Logic block module | 101 |
| 6.12 | Action Control | 102 |

| | | |
|--------|---|-----|
| 6.12.1 | Action Command | 102 |
| 6.12.2 | Scheduled Action Command | 103 |
| 6.13 | Event Control | 104 |
| 6.14 | User Set Control | 107 |
| 6.15 | Chunk Data Control | 108 |
| 6.16 | OE Serial Interface Control | 109 |
| 6.16.1 | Serial interface | 110 |
| 6.17 | OE Liquid Lens Control | 111 |
| 6.17.1 | Liquid Lens interface | 113 |
| 6.17.2 | Autofocus | 114 |
| 6.18 | OE Defective Pixel Correction Control | 115 |
| 6.18.1 | Defective Pixel Correction | 116 |
| 6.19 | Test Control | 117 |
| 6.20 | Transport Layer Control | 117 |
| 6.20.1 | Precision Time Protocol (PTP) | 120 |
| 6.21 | Sequencer Control | 121 |
| 6.21.1 | Sequencer overview | 121 |
| 6.21.2 | Configuration of a Sequencer set | 122 |

7 USE CASES 124

| | | |
|-------|--|-----|
| 7.1 | Wiring connection examples | 124 |
| 7.1.1 | Triggering the camera by an external device | 124 |
| 7.1.2 | Synchronizing an external device with Itala cameras | 125 |
| 7.2 | How to add a delay on the Output Lines of the camera | 127 |
| 7.3 | Streaming bandwidth management | 129 |
| 7.4 | Cognex Vision Pro compatibility | 133 |
| 7.5 | Sequencer Control configuration example | 140 |
| 7.5.1 | Working with Sequencer Paths | 140 |

8 TROUBLESHOOTING 142

| | | |
|-----|---|-----|
| 8.1 | The camera cannot be found in the available device list | 142 |
| 8.2 | Why some features are not present in the GenICam tree of the camera viewer? | 142 |
| 8.3 | Why does the camera give frame losses? | 142 |

1 GENERAL INFORMATION

1.1 Disclaimer

Always use and store Opto Engineering® products in the prescribed conditions in order to ensure they function properly. Failing to comply with the following conditions may shorten the product lifetime and/or result in malfunctioning, performance degradation or failure.

Be aware that incorrect functioning of this equipment may cause dangerous situations or significant financial losses. It is essential that the users ensure that the operation of the camera is suitable for their applications.

All trademarks mentioned herein belong to their respective owners.
Except where prohibited by law:

- All hardware, software and documentation are provided on an "as is" basis.
- Opto Engineering® accepts no liability for consequential loss, of any kind.

Upon receiving your Opto Engineering® product, visually examine it for any damage during shipping. If the product is damaged upon receipt, please notify Opto Engineering® immediately.

1.2 Intended use

This product is a **machine vision camera** for the detection, inspection and / or measurement of physical properties of items / objects. It provides a stream of images to a host device over a high-speed connection. Its operation can be synchronized with other devices using various types of electrical signals.

Please note that this product is not intended to be used as a CCTV camera in a video surveillance system.

1.3 Forbidden use

Please read the following notes before using this camera.
Contact your distributor or dealer for any doubts or further advice.

- Do not disassemble, modify or repair the product yourself. It may cause permanent malfunctioning or even fire or electric shock, possibly resulting in serious injury;
- Do not place the product in dusty, humid or hot places or near flames. These conditions may cause malfunctioning and damage or even fire or electric shock, possibly resulting in serious injury;
- Do not spray insecticide or apply other volatile chemicals on or around the product;

- This device must not be used in an application where its failure could cause a hazard to human health or damage to other equipment. Keep in mind that if the device is used in a manner not foreseen by the manufacturer, the protection provided by its circuits and by its enclosure may be impaired;
- This is a low voltage power supplied device. As such, the potential difference between any combination of applied signals must not exceed the supply voltage at any time;
- Higher voltages may cause a fault and can be dangerous to human health;
- This device has limited protection against transients caused by inductive loads. If necessary, use external protection devices like fast diodes or better still, specific transient protectors;
- Do not allow foreign objects to enter the unit or drop into holes, terminals and other openings or gaps. This may cause fire or electric shock, possibly resulting in serious injury;
- Disconnect the power cable before moving the product. Failure to comply with this precaution may damage the power cable, cause fire or electric shock, possibly resulting in serious injury;
- Do not scratch, cut, open or twist the power cables. It may cause malfunctioning, fire or electric shock, possibly resulting in serious injury;
- If the power cable is damaged or cracked, please contact our technical support and do not use the product. Damaged cables may cause malfunctioning, fire or electric shock, possibly resulting in serious injury;
- Do not insert or remove the plug of the power cable with wet hands. It may cause electric shock, possibly resulting in serious injury;
- Do not use the product in presence of inflammable gas. It may cause outbreaks and flames, possibly resulting in serious injury;
- If you notice any abnormality such as smell, smoke or overheating, turn off the power and disconnect the power cables. Continuing to use the product in these conditions may cause fire or electric shock, possibly resulting in serious injury;
- If you have dropped the product or damaged the product case, turn off the power and disconnect the power cables. Continuing to use the product in these conditions may cause fire or electric shock, possibly resulting in serious injury.

1.4 Revisions

In Table 1 are listed all the user manual revisions.

In the column *Description* are listed all the relevant differences between different revisions.

| Rev. | Date | Description | FW ver. |
|------|------------|--|---------------|
| 1.0 | 22/09/2021 | First manual release | 1.0.0 - 1.1.2 |
| 1.1 | 14/06/2022 | <ul style="list-style-type: none"> - Added references to new camera features - Added SDK installation section - Added Driver installation section - Added Itala View quick start section - Added Itala View wizards sections | 1.1.3 - 1.1.5 |
| 1.2 | 10/02/2023 | <ul style="list-style-type: none"> - Added CCM calibration procedure - Added TimerDelay feature - Added references to SDK documentation - Added PTP documentation - Updated storage and operating conditions - Updated sensor ADC resolution - Added references to GigE mechanical specifications - Added new part numbers in "Ordering code" section - Added use case in Section 7.2 | 1.2.0 - 1.2.2 |
| 1.3 | 07/03/2023 | <ul style="list-style-type: none"> - Added TriggerDelay feature - Added Action Commands feature - Added Serial interface configuration - Added Serial protocol (ASCII, Modbus, Binary) - Added PTP offset from UTC feature | 1.3.0 - 1.3.2 |
| 1.4 | 04/05/2023 | <ul style="list-style-type: none"> - Added DualExposure feature - Added new part numbers in "Ordering code" section - Added Variant in "Ordering code" section - Updated trigger input delay in "Electrical Specifications" | 1.4.0 |
| 1.5 | 30/05/2023 | <ul style="list-style-type: none"> - Added TestControl feature - Updated bandwidth limit formula - Updated access attribute of some features - Added Action Command section - Correction of Dual Exposure timings | 1.4.1 - 1.5.3 |
| 1.6 | 04/08/2023 | <ul style="list-style-type: none"> - Added "Streaming bandwidth management" use case - Added a caution note in the Liquid lens hardware installation section - Added Shock and vibration data | 1.4.1 - 1.5.3 |
| 1.7 | 23/08/2023 | - Added FCC Declaration section | 1.4.1 - 1.5.3 |

| | | | |
|------|------------|--|---------------|
| 1.8 | 27/09/2023 | <ul style="list-style-type: none"> - Added Cognex Vision Pro compatibility test procedure - Updated Encoder output mode section - Added Encoder mode section - Updated Dual Exposure timings | 1.5.3 - 2.0.0 |
| 1.9 | 07/11/2023 | <ul style="list-style-type: none"> - Updated the firmware update procedure - Added note to TriggerWidth Exposure feature | 2.0.0 - 2.0.2 |
| 1.10 | 20/11/2023 | <ul style="list-style-type: none"> - Added note to DualExposure feature - Added note to TriggerWidth Exposure feature | 2.0.0 - 2.0.2 |
| 1.11 | 16/01/2024 | <ul style="list-style-type: none"> - Correction of the monochrome test pattern sequence - Update of the sensors optical responses plots - Revision of all the feature names cited in the document - Added Sequencer Control section - Added Sequencer configuration use case | 2.1.0 - 2.1.2 |
| 1.12 | 30/05/2024 | <ul style="list-style-type: none"> - Added Dual Use classification info for IMX990 cameras - Added ChunkSequencerSetActive features - Added a note for Sequencer paths priority - Improved sequencer multi-path example - Added IMX249 support (ordering code, dual exposure, QE plots) - Correction of Dual Exposure timings - Correction of random vibration test PSD - Added sensor centering data - Updated Dual Exposure timings - Updated ordering code information - Added Section 6.3.1 - Added Section 6.10.4 | 2.2.0 - 2.2.2 |
| 1.13 | 31/07/2024 | <ul style="list-style-type: none"> - Added Autofocus section - Added Image compression section - Added Encoder and Counter chunk data - Added Polarized pixel formats | 2.3.0 |
| 1.14 | 20/12/2024 | <ul style="list-style-type: none"> - Added Device Pressure information - Added YUV411 pixel format - Added Linux driver installation section - Added Itala IP67 specifications - Added AutofocusDone event - Added oeLiquidLensAutofocusTriggerSource feature - Added oeFramesInBuffer feature - Added torque specification for mounting screws - Removed F-mount option - Added maximum current available on digital outputs - Added power supply requirements to satisfy 62368-1 | ≥ 2.4.0 |

Table 1: Manual revisions

1.5 Ordering code

The camera part number is composed as follow:

ITA000-WX-00Y-ZZ

The ordering code is explained in Table 2.

Refer to Opto Engineering website to check the availability of the desired part number.

NOTE: *ITA13-GM-10C-SWIR product is subject to statutory export control regulations and may require written information on intended end use and final destination.*
Dual Use classification: 6A003.B.4.A.

| Code | Description | Options | Value |
|------|-------------------|---------|--------------------------------------|
| ITA | Series Name | | |
| 000 | Sensor resolution | 04 | IMX287 - 0.40 Mpixels |
| | | 13 | IMX990 - 1.34 Mpixels |
| | | 16 | IMX273 - 1.58 Mpixels |
| | | 23 | IMX249 - 2.35 Mpixels |
| | | 24 | IMX392 - 2.35 Mpixels |
| | | 32 | IMX265 - 3.19 Mpixels |
| | | 50 | IMX264 - 5.07 Mpixels |
| | | 51 | IMX547 - 5.10 Mpixels |
| | | 81 | IMX546 - 8.13 Mpixels |
| | | 89 | IMX267 - 8.95 Mpixels |
| | | 120 | IMX304 - 12.37 Mpixels |
| | | | IMX253 - 12.37 Mpixels |
| | | 124 | IMX545 - 12.41 Mpixels |
| | | 162 | IMX542 - 16.19 Mpixels |
| | | 168 | IMX387 - 16.88 Mpixels |
| | | 196 | IMX367 - 19.66 Mpixels |
| | | 204 | IMX541 - 20.35 Mpixels |
| | | 246 | IMX540 - 24.55 Mpixels |
| | | 315 | IMX342 - 31.49 Mpixels |
| W | Interface | G | Ethernet |
| X | Mono/Color sensor | M | Monochrome |
| | | C | Color |
| 00 | Variant | 1X | Sony IMX Pregius™ 1st/2nd gen sensor |
| | | 2X | Sony IMX Pregius S™ 4th gen sensor |
| Y | Mount | C | C-mount |
| | | J | J-mount (M42x1 FD 12) |
| ZZ | Optional features | – | Standard version |
| | | EL | With liquid lens controller |
| | | PL | Polarized Polarsens™ sensor |
| | | SWIR | VIS-SWIR SenSWIR™ sensor |

Table 2: Ordering code

2 WARRANTY AND CERTIFICATIONS

2.1 Warranty

The device warranty is 5 years from the effective delivery date with reference to the device serial number.

Warranty covers the replacement or the repair of the defective part (components, device or part of it) with the exclusion of dismantling and shipping costs.

The replacement of one or more components does not renew the warranty period of the entire device.

The electronics and parts subjected to normal use or deterioration due to atmospheric agents and external environment are excluded from the warranty. Also, all failure caused by the lack of, insufficient or incorrect maintenance performed by unskilled or unauthorized personnel or due to unintended use or unauthorized replacements, alterations or repairs is excluded from the warranty.

The general validity of the warranty depends on:

- Maintenance being performed correctly as described in the device manual;
- The intended use of the device as specified in this manual.

2.2 CE Declaration

Itala cameras are conformal to the EMC directive 2014/30/EU and therefore comply with the following standards:

| Standard | Description |
|--------------|---|
| EN 61000-6-2 | Generic standards - Immunity standard for industrial environments |
| EN 61000-6-4 | Generic standards - Emission standard for industrial environments |

Table 3: EMC standards

2.3 FCC Declaration

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off

and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

2.4 KC Certification

Electrical and electronic equipment imported into South Korea is subject to KC certification. This is a mandatory certification system that ensures that only products certified by recognized bodies can enter the South Korean market. In particular, the certification checks the requirements for electromagnetic compatibility (EMC).

Itala cameras available for the South Korean market are identified by the KC mark and the KC registration number. For more information about the KC identification data, please scan the QR code on the product or its packaging.

2.5 Shock and vibrations

Itala cameras have been tested for shocks and vibrations according to the following standards and limits:

2.5.1 ITALA G - G.EL

| Standard | Date of issue | Test | Parameters |
|---------------|---------------|------------------|---|
| EN 60068-2-27 | 2009 | Shock | x/y/z axis, 20g, 11ms, 10 pos. / 10 neg. shocks |
| EN 60068-2-6 | 2008 | Sine vibration | x/y/z axis, 10g, 50-500 Hz, 10 sweep |
| EN 60068-2-64 | 2008+A1:2019 | Random vibration | x/y/z axis, 5g RMS, 0.056g ² /Hz PSD, 30 min for each axis |

Table 4: Shock and vibrations standards and limits

2.5.2 ITALA G.IP

| Standard | Date of issue | Test | Parameters |
|---------------|---------------|------------------|---|
| EN 60068-2-27 | 2009 | Shock | x/y/z axis, 50g, 11ms, 10 pos. / 10 neg. shocks |
| EN 60068-2-6 | 2008 | Sine vibration | x/y/z axis, 10g, 50-500 Hz, 10 sweep |
| EN 60068-2-64 | 2008+A1:2019 | Random vibration | x/y/z axis, 5g RMS, 0.056g ² /Hz PSD, 30 min for each axis |

2.6 RoHS, REACH and WEEE

Itala cameras are conformal to the following directives and standards:

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

3 INTRODUCTION

3.1 Manual and conventions

Opto Engineering® SpA, with its registered office in Strada Circonvallazione Sud 15, 46100 Mantova (Mn) - Italy, hereinafter the manufacturer, provides all the necessary information in this installation, use and maintenance manual in a clear and simple way to install, use and service the product Itala.

The recipients of this manual are all those who have the knowledge, experience and capability of understanding the standards, prescriptions and safety measures indicated in this manual. Such people will be later identified as qualified personnel who are authorized to transport, install, use and service the products described in this manual.

This material can only be used by the customer whom this manual has been delivered to, in order to install, use and service the product.

The manufacturer will retain the right to modify or improve the manual and/or the product referred to in this manual without any prior notice.

The following typographical conventions are used in this document:

NOTE: notes contain important information. Highlighted outside the text to whom they refer



CAUTION: these indications highlight procedures that, if not observed in their entirety or in part, can cause damage to the machine or to the appliances



HAZARD: these indications highlight procedures that, if not observed in their entirety or in part, can cause injuries or affect the health of the operator

3.2 Product identification data

Itala cameras are identified with a label placed on the lateral side of the packaging box. A compact version of the label is also placed on the back side of the camera, near the connectors.

This label is used to identify the part number, serial number and MAC address of each device. A label example is shown in Figure 1.

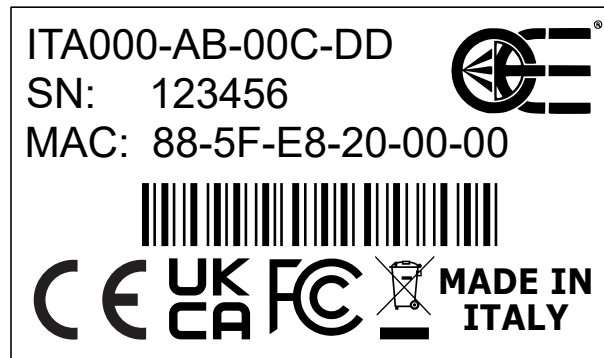


Figure 1: Example of camera label.

3.3 Storage and use conditions

3.3.1 Storage conditions

Storage environment between -10°C and 60°C.

Avoid thermal shock by not exposing the product to sudden changes in temperature.

Store the product in a dry place: storage environment with relative humidity (RH) less than 80% (no condensation).

3.3.2 Operating conditions

Extreme temperatures affect the product functionality, especially the electronic components.

Avoid thermal shock by not exposing the product to sudden changes in temperature.

Since the product includes electronic components, it may generate heat when functioning: it's very important to dissipate an appropriate amount of heat (if necessary, operate the device with a forced air cooling system).

Use the product in a dry place: operating environment with relative humidity (RH) less than 80% (no condensation).

In general, avoid to store and use the camera in the following environments:

- Environments with strong electric/magnetic fields.
- Places exposed to direct sunlight, rain or snow.
- Environments exposed to particular gas and dangerous substances.
- In extremely vibrating systems.
- Dusty places.
- Extremely humid places.
- Excessive hot/cold environments.

Itala cameras have been tested in climatic chamber in order to prove the temperature capabilities.

Case temperature must not exceed the range $-25^{\circ}\text{C} \div 65^{\circ}\text{C}$.

NOTE: The case temperature has been measured on the external part of the aluminum case, near the mount of the camera.



CAUTION: Ambient and case temperatures can highly differ in case of poor heat dissipation: the user must monitor Itala temperature in order to avoid damage to the device.

Opto Engineering will not be responsible in case of overheating faults.



CAUTION: Image sensor has an operating temperature ranging from -30°C to $+75^{\circ}\text{C}$. Please monitor the **Device Temperature** feature in order not to exceed the allowed range.



CAUTION: Be aware that image sensor quality easily degrades when its temperature exceeds the range $-10^{\circ}\text{C} \div +60^{\circ}\text{C}$.

3.4 Cleaning and maintenance

Even if the camera is equipped with a rugged mechanical case, some measures must be followed in order not to damage the camera itself.

In particular, when cleaning the Itala camera, remember:

- To avoid disassembling the camera.
- To avoid liquids or inappropriate cleaning chemicals like benzene, alcohol, spray-like cleaners.
- To use an appropriate soft cloth or soft brush.

The protection glass or filter placed in front of the sensor can be cleaned using compressed air or a cotton swab soaked with isopropyl alcohol.

4 GETTING STARTED

4.1 Overview

Itala is an industrial **Gigabit Ethernet camera** compliant with the *GigE Vision* and *GenICam* specifications. This camera is capable of transferring image data at high frame rates and over long distances, up to hundreds of meters.

Specifically designed for harsh industrial environments, Itala cameras guarantee reliable operation and top notch performances in their class. The *GigE Vision* and *GenICam* compliance allow easy camera integration and replacement. With flexible powering options (12-24 Vdc and **Power over Ethernet**), Itala cameras are compatible with most vision systems, allowing simple and flexible wiring configurations.

4.2 Hardware installation

4.2.1 Camera installation

The camera is provided with 4 x M3 threaded holes on each side, allowing for flexible and robust mounting. It is recommended to mount the camera to a metal object using a metal bracket in order to facilitate the heat dissipation. Before installing the camera make sure to align it correctly, as requested by your application. Keep in mind that you can also exploit **ReverseX** and **ReverseY** camera features to flip the image on X and Y axis directly in camera, without performance loss. Room should be provided to ensure a good cables setting on the back of the camera.

NOTE: tighten the screws to a torque of 1.2-1.4 Nm. Use a thread-locking fluid if the device is subject to high levels of vibrations.

4.2.2 Lens

ITALA G - G.EL

Cameras which come in **TYPE 1** enclosure are equipped with a standard **C-mount** (1 inch diameter, 32 threads per inch), with a flange distance of 17.526 mm.

Cameras which come in **TYPE 2** enclosure are equipped with an **M42x1** threaded mount, with a flange distance of 12 mm.

See section 5.5 for the cameras dimensional drawings.

Before installing the lens, make sure that the lens and the camera protection glass are perfectly clean. Refer to section 3.4 for cleaning instructions.

NOTE: for heavy lenses, consider to directly mount the lens with an appropriate clamping system instead of relying on the camera mounting holes. If the lens allows you to adjust the phase of the mount, this operation is straightforward. Otherwise, you need to ensure that the camera orientation will be correct after screwing it in final position.

ITALA G.IP

Cameras which come in **IP67** enclosure are equipped with a standard **C-mount** (1 inch diameter, 32 threads per inch), with a flange distance of 17.526 mm.

See section 5.5 for the cameras dimensional drawings.

Before installing the lens, make sure that the lens and the camera protection glass are perfectly clean. Refer to section 3.4 for cleaning instructions.



CAUTION: a proper lens enclosure is required to achieve full IP67 protection.

IP67 Lens Enclosure

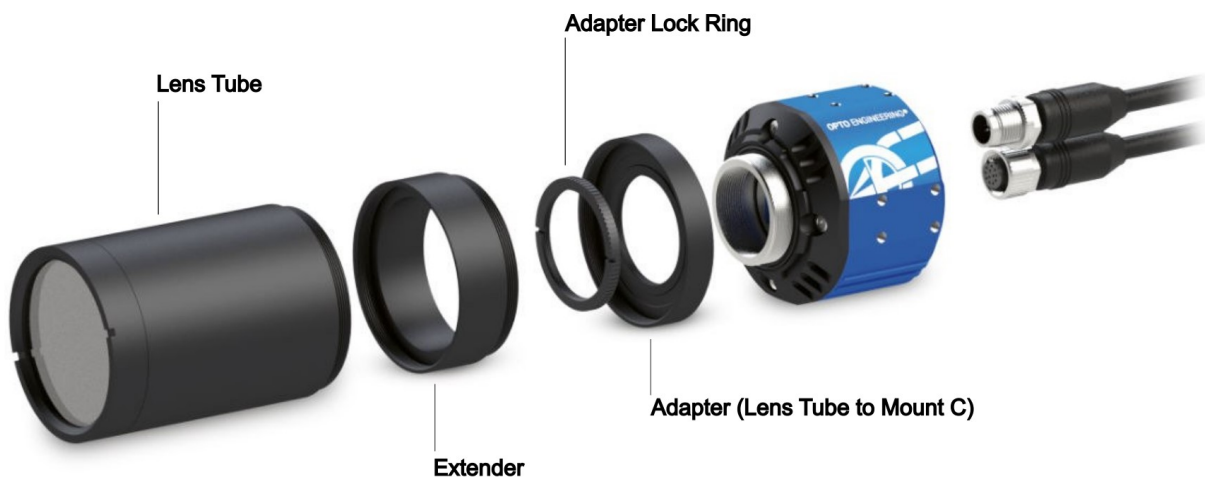


Figure 2: IP67 lens enclosure installation.

The **Adapter** should be properly tightened onto the C-mount to achieve a good seal. The **Adapter Lock Ring** is designed to lock the **Adapter** in place and prevent it from moving. The **Extender** is optional and its installation depends on the lens size. The **Lens Tube** should also be tightened to ensure a good seal.



CAUTION: do not overtighten the lens enclosure parts as this may cause damage to the camera and lens enclosure parts.

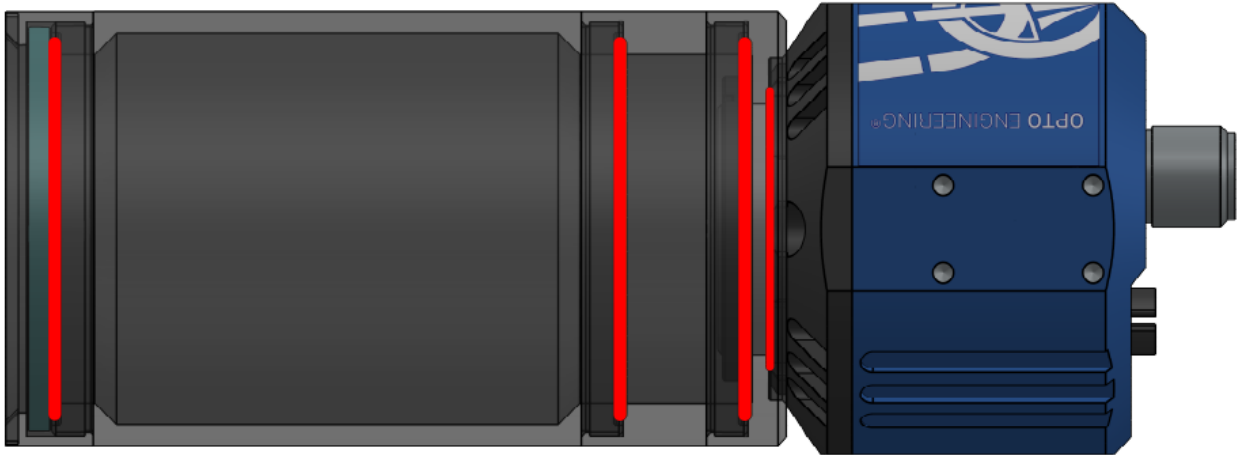


Figure 3: O-Ring placement for IP67 lens enclosure.



HAZARD: make sure that all the O-rings required by the mechanical parts of the lens enclosure are in place to ensure a good seal. Failing to do so may result in damage to the camera.

4.2.3 Ethernet cable

Connect the camera to the host device with a suitable Cat 5e Ethernet cable or better, compliant with ANSI/TIA-568 standard, with a maximum length of 100m. A shielded cable should be used to improve the system EMI immunity, especially in harsh industrial environments.

ITALA G - G.EL

Itala G - G.EL cameras come with 2 x M2 threaded holes for use with screw lock RJ45 connectors. For moving applications (e.g. camera mounted on a robotic arm) use screw lock connectors and cable strain reliefs to ensure a reliable connection. A high-flexibility cable specifically designed for a high number of bending cycles is also recommended. See section 5.5 for the cameras dimensional drawings.

ITALA G.IP

Itala G.IP cameras are equipped with an M12 X-Coded female IP67 Ethernet connector that must be screwed tightly for proper sealing. Additional strain relief is not required for moving applications.

A high-flexibility cable specifically designed for a high number of bending cycles is recommended. See section 5.5 for the cameras dimensional drawings.

If you intend to power the camera with PoE (Power over Ethernet), connect the cable to a suitable PoE injector or NIC (network interface card).



CAUTION: *always use certified IEEE 802.3af PoE power supplies, injectors and NICs. Failing to do so may result in damaging the camera.*



HAZARD: *cables should be arranged carefully, avoiding pinching, sharp corners and excessive tension. Failing to do so can lead to short circuits, damage to the appliances or even fire.*

4.2.4 GPIO cable

The camera can also be powered through the GPIO (General-Purpose Input/Output) port using a GPIO cable up to 30m in length and a suitable power supply. To ensure optimal performance and EMI immunity in particularly disturbed environments, use shielded cables. For more information on the GPIO connector pinout and a complete list of Itala cameras' electrical specifications, see sections 5.6 and 5.2, respectively.

ITALA G - G.EL

Plug the push-pull connector in firmly, paying attention to the correct orientation.



CAUTION: *don't force the connector if you encounter too much resistance. Check the connector orientation and try again.*

ITALA G.IP

Itala G.IP cameras are equipped with an M12 A-coded male IP67 GPIO connector, which must be screwed tightly for proper sealing. Additional strain relief is not required for moving applications.

See section 5.5 for the cameras dimensional drawings.

For moving applications (e.g. camera mounted on a robotic arm) a high-flexibility cable specifically designed for a high number of bending cycles is recommended.



HAZARD: maximum power to be used to supply product PS2 per 62368-1 and LPS per Annex Q.1 less than 100W.



HAZARD: always use suitable cables and power supplies that satisfy all the device specifications. Failing to do so may result in damaging the camera, fire or injury to the operator.



HAZARD: cables should be arranged carefully, avoiding pinching, sharp corners and excessive tension. Failing to do so can lead to short circuits, damage to the appliances or even fire.

4.2.5 Liquid lens

If the camera comes with the **Liquid Lens Controller** option, use a specific cable to connect the camera GPIO port to an Opto Engineering® product integrating an electrically tunable lens. There are two possible configurations:

- A dedicated **point-to-point cable** allows for a simple connection of the liquid lens when the camera is powered via PoE.
- A dedicated **Y-cable** allows for simultaneous connection of liquid lens, power supply and synchronization devices.

See section 6.17.1 for more information about the liquid lens connection and operation.



CAUTION: when using the **Y-cable** be sure to **first connect the camera**, then the lens and finally the power supply. Never connect/disconnect the camera connector while the lens and/or the power supply are connected. Failing to do so may result in damaging the liquid lens integrated EEPROM.



CAUTION: always use the specific cable provided by Opto Engineering® to connect the liquid lens to the camera. Failing to do so may result in damaging the camera or the liquid lens.

NOTE: *if the lens allows you to adjust the phase of the mount, choose an orientation that will reduce the strain on the cable.*

The orientation of the lens mounting can affect the image quality. Read the lens specifications before the final installation of the vision system.

4.3 System configuration

NOTE: *the camera firmware is updated frequently to add new features and improve the existing ones. Regularly check on Opto Engineering website for the availability of a new firmware version and **update the camera before installation.***

4.3.1 System requirements

The Itala SDK can be installed in a system working with one of the following OS:

- Microsoft Windows 10 / 11 (64-bit)
- Ubuntu 18.04 / 20.04 / 22.04 (64-bit)

Proper functioning of the camera has not been tested with other operating systems. Contact your Opto Engineering representative for further information.

Be careful to use a NIC (network interface card) which supports Gigabit Ethernet communication: in particular, choose a NIC with jumbo frame packets capability.

The Itala camera is a high performance device which streams uncompressed images with high data throughput. In order to guarantee optimal performances, the host system should be sufficiently powerful to handle the large amount of data sent by the camera. Consider choosing a high performance CPU and enough amount of RAM for the image acquisition and processing of your specific application.

4.3.2 Camera Driver

To better handle the image streaming high throughput, it's recommended to use the **Itala filter driver**. The filter driver intercepts *GigE Vision* streaming protocol packets, reassembles the whole payload and sends it directly to the application image buffer. This allows to skip the standard network protocol stack that would increase latency and CPU usage on the host machine (Fig.4). The result is a low level packet handling offload which optimizes the host system resources consumption.

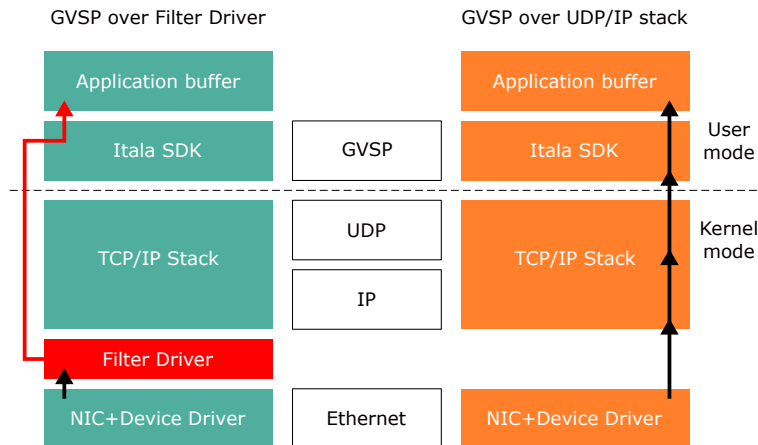


Figure 4: GigEVision Streaming (GVSP) with and without the Filter Driver

4.3.3 Install Driver on Windows

The Itala SDK installer takes care of the necessary filter drivers which are automatically installed on the host computer.

You can check for successful installation of the filter driver on your Ethernet connection property window. Go to *Control Panel > Network and Sharing Center > Change adapter settings*, right-click on your Ethernet connection and select *Properties*. In the *Networking* tab you should see the filter driver entry with a selected checkbox, as shown in Fig.5.

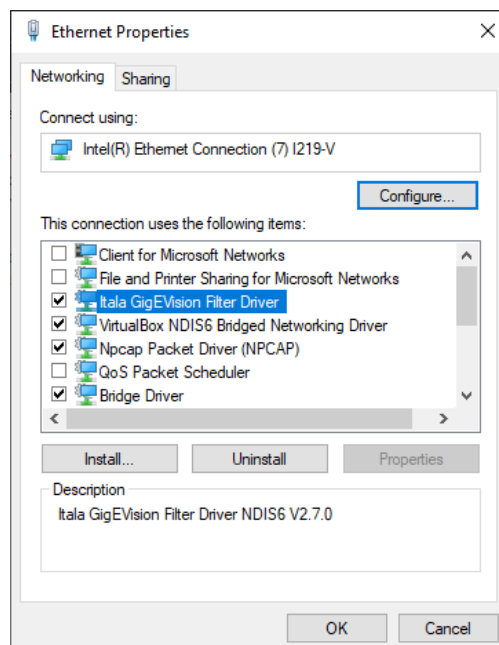


Figure 5: Successfully installed filter driver.

NOTE: in order to avoid conflicts, it's recommended to disable filter drivers from other camera vendors that you may have installed on your system.

If you need to re-install the drivers after an unwanted removal, follow this procedure:

1. Go to the Itala SDK installation directory.
2. Open the Filterdriver folder.
3. Launch *install_driver_win10.bat* in case of Windows 10 OS. Please be careful to choose the correct batch depending on the operating system.
4. At the end of the installation, the filter driver will appear in the NIC property window (Fig.4).

4.3.4 Install Driver on Ubuntu

Ubuntu GEV module is not mandatory for 1G cameras but is strongly suggested for higher speed cameras, specially if you are experiencing package incomplete or package loss problems. There are two ways to install the GEV driver in ubuntu (*oegevmodule*).

The first one is executing the script */opt/itala-sdk/scripts/oegevmodule/install_oegevmodule.sh* with administrator privilege. It will try to download and install the correct module version in your system.

The second way is to download and install GEV module manually from the Opto Engineering website (<https://www.opto-e.com/en/resources/download-itala-drivers/>). To choose the correct version you need to know:

- the driver version supported by your Itala SDK version: check the file */opt/itala-sdk/changelog.md*
- your kernel version: open a terminal and type `'uname -r'`.

The file you are looking for follow the format *oegevmodule-<driver-version>-<kernel-version>.tar.gz*, e.g. *oegevmodule-24.04.0-5-15-119-generic.tar.gz*. Once the package is downloaded, you can unzip the archive and follow the *README* instructions in the unzipped folder.

Each time the kernel is upgraded, i.e. for a system upgrade, you need to install again the driver.

Each time you upgrade the Itala SDK you may need to upgrade the GEV module too. Check */opt/itala-sdk/changelog.md* or the official Opto Engineering website to verify if the GEV module version installed is supported by your Itala SDK version.

You can check if the installation complete correctly by executing `'lsmod | grep oegevmodule'` from a terminal. If the command return nothing, *oegevmodule* is not installed correctly otherwise it is. You can also evaluate the *oegevmodule* activity in the kernel logs (by executing `'dmesg'` linux utility).

You can always refer to our Tech Support if you are issuing installation problem or your target kernel version is not supported yet.

4.3.5 Network and configuration

The camera is factory configured to automatically obtain an IP address in DHCP / LLA mode. This ensures the highest compatibility with different network configurations. For the first connection, it's recommended to configure your network settings in order to use DHCP.

Go to *Control Panel > Network and Sharing Center > Change adapter settings*, right-click on your Ethernet connection and select *Properties*. In the *Networking* tab select *Internet Protocol Version 4 (TCP/IPv4)* from the list and then click *Properties*.

Select *Obtain an IP address automatically* and click *OK*. Finally, click *OK* on the previous window.

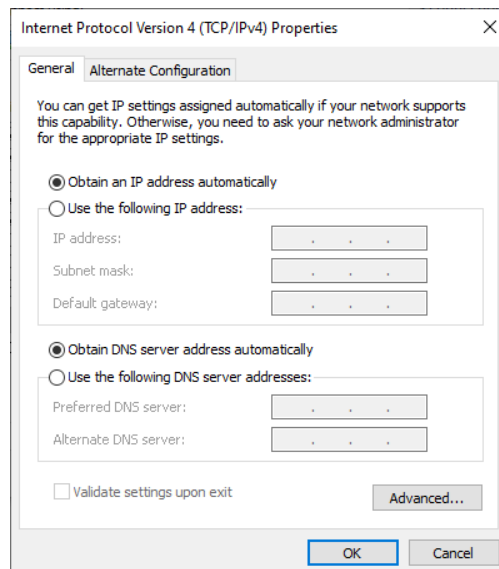


Figure 6: DHCP configuration of the network connection.

If the camera is not accessible you can force it to adopt an **IP configuration** which is compatible with the current NIC settings. To do so, refer to section 4.7.2.

After the first connection, it is recommended to set a static IP address for both NIC and device whenever possible. This ensures a faster discovery process and IP negotiation.

It is recommended that the connection is as simple as possible. To achieve best performance use direct connection with the NIC or connect the camera and the host computer to the same Ethernet switch (without any other heavy traffic routed through the same switch).

4.3.6 Bandwidth management

To achieve the best streaming performances, connection reliability and to reduce CPU consumption, it's recommended to configure the NIC (network interface card) to use **Jumbo** frames. Jumbo

frames are Ethernet frames which are larger than 1500 bytes and allow to increase the connection efficiency, reducing the amount of protocol overhead. Opto Engineering® recommends to use a NIC which supports Jumbo frames of at least 9000 bytes.

Jumbo frames are usually turned off by default. To enable them, go to *Control Panel > Network and Sharing Center > Change adapter settings*, right-click on your Ethernet connection and select *Properties*.

In the *Networking* tab click on *Configure*. The NIC settings window will appear.

In the *Advanced* tab locate the *Jumbo frame* or similar entry and enable it (Fig.7). The value to set may differ depending on the specific NIC model and manufacturer.

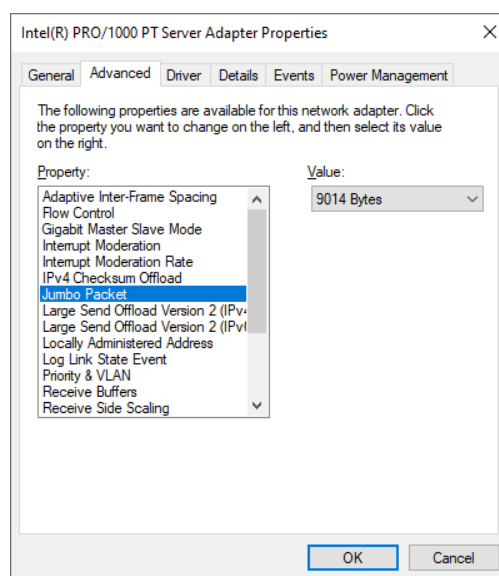


Figure 7: NIC advanced settings with Jumbo frames enabled.

If you still experience issues with the camera connection you can try:

- Installing the latest NIC drivers.
- Increase the *receive buffer size* of your NIC.

When connecting **multiple cameras** to a single computer, it is recommended to connect all devices directly using multiple gigabit NICs. If you're connecting the camera through an Ethernet switch, make sure it also supports jumbo frames. Keep in mind that if multiple devices are connected to the same Ethernet switch, they will share the available bandwidth.

For more information about bandwidth management and multi-camera system configuration, refer to section 6.1.1.

4.4 Itala SDK

Itala cameras comes with a complete Software Development Kit, Itala SDK, which takes full advantage of the latest standards and technologies in machine vision industry.

The SDK includes:

- Itala API
- Itala View
- GenTL producer (.cti)
- Filter driver
- Documentation with code examples

In order to install the Itala SDK correctly, do the following steps:

1. Download the Itala SDK from Opto Engineering website and run the installer.
2. The Itala SDK setup window is displayed: make sure to follow the instructions listed (Fig.8).

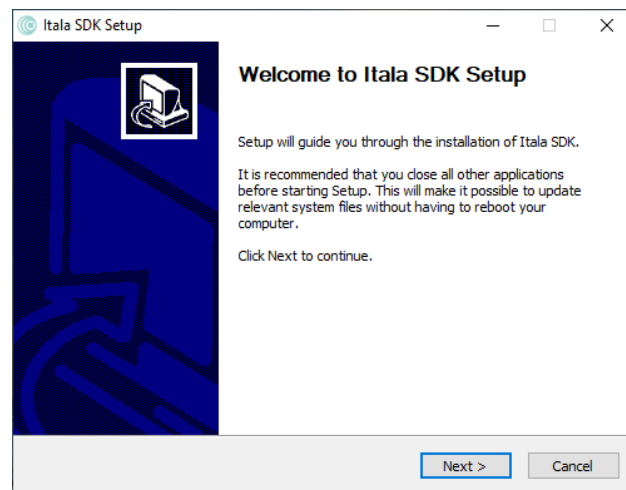


Figure 8: Itala SDK setup window.

3. Check the licence terms before installing Itala SDK (Fig.9).
4. Choose the destination folder (Fig.10).
5. Select the components which need to be installed (Fig.11). In case of installation of the filter driver only, step 6 can be skipped.
6. (Optional) In case the .NET runtime needs to be installed, click *Install* in the .NET runtime installation window (Fig.12). In case of successful installation, the window shown in Fig.13 will appear.
7. The Itala SDK installation will be performed automatically. The progress bar can be monitored to check the installation status. At the end of the installation, a confirmation window will be displayed (Fig.14).

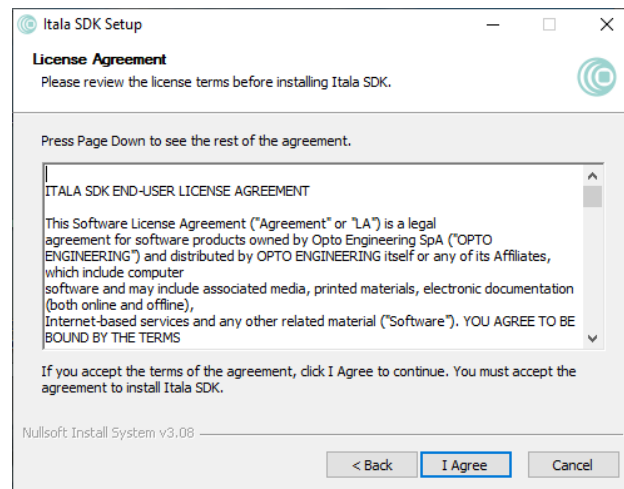


Figure 9: Itala SDK licence agreement window.

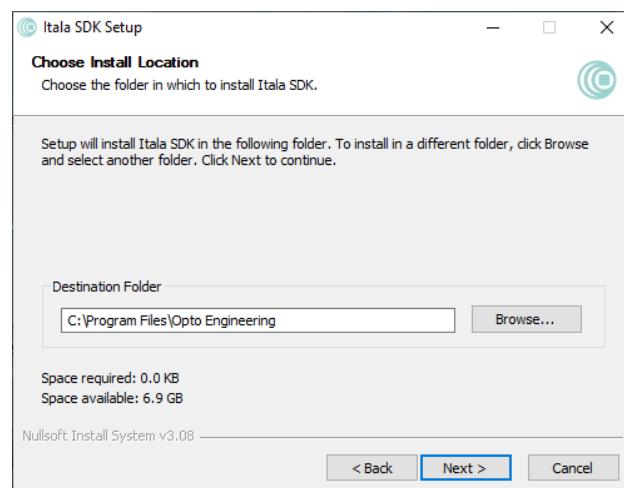


Figure 10: Itala SDK destination folder window.

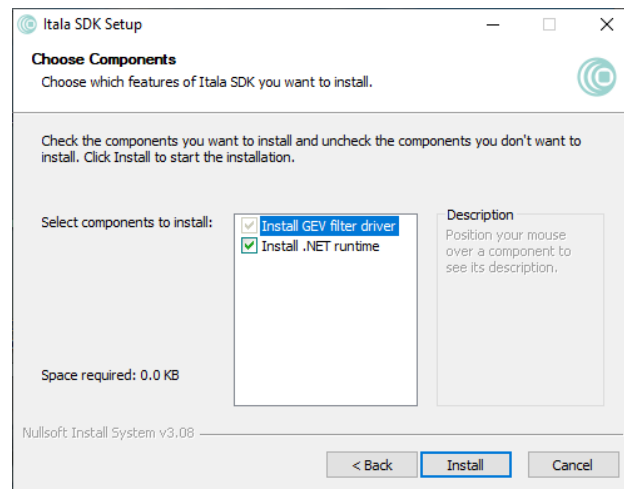


Figure 11: Components selection window.

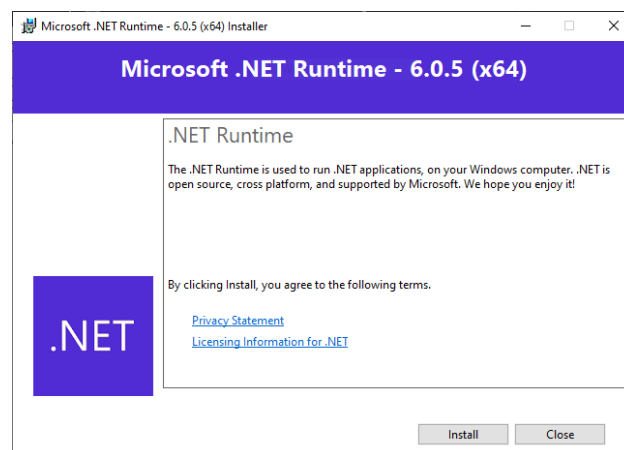


Figure 12: .NET runtime installation window.

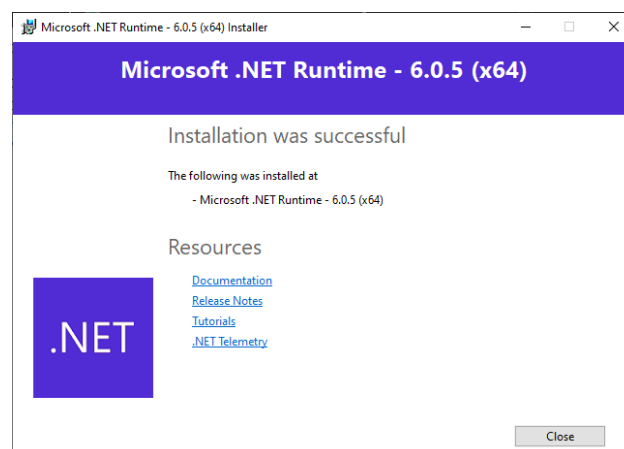


Figure 13: .NET runtime successful installation window.

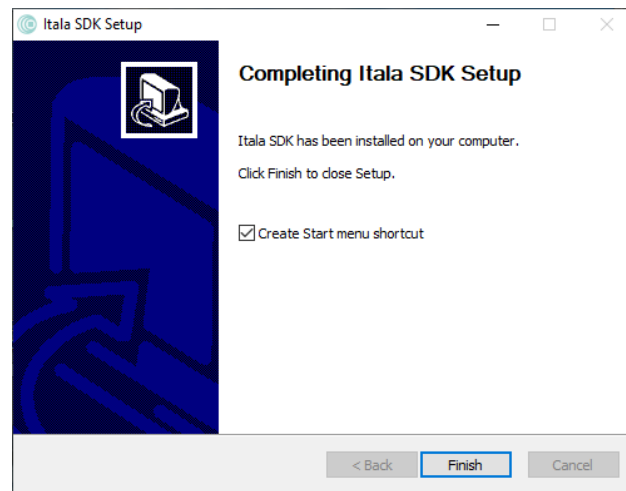


Figure 14: Itala SDK installation finished successfully.

4.5 Using the camera with Itala API

With Itala API it's easy to integrate Itala devices in custom applications, thanks to an extensive set of examples and complete documentation. For more information about the use of the library, refer to Itala API documentation in the SDK installation folder.

4.5.1 Itala SDK documentation

SDK documentation can be found in the installation directory (*Itala SDK > Development > doc > html*).

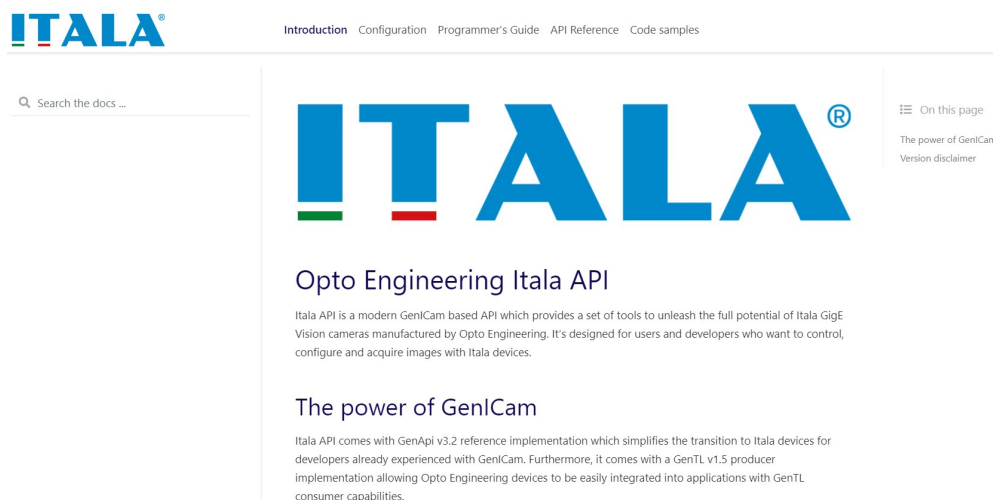


Figure 15: Home page of Itala SDK documentation.

Figure 15 depicts the home page of Itala SDK documentation.

4.6 Using the camera with third party software

Itala cameras are compliant to **GigE Vision** and **GenICam** standards, allowing easy integration with third party vision software. In addition, the SDK includes a **GenTL producer** (.cti file) compliant with the GenTL specifications hosted by EMVA. This further enhances the interoperability with other compliant devices and software.

4.7 Using the camera with Itala View

Itala View is a GUI tool which allows the evaluation, configuration and troubleshooting of Itala cameras. With a comprehensive set of utilities and wizards, Itala View speeds up the evaluation and deployment of a vision system built around Itala cameras.

To facilitate the first use of the application, a brief overview is given in the following sections.

4.7.1 Tabs and panels

With reference to Fig.16, the main window of Itala View can be divided in different functional areas:

1. Menu bar
2. Device discovery
3. Device information and control
4. Video streaming
5. Image data analysis and logging
6. GenICam feature tree

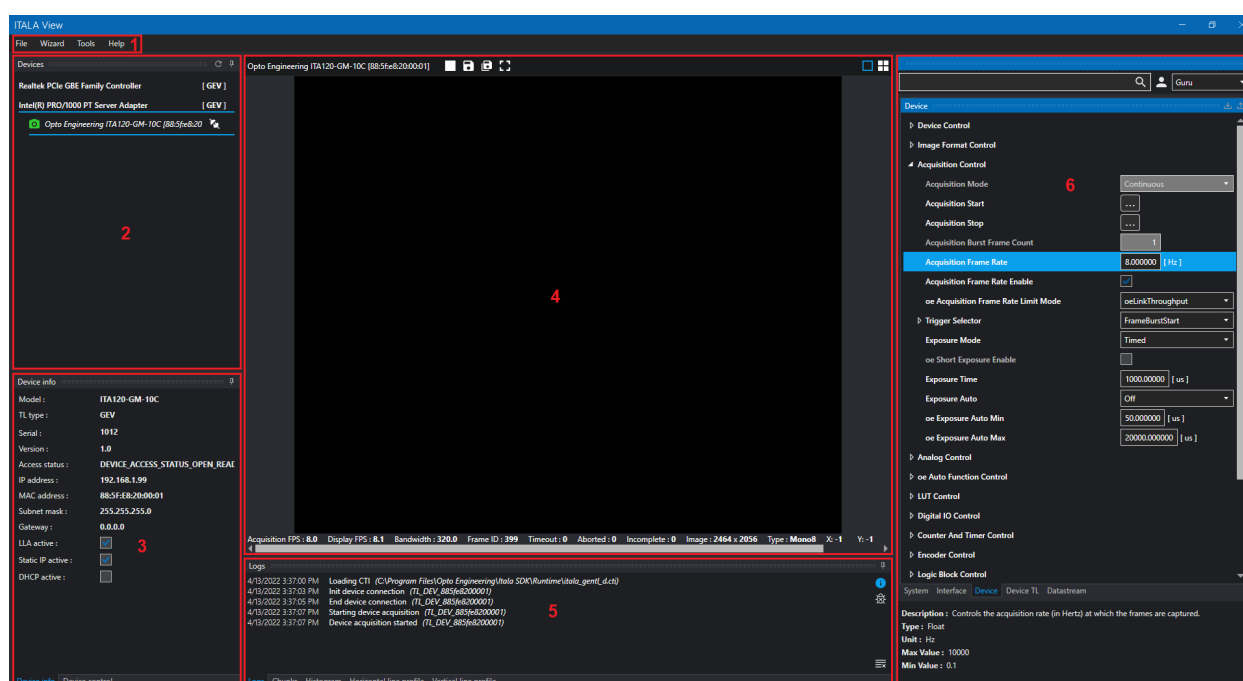


Figure 16: Viewer main window.

The **menu bar** gives you access to the settings, tools and wizards of the application.

The **device discovery** panel lists the NICs of your computer and the cameras connected to them. A refresh button on the top allows you to perform a discovery and enumeration of the GigEVision devices connected to your network.

Each device has a status icon which signals if it's reachable or not. An unreachable device can be symptom of a wrong IP configuration or that the same device is currently in use by another client application.

Next to each device there's a connect/disconnect button used to gain access to it.

In the **device info** tab is shown some essential information about the device currently selected in the discovery panel. This includes the device model, serial number, MAC address and current IP address.

In the **device control** tab you can select the desired access mode and the number of buffers you want to allocate for image grabbing.

The **video streaming** view allows you to control the acquisition process and see the actual images grabbed by the camera.

On the top of the panel there are some quick access buttons to start/stop the acquisition, save images and toggle the full-screen mode.

On the bottom there's a status bar with statistics about the acquisition and useful information about the current image.

Using the **image data analysis and logging** tabs you can see the application log, current image chunk data and perform different types of analysis on the acquired image.

Through the **GenICam feature tree** you can access the camera parameters. The features are grouped by functions in a hierarchical manner and allow to configure the camera peripherals and/or read their status. These include basic functions, like the exposure time, gain or trigger settings and more advanced ones, like the encoder or the liquid lens controller.

You can use the tabs below the tree view to switch between the node maps of both GenTL modules and the connected device (selected by default).

4.7.2 IP configurator

From the *Tools* menu you can access the *IP Configurator* utility. The IP configurator has been designed to efficiently address network configuration issues of Itala cameras, including but not restricted to:

- Camera and NIC set with persistent IPs but different subnets
- Camera and NIC set with persistent IPs but different subnet masks
- Camera set in DHCP mode and NIC set with a persistent IP
- Camera set with a persistent IP and NIC set in DHCP mode

As can be seen in Fig.17, the IP configurator presents a panel for device discovery and enumeration similar to the one of the viewer main window. On the right relevant NIC and camera information related to the currently selected device in the aforementioned discovery panel can be seen.

IP configuration issues can be solved from the **settings** panel in the lower right corner. For example camera can be forced to adopt a persistent IP coherent with the current NIC IP settings. Just input the correct data and click the *Apply* button. In the log panel the configuration progress and check that the settings has been correctly applied can be seen.

As in the viewer main window, the icon next to each enumerated device show potential issues with a red warning sign.

For more information about the IP configuration of the camera refer to section 4.3.5.

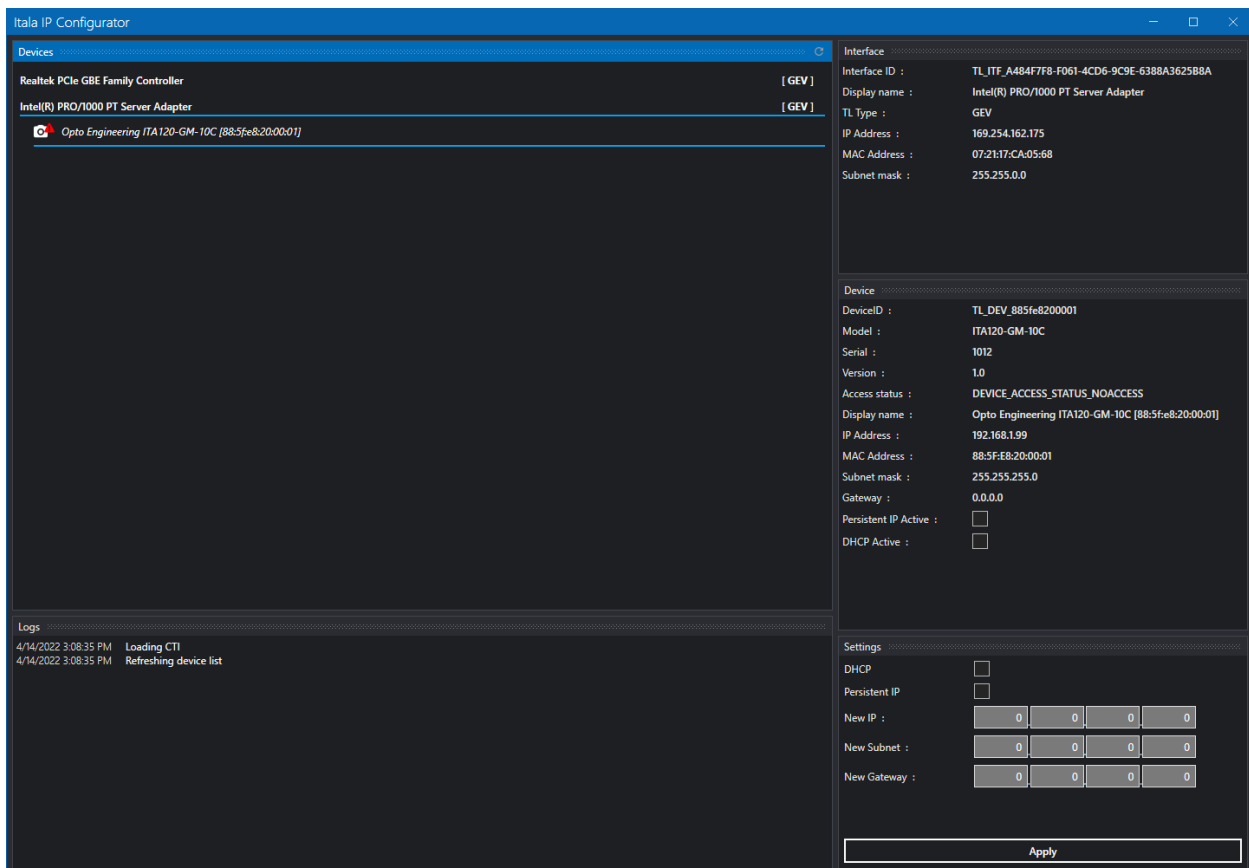


Figure 17: IP configurator window.

4.7.3 Firmware update

From the *Tools* menu you can access to the *Firmware Update* utility (Fig.18). You can now select a firmware file for each device that has been enumerated. The latest firmware for Itala cameras can be downloaded from Opto Engineering website.

Follow this steps to update one or more devices:

1. Select a firmware update file for each device you want to update.
2. Click on the *Update* button to start a batch update process.
3. **Do not disconnect or power down** the devices during the update process (Fig.19). The LED blink signals that the camera is writing to the flash memory.
4. Wait until all the devices are correctly updated.

NOTE: *do not disconnect or power down* the device during the update process. Failing to do so could lead to a condition where the device is no longer bootable and it should be returned to Opto Engineering for a factory reset.

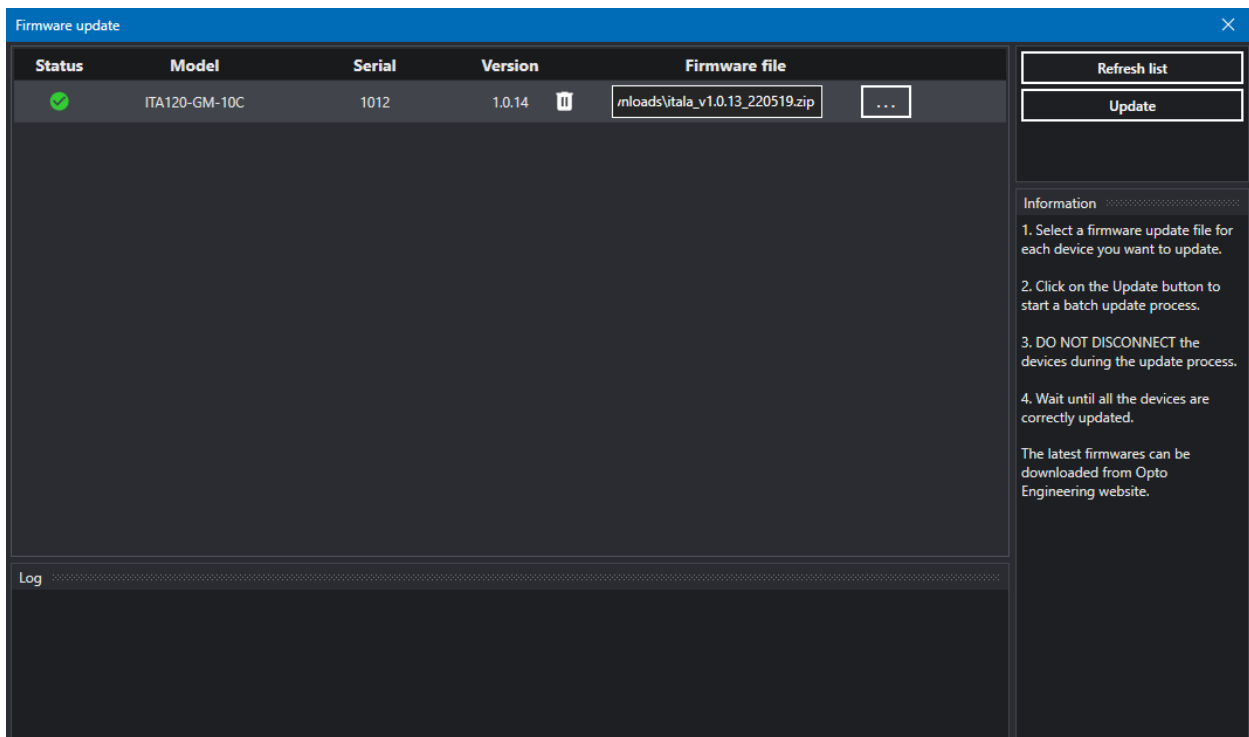


Figure 18: FW updater window.

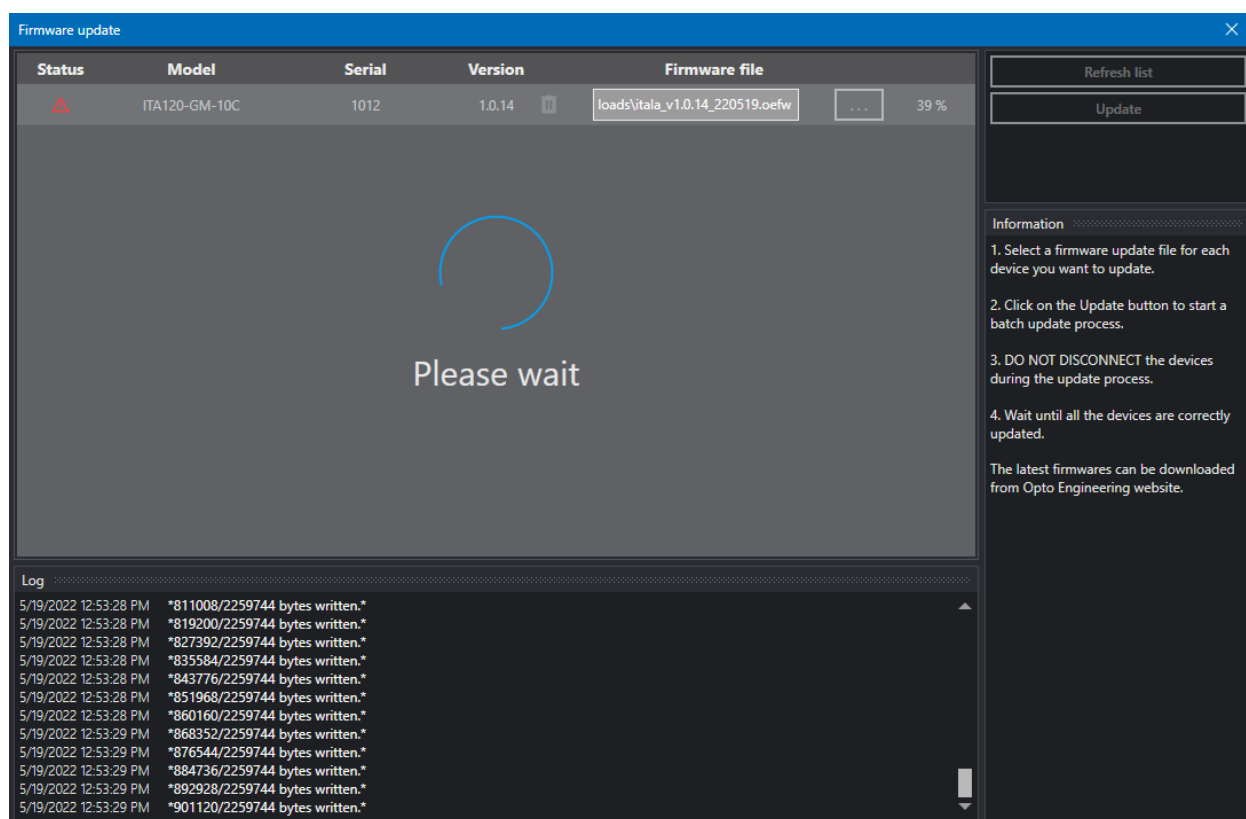


Figure 19: FW update in progress.

4.7.4 LUT wizard

From the *Wizard* menu you can access to the *LUT* wizard. This allows to view and edit the LUT of the selected camera.

Click on *Import from camera* to read the LUT from the camera registers and display it in the *Chart* tab (Fig.20). You can now edit each value the LUT in the *Table* tab (Fig.21). A better way to set a specific LUT is to load a CSV file previously generated with a spreadsheet editor or similar software. You can also write the current LUT on a CSV file, edit it and then read it back.

When you're satisfied with the resulting LUT, click on *Apply* to save it in the camera memory. For more information about the LUT feature refer to section 6.6.1.

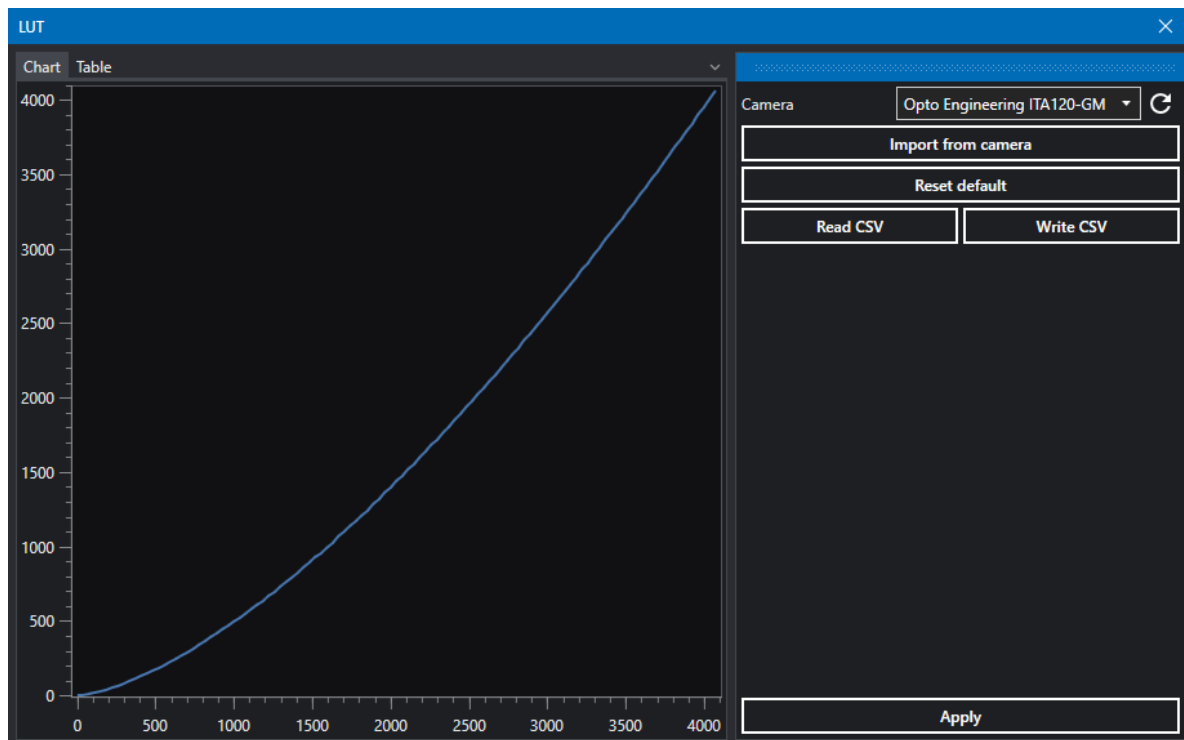


Figure 20: LUT wizard.

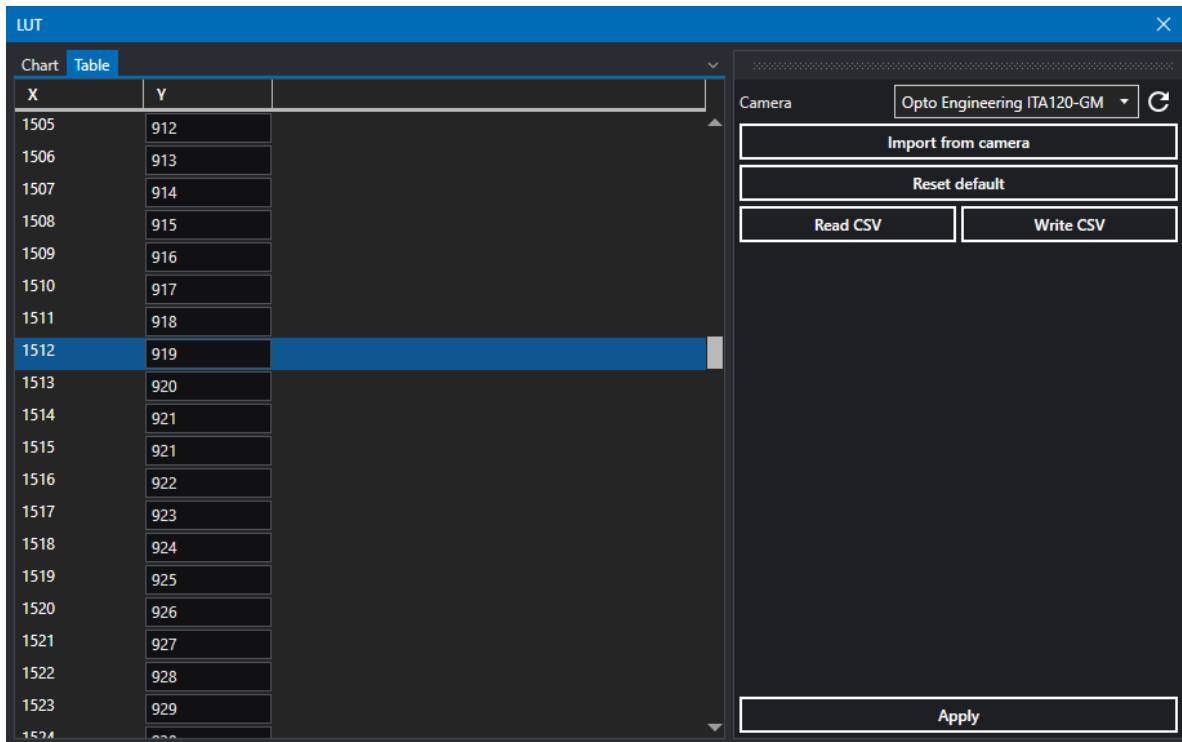


Figure 21: LUT wizard.

4.7.5 Defective pixels correction wizard

The image sensor defective pixels are mapped during Itala cameras production and testing procedures, since most of them are related to the sensor silicon production process. Their values are then corrected real-time in the camera acquisition pipeline. For more information about the defective pixel correction refer to section 6.18.

Anyway, there are other environmental factors which can increase the amount of defective pixels during the camera life. For this reason, Itala cameras enable the user to perform a custom pixel correction that takes into account these defects.

In order to enable the user defined defective pixel correction, the result of this operation must be saved in one of the available user set. The default user set, in fact, will correct only the defective pixels detected in factory.

From the *Wizard* menu you can access the *Defective Pixel Correction* wizard (Fig.22).

1. Put the camera in freerun acquisition or make sure there is a continuous stream of incoming trigger pulses.
2. For best results, consider using a raw pixel format with a bit depth of 12 bits, for example *Mono12p* or *BayerRG12*.
3. Start the image acquisition.
4. Cover the camera sensor.

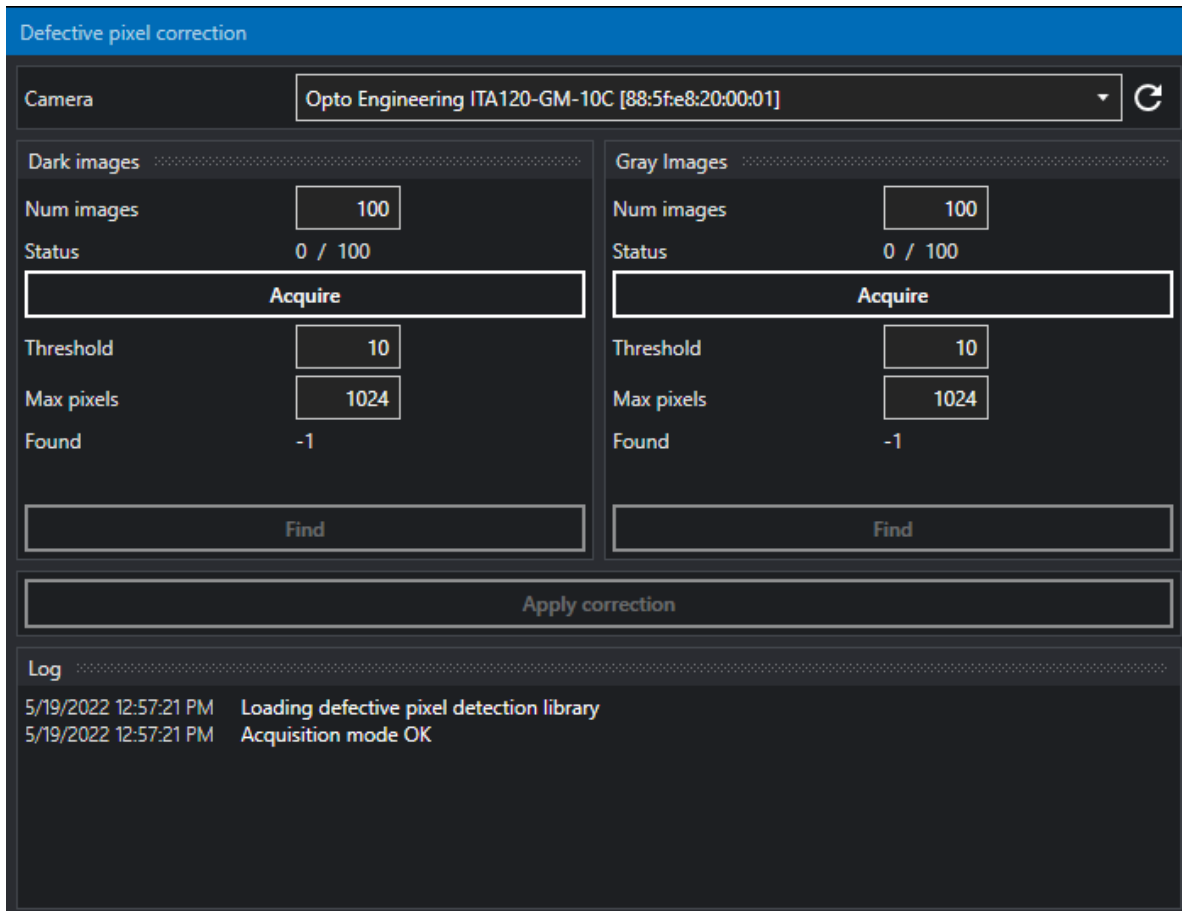


Figure 22: Defective pixels correction wizard.

5. Click on the *Acquire* button in the *Dark images* panel to acquire a first batch of dark images.
6. Click on the *Find* button on the same panel to detect the *leaky* pixels.
7. Expose the sensor to a uniform light source (suggested uniformity: >97%) in order to obtain an image with an average brightness of 50% of the maximum saturation level. You can adjust the exposure time to reach the desired level. **Keep in mind that the same exposure time should be used to acquire the dark images.**
8. Click on the *Acquire* button in the *Gray images* panel to acquire a second batch of gray images.
9. Click on the *Find* button on the same panel to detect the hot and cold pixels.
10. Click on *Apply correction* to upload the data to the camera.

In order to make this change permanent, you should save the current user set. Loading the default user set will restore the factory defective pixel correction.

4.7.6 Color correction wizard

From the *Wizard* menu you can access the *Color correction* wizard (Fig.23). With a reference color checker (Fig.24) is possible to calibrate the camera in specific light conditions and obtain an optimal color rendering. For more information about the color correction matrix refer to section 6.7.1.

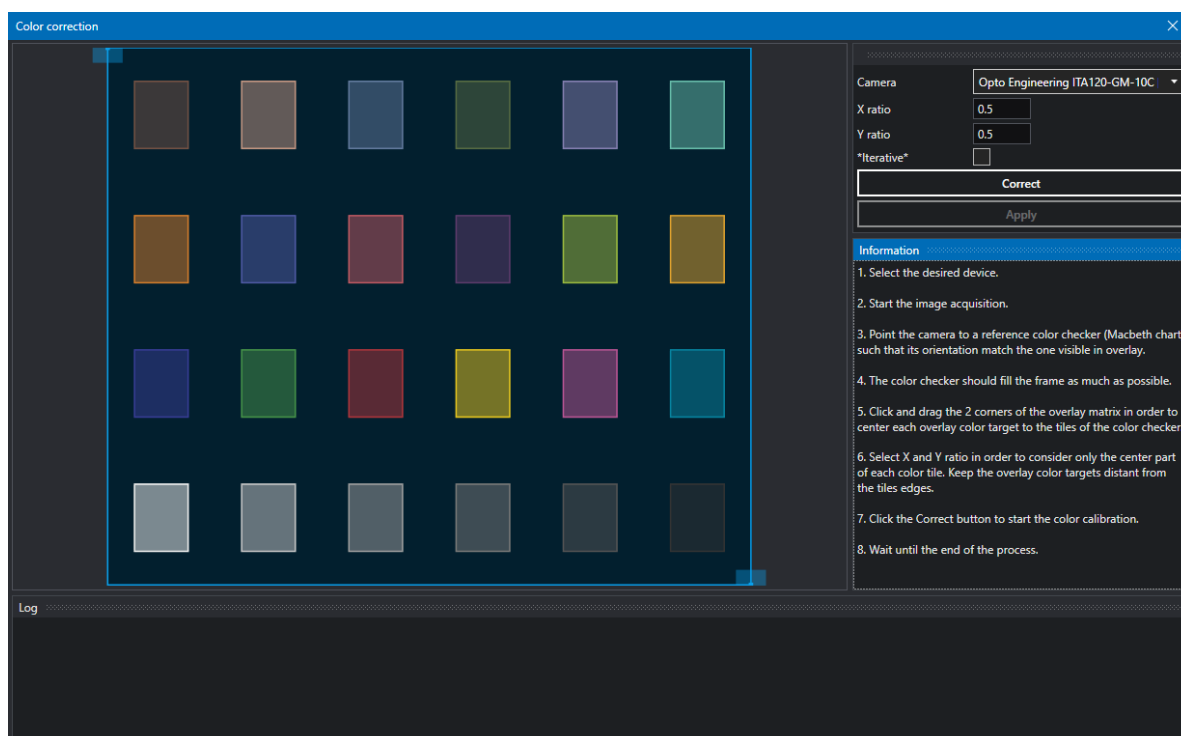


Figure 23: Color correction wizard.

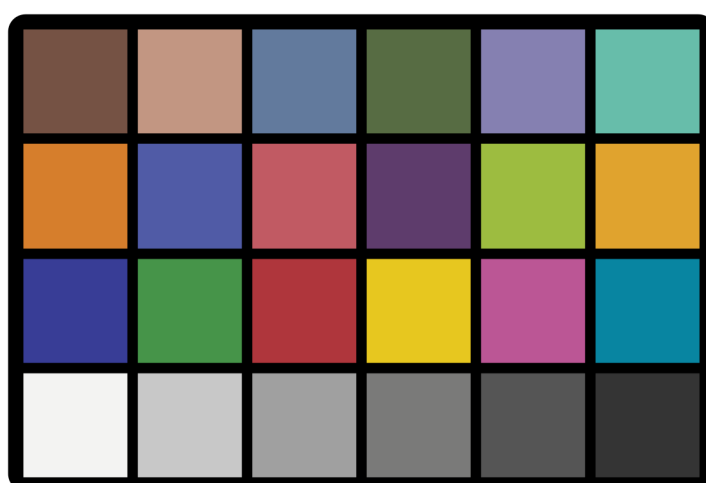


Figure 24: Reference color checker.

Follow these steps to achieve a correct calibration:

1. Select the desired device.
2. Start the image acquisition.
3. Open the Color Correction Wizard (*Wizard > Color correction*) (Fig.25).
4. Point the camera to a reference color checker (Macbeth chart) such that its orientation match the one visible in overlay.
5. The color checker should fill the frame as much as possible (Fig.26).
6. Click and drag the 2 corners of the overlay matrix in order to center each overlay color target to the tiles of the color checker.
7. Select X and Y ratio in order to consider only the center part of each color tile. Keep the overlay color targets distant from the tiles edges.
8. Click the *Correct* button to start the color calibration.
9. Wait until the end of the process.

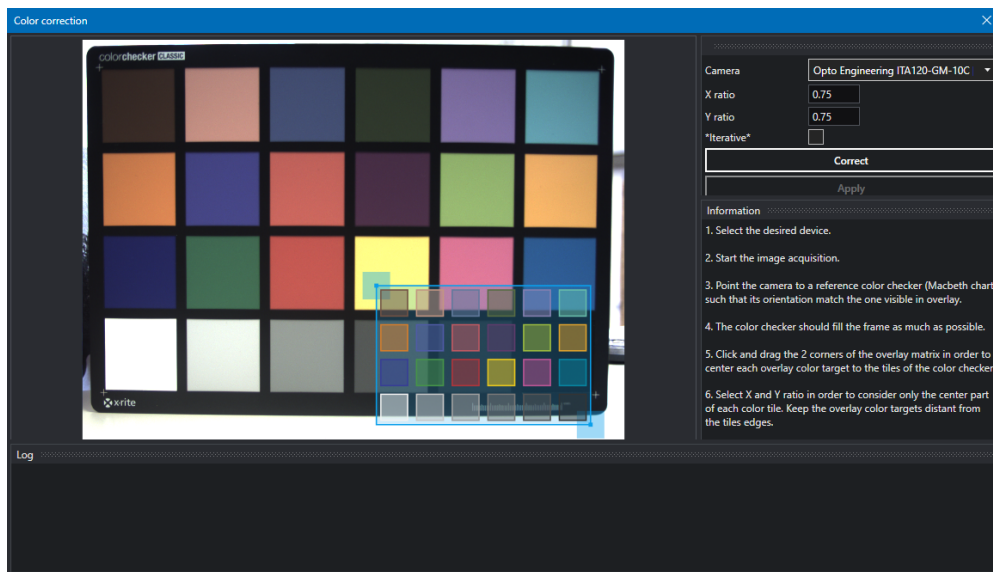


Figure 25: Window of the color correction wizard.

In order to make this change permanent, you should save the current user set. Loading the default user set will restore the factory color correction matrix.

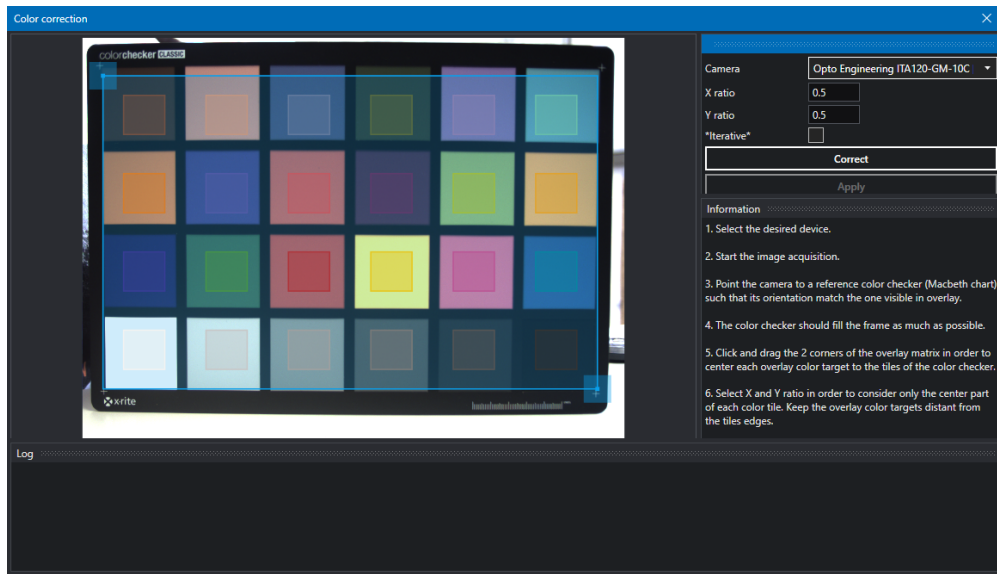


Figure 26: For optimal results, make sure to align the wizard color mask to the Macbeth color chart tiles.

5 TECHNICAL SPECIFICATIONS

5.1 Technical specifications

In Table 5.1 are summarized the main camera features.
Further explanations will be given in the following chapters.

| Features | Standard version | Liquid Lens version | Ref. page |
|----------------------------|--|--|-----------|
| IMAGE FEATURES | | | |
| ADC resolution | 10/12bit ¹ | 10/12bit ¹ | - |
| Image buffer size | ~ 384MB | ~ 384MB | - |
| Image mirror (X/Y) | ✓ | ✓ | p.67 |
| ROI mode | ✓ | ✓ | p.65 |
| Decimation/binning | ✓ | ✓ | p.65 |
| Pixel format | Mono8, Mono10Packed, Mono10p, Mono12Packed, Mono12p, RGB8, YUV422, YUV411, BayerRG8, BayerRG10p, BayerRG10Packed, BayerRG12p, BayerRG12Packed, Polarized ² | Mono8, Mono10Packed, Mono10p, Mono12Packed, Mono12p, RGB8, YUV422, YUV411, BayerRG8, BayerRG10p, BayerRG10Packed, BayerRG12p, BayerRG12Packed, Polarized ² | - |
| LUT/Gamma correction | ✓ | ✓ | p.81 |
| Test pattern | ✓ | ✓ | p.68 |
| Gain | ✓ | ✓ | p.79 |
| Black level | ✓ | ✓ | p.82 |
| Autoexposure | ✓ | ✓ | p.83 |
| Autogain | ✓ | ✓ | p.83 |
| Defective pixel correction | ✓ | ✓ | p.115 |
| Debayering | ✓ ³ | ✓ ³ | - |
| White balance | ✓ ³ | ✓ ³ | p.79 |
| Color correction matrix | ✓ ³ | ✓ ³ | p.87 |
| Chunk data | ✓ | ✓ | p.108 |
| CAMERA FEATURES | | | |
| Status LED indicator | ✓ | ✓ | p.60 |

| | | | |
|--|-------------------------|-------------------------|-------|
| OS compatibility | Windows 10, 11 (64 bit) | Windows 10, 11 (64 bit) | p.22 |
| PoE (Power over Ethernet) | ✓ | ✓ | - |
| Gigabit ethernet | ✓ | ✓ | - |
| Packet resend option | ✓ | ✓ | - |
| Static IP/DHCP | ✓ | ✓ | - |
| IEEE 1588 (PTP) | ✓ | ✓ | - |
| Opto-isolated inputs | 2 | 2 | p.59 |
| Opto-isolated outputs | 4 | 1 | p.59 |
| Temperature sensor | Image sensor, FPGA | Image sensor, FPGA | - |
| User sets | Factory + 2 user sets | Factory + 2 user sets | - |
| Remote FW update | ✓ | ✓ | p.34 |
| Burst acquisition | ✓ | ✓ | - |
| Trigger hardware | ✓ | ✓ | - |
| Trigger software | ✓ | ✓ | - |
| Timers | 2 | 2 | p.96 |
| Counters | 4 | 4 | p.96 |
| Encoder control | 1 ⁴ | 1 ⁴ | p.97 |
| Logic blocks | 4 | 4 | p.100 |
| Logic functions | OR, AND, LUT | OR, AND, LUT | p.101 |
| Serial communication | RS232/485 | | p.109 |
| Liquid Lens controller | | ✓ | p.111 |
| CERTIFICATIONS AND COMPLIANCES | | | |
| GigEVision compliance | ✓ | ✓ | - |
| GenICam compliance | ✓ | ✓ | - |
| CE certificate | ✓ | ✓ | p.11 |
| Shock and Vibrations | ✓ | ✓ | p.12 |
| RoHS | ✓ | ✓ | p.13 |
| REACH | ✓ | ✓ | p.13 |
| WEEE | ✓ | ✓ | p.13 |
| ENVIRONMENTAL | | | |
| Storage temperature | -10°C - 60°C | -10°C - 60°C | p.15 |
| Storage Humidity | RH < 80% | RH < 80% | p.15 |
| Operating case temperature ⁵ | -25°C - 65°C | -25°C - 65°C | p.15 |
| Operating ambient temperature ⁶ | -25°C - 50°C | -25°C - 50°C | p.15 |
| Operating Humidity | RH < 80% | RH < 80% | p.15 |

¹ Sensor specific data.

² With BayerRG pixel format, also BayerGR, BayerGB, BayerBG are included in the available pixel formats. Polarized pixel format includes both Mono and Color, 8, 10p, 10Packed, 12p and 12Packed variants.

³ Not available for monochrome sensors.

⁴ Refer to paragraph 6.10 for the compatible encoder interfaces.

⁵ Case temperature, measured on the front part of the camera body.

⁶ Maximum ambient temperature without lens and without heat dissipation. Higher operating temperature can be achieved with proper heat dissipation (i.e., mounting the camera through a metal plate).

5.2 Electrical specifications

Table 7 summarizes all the electrical specifications of the camera.

Further explanations about the I/O digital pins can be found in the paragraph 5.7 (pag.59).

| Parameter | MIN | TYP | MAX | UNIT |
|--------------------------------|-----|-----|-------------------|------|
| GENERAL | | | | |
| Supply Voltage | 12 | - | 24 | [V] |
| Power consumption ¹ | - | - | 5 | [W] |
| DIGITAL INPUT | | | | |
| Input voltage | 0 | - | 30 | [V] |
| Input HIGH voltage threshold | 2.2 | - | - | [V] |
| Input LOW voltage threshold | - | - | 1.9 | [V] |
| DIGITAL OUTPUT | | | | |
| Output voltage | 0 | - | +Vcc ² | [V] |
| Output current | - | - | 50 ³ | [mA] |

¹ Maximum power to be used to supply product PS2 per 62368-1 and LPS per Annex Q.1 less than 100W.

² External power supply connected to the digital output + pin. Maximum power to be used to supply product PS2 per 62368-1 and LPS per Annex Q.1 less than 100W.

³ Output current must be limited by the external load or an appropriate external resistor.

Table 7: Electrical specifications

In Table 8 are shown the results of the minimum/maximum input delays when Itala camera is triggered by an external device: the column *Input voltage* defines the triggering-signal voltage level, while the columns *MIN Input delay* and *MAX Input delay* show the signal propagation time from the digital input, on the connector, to the sensor acquisition trigger, in the case the trigger overlap is disabled. The difference between the maximum and minimum delay is the jitter due to the sampling of the input trigger signal. A high performance input isolator is used in the design in order to handle the input trigger signal with minimum delay.

| Input voltage (V) | MIN Input delay (μ s) | MAX Input delay (μ s) |
|---|----------------------------|----------------------------|
| TRIGGERING DEVICE (RISING-EDGE LOGIC) | | |
| 3.3 | 1.5 | 2.5 |
| 5 | 1.5 | 2.5 |
| 12 | 1.5 | 2.5 |
| 24 | 1.5 | 2.5 |
| TRIGGERING DEVICE (FALLING-EDGE LOGIC) | | |
| 3.3 | 1.5 | 2.5 |

| | | |
|----|-----|-----|
| 5 | 1.5 | 2.5 |
| 12 | 1.5 | 2.5 |
| 24 | 1.5 | 2.5 |

Table 8: Minimum and Maximum input delays when Itala cameras are triggered by an external device

In Table 9 are shown the results of the minimum/maximum output delays when Itala camera triggers external devices: the column *Supply Voltage* refers to the voltage connected to the *OPTO REF V+* pin while the columns *Load Resistance* indicates the load connected to the opto-isolated output port.

| Supply Voltage (V) | Load Res (ohm) | MAX Output delay (μs) | Meas Output Voltage (V) | Meas Output Current (mA) |
|--|----------------|------------------------------|-------------------------|--------------------------|
| CAMERA OUTPUT SIGNAL (RISING-EDGE LOGIC) | | | | |
| 3.3 | 150 | 3.5 | 2.1 | 14.0 |
| | 330 | 3.5 | 3.0 | 8.8 |
| | 560 | 3.4 | 3.2 | 5.6 |
| | 1000 | 3.3 | 3.2 | 3.2 |
| 5 | 330 | 3.5 | 4 | 12.1 |
| | 560 | 3.5 | 4.5 | 8.0 |
| | 1000 | 3.5 | 4.7 | 4.7 |
| | 2200 | 3.4 | 4.8 | 2.2 |
| 12 | 330 | 3.8 | 8.5 | 25.6 |
| | 560 | 3.8 | 10.2 | 18.2 |
| | 1000 | 3.7 | 11.4 | 11.3 |
| | 2200 | 3.6 | 12.0 | 5.4 |
| 24 | 560 | 4.0 | 16.5 | 29.5 |
| | 1000 | 4.0 | 21.0 | 21.0 |
| | 2200 | 3.9 | 23.2 | 10.6 |
| | 4700 | 3.8 | 23.7 | 5.1 |
| CAMERA OUTPUT SIGNAL (FALLING-EDGE LOGIC) | | | | |
| 3.3 | 150 | 6.6 | 2.1 | 14.2 |
| | 330 | 17.3 | 3.0 | 9.1 |
| | 560 | 27.3 | 3.1 | 5.6 |
| | 1000 | 34.8 | 3.2 | 3.2 |
| 5 | 330 | 9.6 | 4.0 | 12.2 |
| | 560 | 20.6 | 4.6 | 8.2 |

| | | | | |
|----|------|------|------|------|
| | 1000 | 30.7 | 4.7 | 4.7 |
| | 2200 | 42.2 | 4.8 | 2.2 |
| 12 | 330 | 1.8 | 8.4 | 25.6 |
| | 560 | 4.7 | 10.2 | 18.2 |
| | 1000 | 12.0 | 11.4 | 11.4 |
| | 2200 | 31.3 | 12.0 | 5.4 |
| 24 | 560 | 1.5 | 17.0 | 30.3 |
| | 1000 | 4.0 | 21.2 | 21.2 |
| | 2200 | 15.8 | 23.3 | 10.6 |
| | 4700 | 36.9 | 23.8 | 5.1 |

Table 9: Maximum output delays and electrical measurements for Itala cameras output signals

5.3 Sensor optical response

Typically all the 1st generation Sony IMX sensors exhibit the same optical response. The relative transmittance for monochrome and color sensors is shown in Fig.27 and Fig.28. Please refer to Section 1.5 to retrieve the information about the image sensor family.

Typically all the 2nd generation Sony IMX sensors exhibit the same optical response. The relative transmittance for monochrome and color sensors is shown in Fig.29 and Fig.30. Please refer to Section 1.5 to retrieve the information about the image sensor family.

Typically all the 4th generation Sony IMX sensors exhibit, on average, the same optical response. The relative transmittance for monochrome and color sensors is shown in Fig.31 and Fig.32. Please refer to Section 1.5 to retrieve the information about the image sensor family.

The relative transmittance for Sony IMX990 SenSWIR™ sensor is shown in Fig.33. Please refer to Section 1.5 to retrieve the information about the image sensor family.

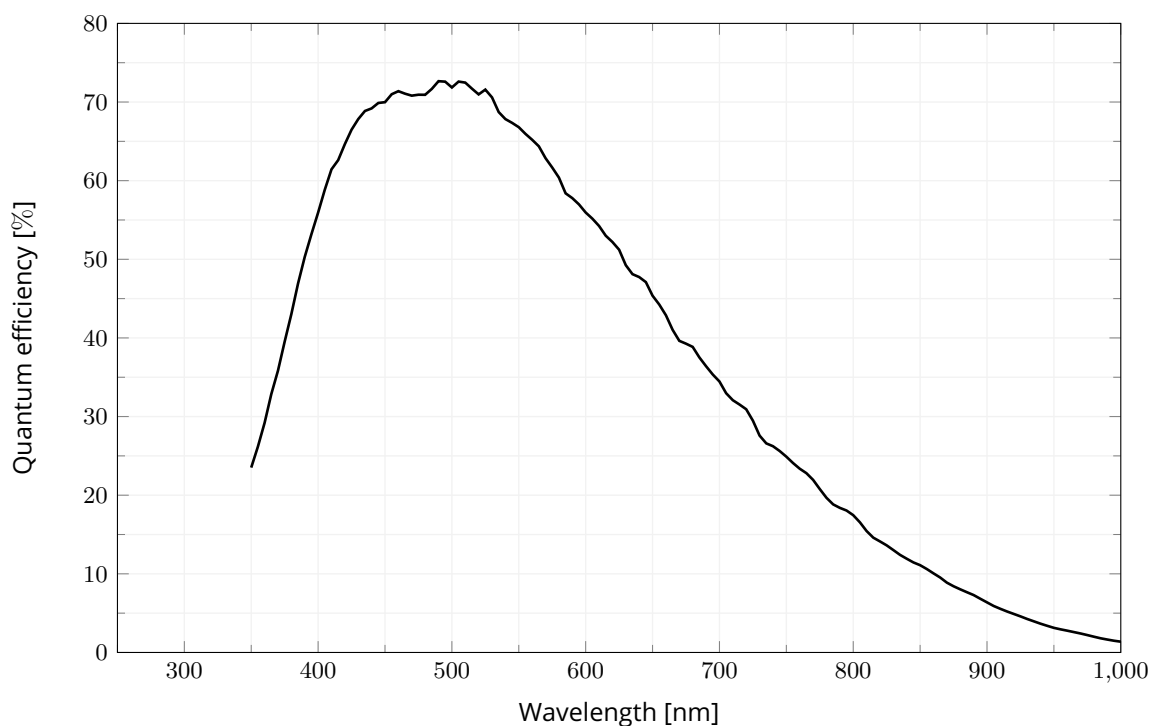


Figure 27: 1st gen Sony IMX monochrome sensors.

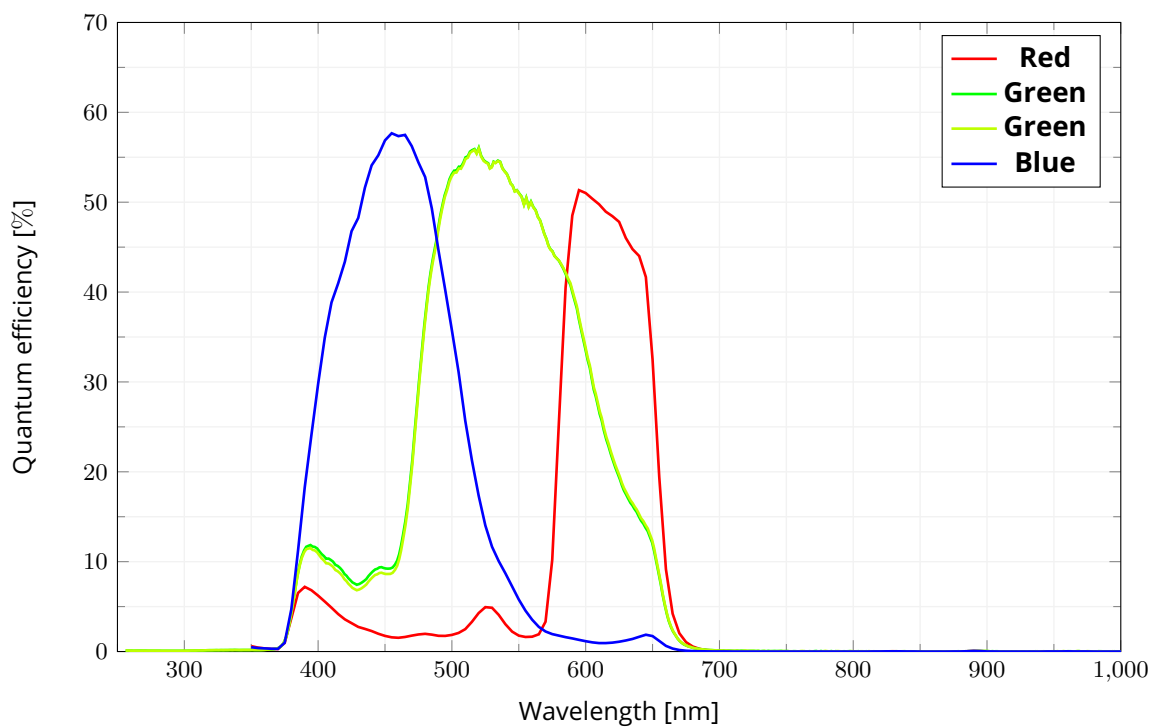


Figure 28: 1st gen Sony IMX color sensors.

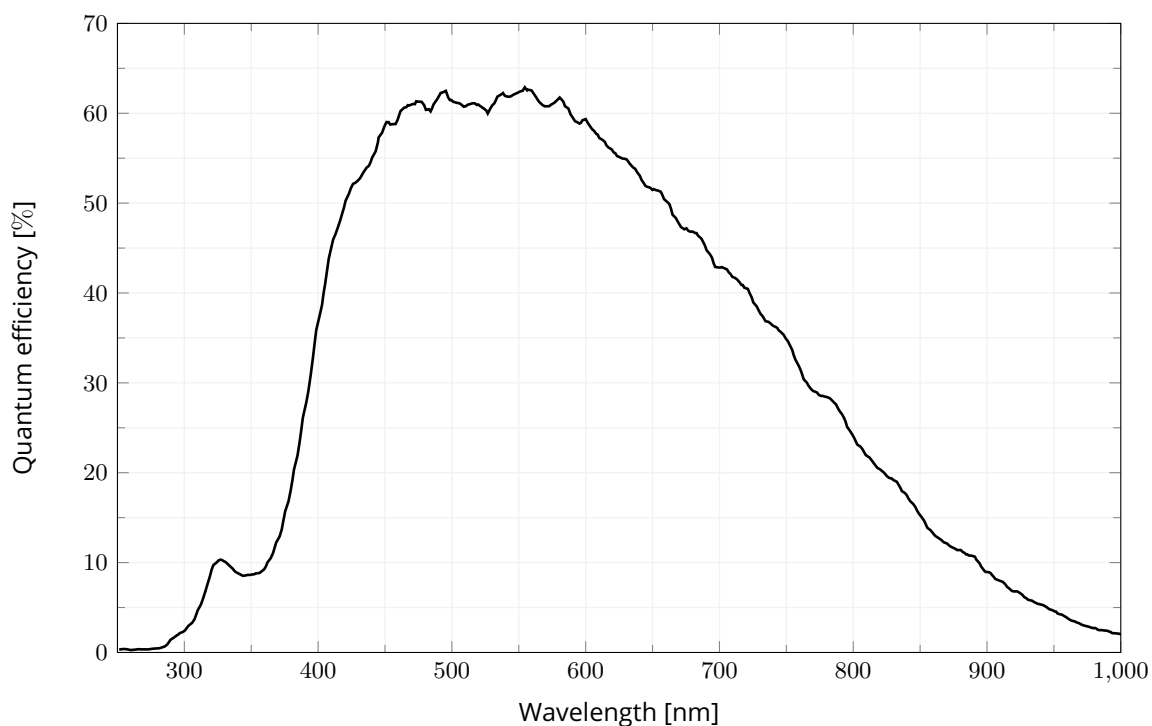


Figure 29: 2nd gen Sony IMX monochrome sensors.

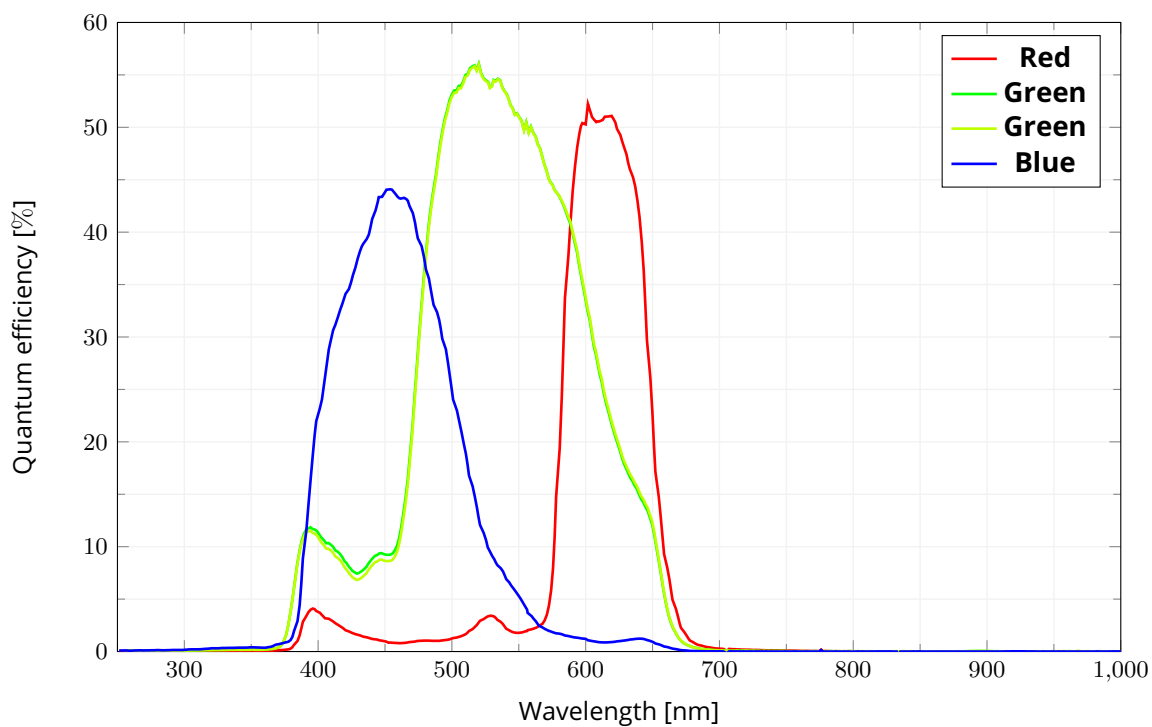


Figure 30: 2nd gen Sony IMX color sensors.

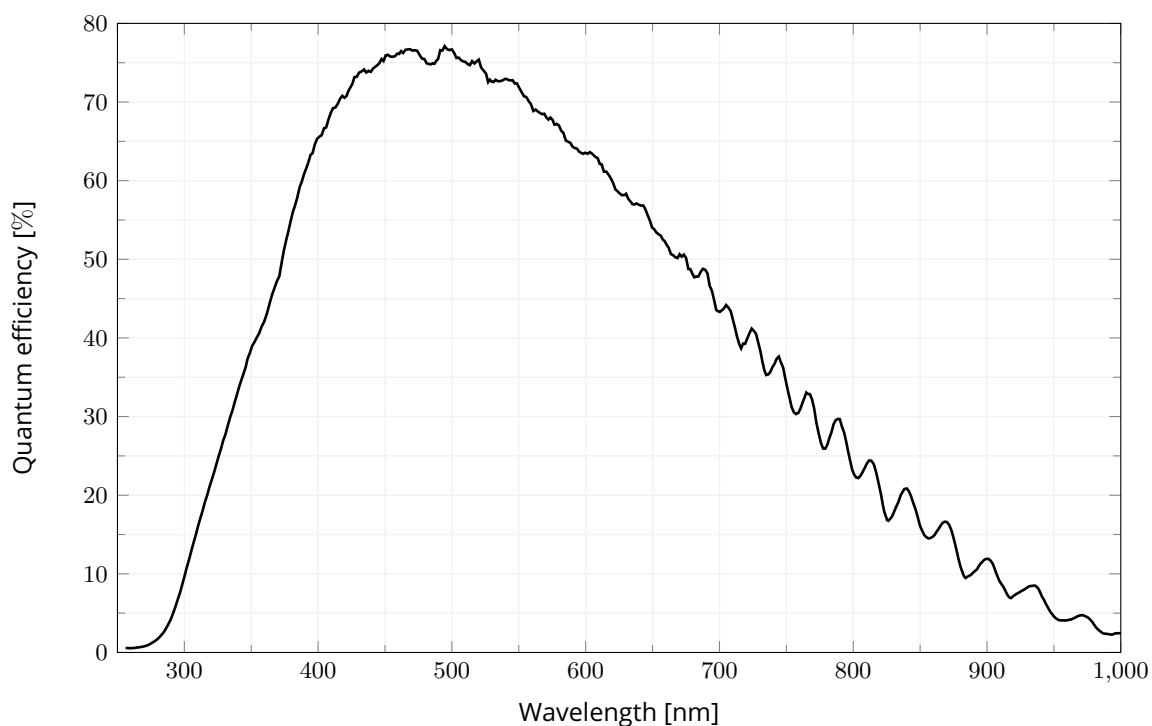


Figure 31: 4th gen Sony IMX monochrome sensors.

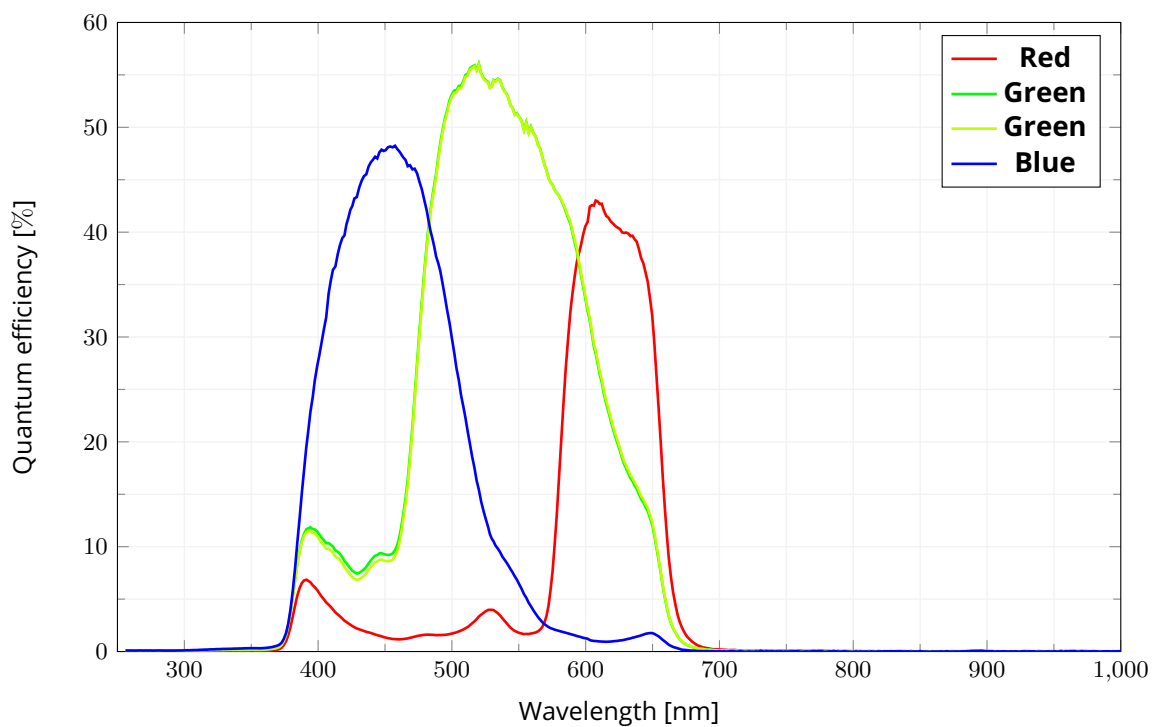


Figure 32: 4th gen Sony IMX color sensors.

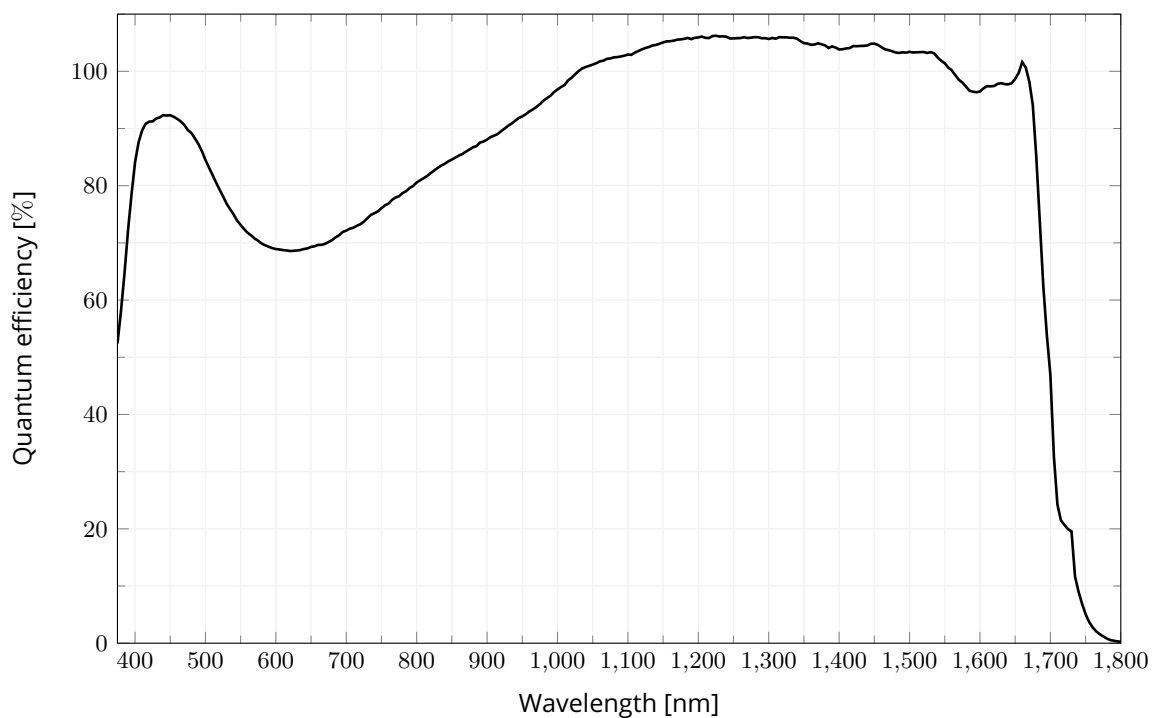
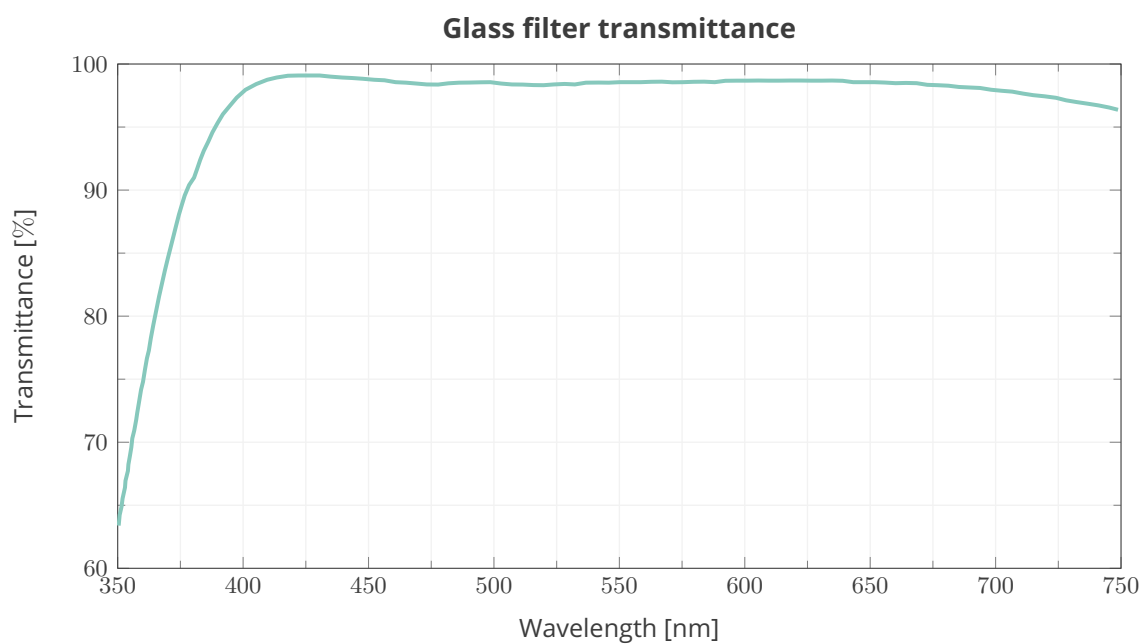
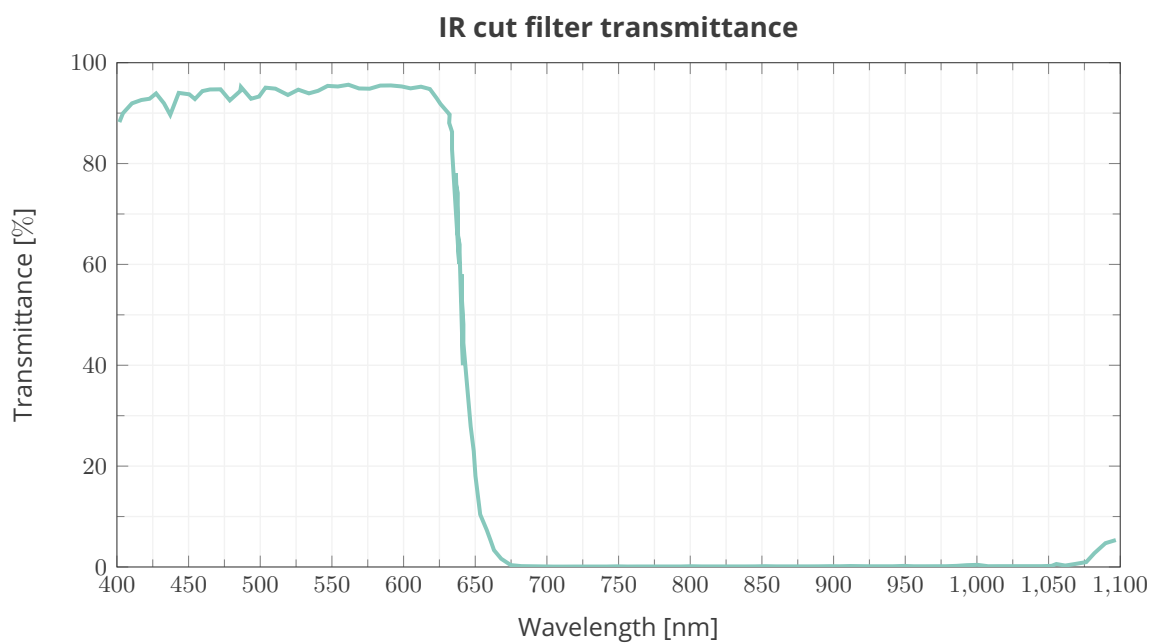


Figure 33: Sony IMX990 SenSWIR™ monochrome sensor.

5.4 Optical filters

The following figures show the transmittance characteristics of the available optical filters for the Itala camera series.





5.5 Mechanical specifications

5.5.1 Dimensional drawings

ITALA G - G.EL

The **TYPE 1** drawings (Figure 34) refers to all cameras with sensors up to 1.2" image format. These are equipped with a standard **C mount** (1 inch diameter, 32 threads per inch), with a flange distance of **17.526 mm**.

The **TYPE 2** drawings (Figure 35) refers to all cameras with sensors from 4/3" to APS-C image format. These are equipped with an **M42x1** threaded mount, with a flange distance of **12 mm**.

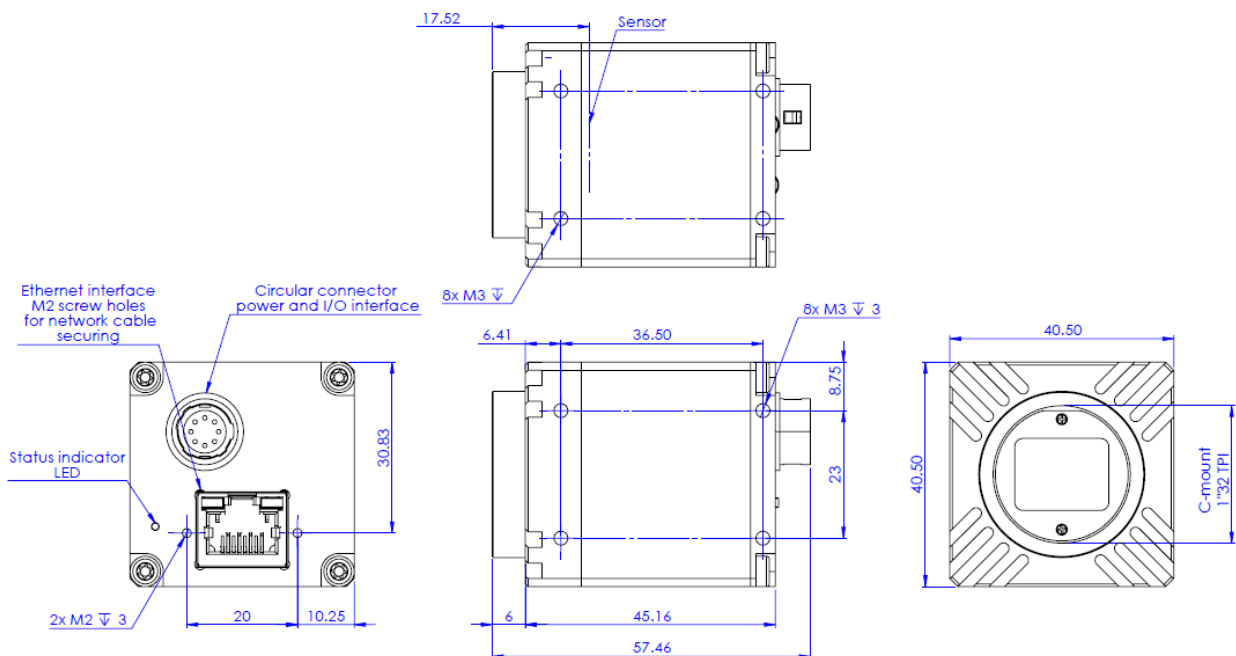


Figure 34: TYPE 1 dimensional drawings.

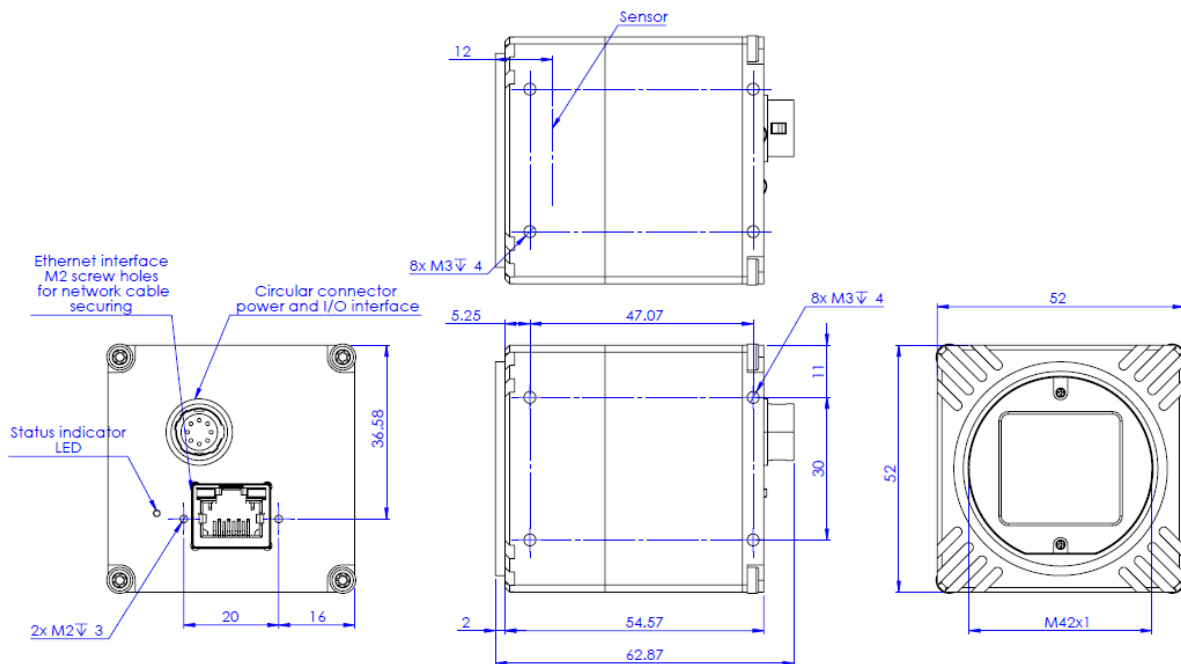


Figure 35: TYPE 2 dimensional drawings.

ITALA G.IP

The drawings in Figure 36 refers to all the IP67 cameras. These are equipped with a standard **C mount** (1 inch diameter, 32 threads per inch), with a flange distance of **17.526 mm**.

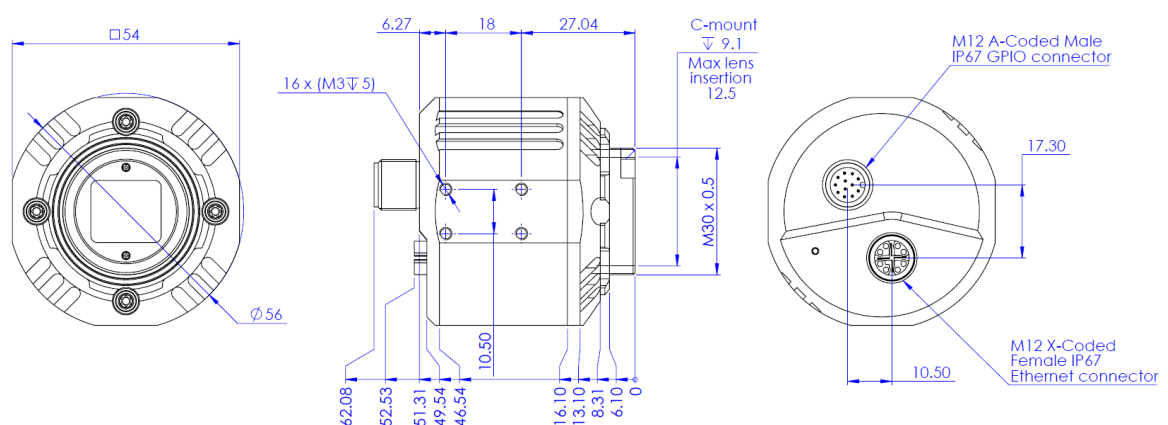


Figure 36: ITALA G.IP dimensional drawings.

5.5.2 GigE Vision mechanical requirements

Itala cameras comply with GigE Vision mechanical requirements. In particular, the standard adopted is the one defined as *TYPE090* in the *GigE Vision Mechanical Supplement*.

5.5.3 Sensor centering data

All cameras are tested after assembly to make sure the sensor is correctly centered. Measurements are taken in all six degrees of freedom relative to the lens mount to ensure optimal imaging performance. In Table 10 and 11 are reported typical production values for the sensor centering relative to Figure 37.

| Axis | Nominal | 3 σ |
|----------------------|----------|------------|
| Roll | 0° | 0.4° |
| Yaw | 0° | 0.3° |
| Pitch | 0° | 0.5° |
| Horizontal shift (x) | 0 mm | 0.2 mm |
| Vertical shift (y) | 0 mm | 0.2 mm |
| FD (z) | 17.53 mm | 0.11 mm |

Table 10: Sensor centering for C-mount cameras

| Axis | Nominal | 3 σ |
|----------------------|---------|------------|
| Roll | 0° | 0.6° |
| Yaw | 0° | 0.3° |
| Pitch | 0° | 0.5° |
| Horizontal shift (x) | 0 mm | 0.2 mm |
| Vertical shift (y) | 0 mm | 0.2 mm |
| FD (z) | 12 mm | 0.2 mm |

Table 11: Sensor centering for J-mount cameras

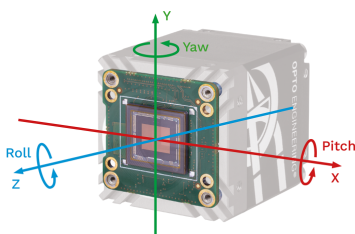


Figure 37: Sensor degrees of freedom.

5.6 Connectors and pinout

ITALA G - G.EL

The camera has two connectors:

- **Standard RJ45 connector with screw locks**

Connection for image streaming and (optionally) for camera powering via PoE.

- **12-pin circular connector (P/N: HR10G-10R-12PB(71))**

This connector has multi-purpose pins: power supply, trigger, synchronism, serial communication, liquid lens driver. The pinout is not fixed and depends on the camera model (standard or with liquid lens controller). Refer to Table 12 to see the pinout for both the camera models.

| PIN | Standard | Liquid lens |
|-----|--------------|--------------|
| 1 | GND | GND |
| 2 | +VIN | +VIN |
| 3 | Opto OUT 3 | Lens - |
| 4 | Opto IN 0 | Opto IN 0 |
| 5 | Opto OUT 2 | Lens + |
| 6 | Opto OUT 0 | Opto OUT 0 |
| 7 | Opto REF GND | Opto REF GND |
| 8 | RS232 RX | Lens SCL |
| 9 | RS232 TX | Lens SDA |
| 10 | Opto REF V+ | Opto REF V+ |
| 11 | Opto IN 1 | Opto IN 1 |
| 12 | Opto OUT 1 | Lens +3.3V |

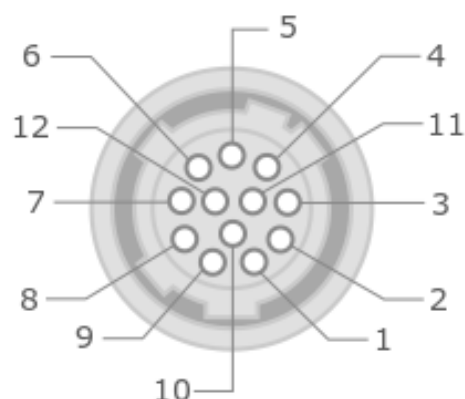


Table 12: Itala pinout for both standard and liquid lens controller version.

Figure 38: 12 pin circular connector pinout (camera front view)

NOTE: If a CBGPIO001 cable is used, check Opto Engineering® website to get the "color vs function" association.

ITALA G.IP

The camera has two connectors:

- **M12 X-Coded Female IP67 Ethernet connector (P/N: 394811-E)**
Connection for image streaming and (optionally) for camera powering via PoE.
- **M12 A-Coded Male IP67 GPIO connector (P/N: 494518-E)**
This connector has multi-purpose pins: power supply, trigger, synchronism, serial communication.

| PIN | Description |
|-----|--------------|
| 1 | GND |
| 2 | +VIN |
| 3 | Opto OUT 3 |
| 4 | Opto IN 0 |
| 5 | Opto OUT 2 |
| 6 | Opto OUT 0 |
| 7 | Opto REF GND |
| 8 | RS232 RX |
| 9 | RS232 TX |
| 10 | Opto REF V+ |
| 11 | Opto IN 1 |
| 12 | Opto OUT 1 |

Table 13: Itala pinout for IP67 version.

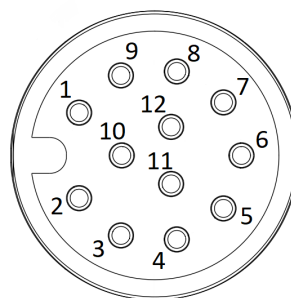


Figure 39: IP67 12-pin circular connector pinout (camera front view).

NOTE: If a RT-MSAS-12BFFM-SL8Dxx cable is used, check Opto Engineering® website to get the "color vs function" association.

5.7 I/O circuitry

All input and output pins of the I/O connector are galvanically isolated.

All the electrical specifications and the maximum voltage/current ratings are listed in Table 7 (pag.46).

5.7.1 Opto Isolated Input

The opto-isolated input topology is schematically shown in Figure 40.

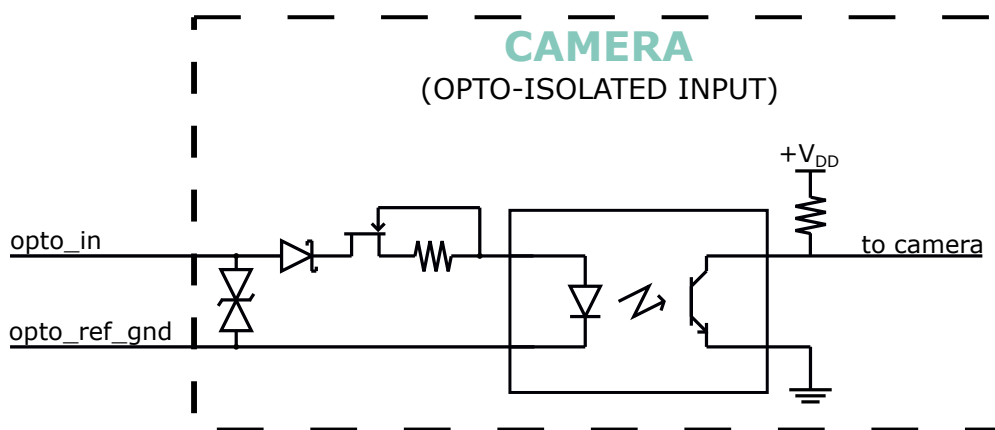


Figure 40: Opto isolated input topology.

A high-speed input isolator is used in the design to handle the input trigger signal with minimum propagation delay. A TVS diode is used as countermeasure against high voltage spikes, while a series diode prevents input polarity inversion. In addition, a current limiter circuit is also included in order to automatically adjust the input current.

NOTE: Be careful that any damage to opto-isolated input circuitry makes it not usable anymore.

Some wiring diagram examples (both for opto-isolated and not-isolated systems) are shown in the "Wiring connection examples" chapter (7.1).

5.7.2 Opto Isolated Output

The opto-isolated output topology is schematically shown in Figure 41.

In case of accidental connection on the output pins a bypass diode protects the transistor providing an alternate path for the current. The reference voltage for the opto-isolated output pins can be different from the power supply voltage, however the maximum specifications listed in Table 7 must not be exceeded.

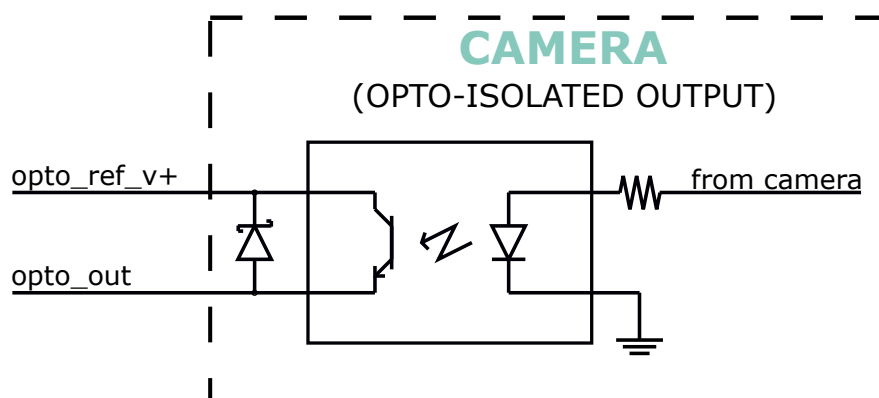


Figure 41: Opto isolated output topology.

5.8 LED and indicators

Itala is equipped with a LED indicator on the back side of the camera, next to the connectors. This indicator shows the current status of the camera. The color codes are listed in Table 14.

| Color | | Camera status |
|-------|---------------|---|
| ○ | White | - |
| ● | Blue | Camera is configuring (after a FW update) |
| ● | Purple | Camera in boot mode |
| ● | Cyan flashing | Camera is updating |
| ● | Yellow | Camera is ready |
| ● | Green | Camera is triggered |
| ● | Red | Error |

Table 14: LED colors vs status of the camera

6 CAMERA FEATURES

This chapter provides a summary of the standard and custom features of the Itala camera series. Features are defined following the *Standard Feature Naming Convention (SFNC)* and *GenICam* nomenclature. The following sections provide more detailed explanation of each feature.

6.1 Device Control

This section contains the features related to the control and information of the device. This is mainly used to identify the device during the enumeration process and to obtain information about the device itself.

In Table 15 are listed all the Device Control parameters.

| Feature | Description | Interface | Access |
|------------------------|--|--------------|--------|
| DeviceType | Returns the device type | IEnumeration | R |
| DeviceScanType | Scan type of the sensor of the device | IEnumeration | R |
| DeviceVendorName | Name of the manufacturer of the device | IString | R |
| DeviceModelName | Model of the device | IString | R |
| DeviceManufacturerInfo | Manufacturer information about the device | IString | R |
| DeviceVersion | Version of the device | IString | R |
| DeviceFirmwareVersion | Version of the firmware in the device | IString | R |
| DeviceSerialNumber | Device serial number | IString | R |
| DeviceUserID | User-programmable device identifier | IString | RW |
| DeviceTLType | Transport Layer type of the device | IEnumeration | R |
| DeviceTLVersionMajor | Major version of the Transport Layer of the device | Integer | R |
| DeviceTLVersionMinor | Minor version of the Transport Layer of the device | Integer | R |
| DeviceLinkSelector | Selects which Link of the device to control | Integer | RW |
| DeviceLinkSpeed | Indicates the speed of transmission negotiated on the specified Link | Integer | R |

| | | | |
|-------------------------------|--|--------------|----|
| DeviceLinkThroughputLimitMode | Controls if the DeviceLinkThroughputLimit is active | IEnumeration | RW |
| DeviceLinkThroughputLimit | Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link | IInteger | RW |
| DeviceLinkHeartbeatMode | Activate or deactivate the Link's heartbeat | IEnumeration | RW |
| DeviceLinkHeartbeatTimeout | Controls the current heartbeat timeout of the specific Link | IFloat | RW |
| DeviceLinkCommandTimeout | Indicates the command timeout of the specified Link. This corresponds to the maximum response time of the device for a command sent on that Link | IFloat | RW |
| DeviceReset | Resets the device to its power up state. After reset, the device must be rediscovered. Note that some Transport Layers require the acknowledgement of the DeviceReset command before starting the actual reset of the device | ICommand | W |
| DeviceFeaturePersistenceStart | Indicate to the device and GenICam XML to get ready for persisting of all streamable features | ICommand | W |
| DeviceFeaturePersistenceEnd | Indicate to the device the end of feature persistence | ICommand | W |
| DeviceRegistersStreamingStart | Prepare the device for registers streaming without checking for consistency | ICommand | W |
| DeviceRegistersStreamingEnd | Announce the end of registers streaming | ICommand | W |
| DeviceTemperatureSelector | Selects the location within the device, where the temperature will be measured | IEnumeration | RW |
| DeviceTemperature | Device temperature in degrees Celsius (°C) | IFloat | R |
| oeDevicePressure | Device internal pressure in hPa | IFloat | R |

Table 15: Device Control features

6.1.1 Bandwidth limit

The **DeviceLinkThroughputLimit** feature allows to limit the bandwidth available for the camera data streaming. Delays will be uniformly inserted between transport layer packets in order to control the peak bandwidth. This is equivalent to directly set the inter-packet delay value through the **GevSCPD** feature in the **Transport Layer Control** section. A suitable delay in data transfer will prevent the camera from "overrunning" the transfer interface limit.

The bandwidth limit is especially useful when setting up a multi-camera system with an installed bandwidth lower than the sum of the bandwidth of the single device. Setting a proper limit on each device ensures the lowest amount of collisions on the network, maximizing performances and improving overall stability.

6.2 Image Format Control

The Image Format Control section describes how to configure image size and format.

| Feature | Description | Interface | Access |
|-------------------|--|-----------|--------|
| SensorWidth | Effective width of the sensor in pixels | Integer | R |
| SensorHeight | Effective height of the sensor in pixels | Integer | R |
| SensorPixelWidth | Physical size (pitch) in the x direction of a photo sensitive pixel unit | IFloat | R |
| SensorPixelHeight | Physical size (pitch) in the y direction of a photo sensitive pixel unit | IFloat | R |
| SensorName | Product name of the imaging Sensor | IString | R |
| WidthMax | Maximum width of the image (in pixels) | Integer | R |
| HeightMax | Maximum height of the image (in pixels) | Integer | R |
| Width | Width of the image provided by the device (in pixels) | Integer | RW |
| Height | Height of the image provided by the device (in pixels) | Integer | RW |

| | | | |
|--------------------------|--|-------------|----|
| OffsetX | Horizontal offset from the origin to the region of interest (in pixels) | Integer | RW |
| OffsetY | Vertical offset from the origin to the region of interest (in pixels) | Integer | RW |
| BinningHorizontalMode | Sets the mode to use to combine horizontal photo-sensitive cells together when BinningHorizontal is used | Enumeration | RW |
| BinningHorizontal | Number of horizontal photo-sensitive cells to combine together | Integer | RW |
| BinningVerticalMode | Sets the mode to use to combine vertical photo-sensitive cells together when BinningVertical is used | Enumeration | RW |
| BinningVertical | Number of vertical photo-sensitive cells to combine together | Integer | RW |
| DecimationHorizontalMode | Sets the mode used to reduce the horizontal resolution when DecimationHorizontal is used | Enumeration | RW |
| DecimationHorizontal | Horizontal sub-sampling of the image | Integer | RW |
| DecimationVerticalMode | Sets the mode used to reduce the vertical resolution when DecimationVertical is used | Enumeration | RW |
| DecimationVertical | Vertical sub-sampling of the image | Integer | RW |
| ReverseX | Flip horizontally the image sent by the device | Boolean | RW |
| ReverseY | Flip vertically the image sent by the device | Boolean | RW |
| PixelFormat | Format of the pixels provided by the device | Enumeration | RW |
| TestPattern | Selects the type of test pattern that is generated by the device as image source | Enumeration | RW |

Table 16: Image Format Control features

6.2.1 Image ROI

The **Width**, **Height**, **OffsetX**, **OffsetY** parameters are used to change the image format and to stream only a part of the full resolution image: in particular, the offsets set the displacement of the ROI (region of interest), while the width and height parameters set the effective dimensions of the image.

The sum of **OffsetX** and **Width** cannot exceed the **WidthMax** value and the sum of **OffsetY** and **Height** cannot exceed the **HeightMax**.

WidthMax and **HeightMax** are sensor specific and cannot be set by the user.

In Fig.42 is shown a graphical explanation of these parameters.

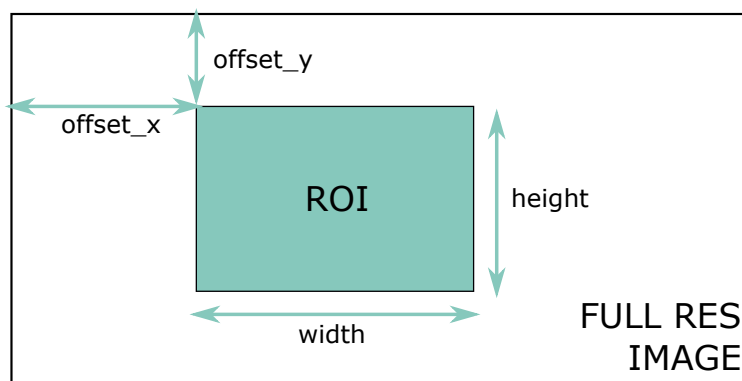


Figure 42: Image ROI parameters.

6.2.2 Binning

Binning mode is used to increase camera sensitivity by summing the values of adjacent pixels, with the drawback of a lower effective resolution.

As depicted in Fig.43, a **2x1 binning** halves the image resolution along the x-axis, but the overall image brightness is doubled (since two adjacent pixels have been combined). If a **2x2 binning** is performed, the image resolution is one quarter of the initial one, but the brightness is four times the initial brightness.

In case of color sensors, Bayer filter must be taken into account: since adjacent pixels have different chroma information, binning is performed on alternate pixels, as depicted in Fig.44. In this way, chroma information is not affected by algorithm artifacts.

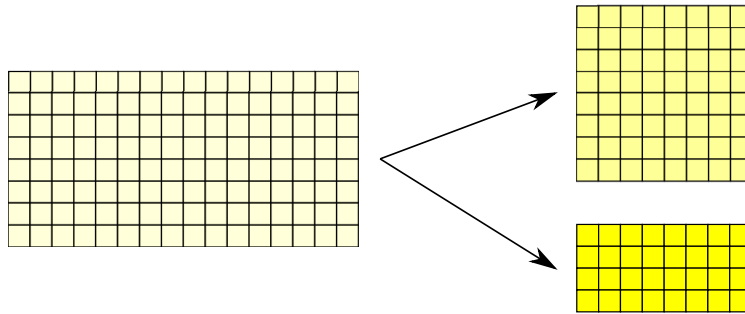


Figure 43: Examples of binning: in the figure above a 2x1 binning is performed, while in the figure below a 2x2 binning is applied.

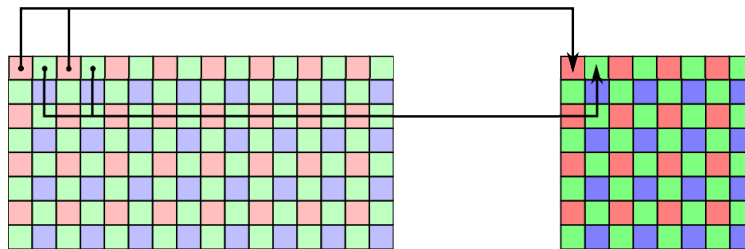


Figure 44: Examples of binning for color sensors: in the figure above a 2x1 binning is performed.

6.2.3 Decimation

Decimation mode is used to discard pixels in order to obtain a sub-sampled image.

Decimation mode has some advantages, e.g. the increasing of the frame rate of the camera.

In Fig.45 are shown two examples of decimation: in the figure above a **2x1 decimation** is performed: only one pixel over two is considered, thus the resulting image has half of the initial horizontal resolution; in the figure below a **4x1 decimation** has been applied, so only one pixel over four is acquired. Also in this case the resulting horizontal resolution has been reduced (by a factor 4).

In case of color sensors, Bayer filter must be taken into account: since adjacent pixels have different chroma information, decimation is performed grouping pixels with alternate colors, as depicted in Fig.46. In this way, chroma information is not affected by algorithm artifacts.

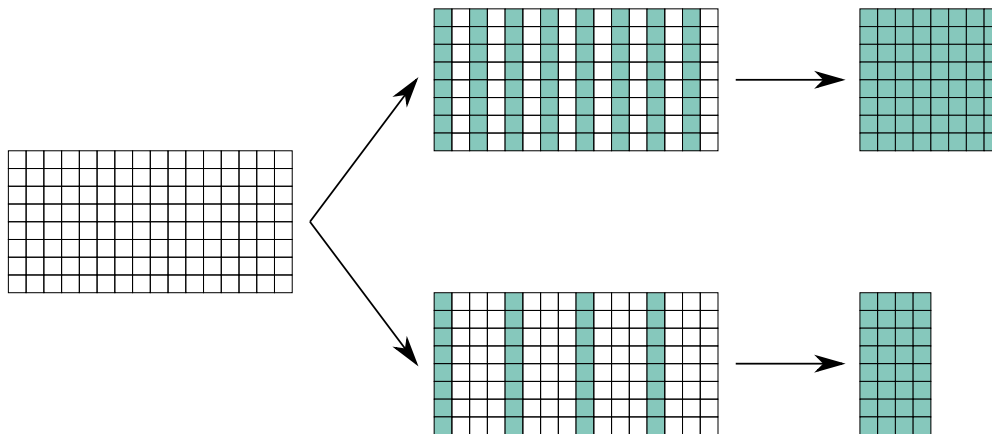


Figure 45: Examples of decimation: in the figure above a 2x1 decimation is performed, while in the figure below a 4x1 decimation is applied.

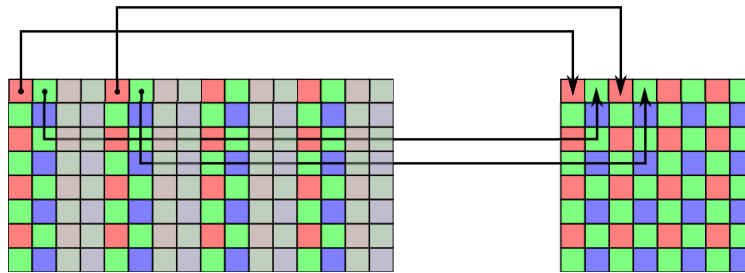


Figure 46: Examples of decimation for color sensors: in the figure above a 2x1 decimation is performed.

6.2.4 Readout direction

The camera supports image mirroring in both horizontal and vertical directions in order to make the integration of the camera insensitive to the mounting position.

In Fig.47 is shown an example of the **ReverseX** and **ReverseY** features.

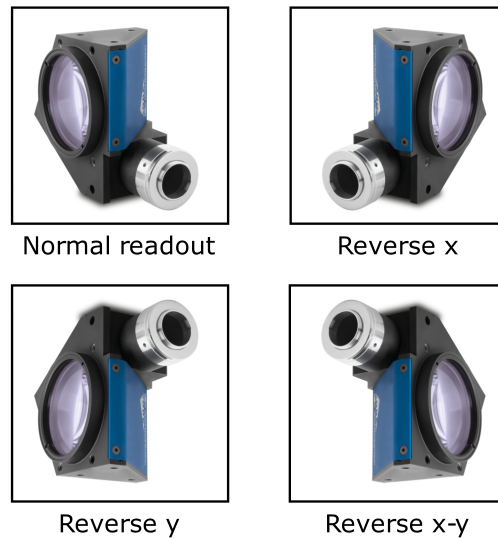


Figure 47: Representation of all the four possible readout modes.

6.2.5 Test pattern

Itala cameras support two different test patterns, one monochrome and one color. The two different test patterns are represented in Fig. 48 and 49.

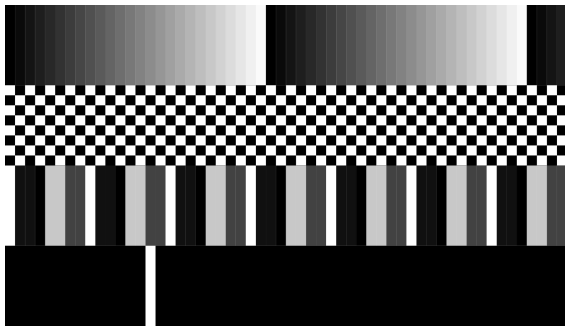


Figure 48: Monochrome test pattern

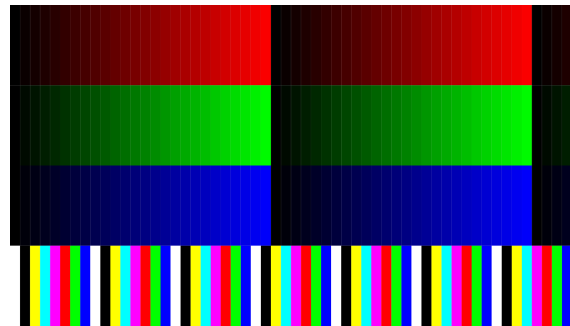


Figure 49: Color test pattern

The **monochrome pattern** is characterized by 4 different section:

- a fixed gradient pattern, from black to white;
- a fixed chess pattern;
- a fixed asymmetrical bars pattern (values: 0xC8, 0x10, 0x10, 0x42);
- a white moving line on a black background.

The **color pattern** is characterized by 4 different section:

- a fixed red gradient pattern, from black to red;
- a fixed green gradient pattern, from black to green;

- a fixed blue gradient pattern, from black to blue;
- a fixed color bars pattern (all the possible combinations of R, G and B coordinates).

6.3 Acquisition Control

The Acquisition Control section describes all features related to image acquisition, including the trigger and exposure control. It describes the basic model for acquisition and the typical behavior of the device.

| Feature | Description | Interface | Access |
|---------------------------------|---|--------------|--------|
| AcquisitionMode | Sets the acquisition mode of the device. It defines mainly the number of frames to capture during an acquisition and the way the acquisition stops | IEnumeration | RW |
| AcquisitionStart | Starts the Acquisition of the device | ICommand | RW |
| AcquisitionStop | Stops the Acquisition of the device at the end of the current Frame | ICommand | RW |
| AcquisitionBurstFrameCount | Number of frames to acquire for each FrameBurstStart trigger | IInteger | RW |
| AcquisitionFrameRate | Controls the acquisition rate (in Hertz) at which the frames are captured | IFloat | RW |
| AcquisitionFrameRateEnable | Controls if the AcquisitionFrameRate feature is writable and used to control the acquisition rate. Otherwise, the acquisition rate is implicitly controlled by the combination of other features like ExposureTime, etc.. | IBoolean | RW |
| oeAcquisitionFrameRateLimitMode | Select what limits the acquisition frame rate | IEnumeration | RW |
| oeResultingFrameRate | Shows the resulting acquisition frame rate | IFloat | RO |

| | | | |
|-----------------------|---|--------------|----|
| oeMaxFrameRate | Shows the maximum acquisition frame rate reachable when the image compression is enabled | IFloat | RO |
| TriggerSelector | Selects the type of trigger to configure | IEnumeration | RW |
| TriggerMode | Controls if the selected trigger is active | IEnumeration | RW |
| TriggerSoftware | Generates an internal trigger | ICommand | RW |
| TriggerSource | Specifies the internal signal or physical input Line to use as the trigger source | IEnumeration | RW |
| TriggerOverlap | Specifies the type 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 | IEnumeration | RW |
| TriggerDelay | Specifies the delay in microseconds (us) to apply after the trigger reception before activating it | IFloat | RW |
| ExposureMode | Sets the operation mode of the Exposure | IEnumeration | RW |
| oeShortExposureEnable | Enable the short exposure mode | IBoolean | RW |
| oeDualExposureEnable | Enable the dual exposure mode | IBoolean | RW |
| ExposureTime | Sets the Exposure time when ExposureMode is Timed and ExposureAuto is Off | IFloat | RW |
| oeWaitTime1 | Time delay between the first and second exposure in Dual Exposure mode. | IFloat | RO |
| oeExposureTime2 | Second Exposure Time in Dual Exposure mode. | IFloat | RO |
| oeWaitTime2 | Time to wait before a new acquisition after the second exposure in Dual Exposure mode. | IFloat | RO |

| | | | |
|--------------------------|--|--------------|----|
| ExposureAuto | Sets the automatic exposure mode when ExposureMode is Timed | IEnumeration | RW |
| oeExposureAutoMin | Set the lower limit for the auto exposure algorithm | IFloat | RW |
| oeExposureAutoMax | Set the upper limit for the auto exposure algorithm | IFloat | RW |
| oeImageCompressionEnable | Enable the image compression algorithm | IBoolean | RW |
| oeFramesInBuffer | Displays the number of frames currently stored in the onboard memory | IInteger | RO |

Table 17: Acquisition Control Features

6.3.1 Trigger overlap

Exposure Time and Frame Readout relationship

By default, the feature **TriggerOverlap** is set to OFF: in this case, as shown in Fig.50, the following exposure time period is not allowed until the end of the current frame transfer, i.e. exposure time and frame transfer cannot be overlapped. In this configuration, however, latency between exposure time and frame transfer is highly repeatable.

In conclusion, a higher determinism can be achieved at the cost of a lower effective camera frame rate.

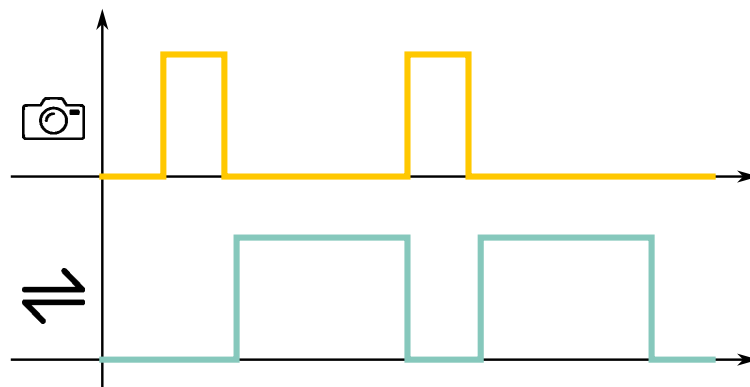


Figure 50: When TriggerOverlap is set to OFF, the latency is highly repeatable, but the following exposure time cannot be accepted until the current frame has been transferred to the internal memory. From top to bottom, the sensor exposure and readout signals are depicted.

When **TriggerOverlap** is set to Readout, the latency between exposure time and frame transfer is affected by a higher uncertainty, but the following exposure can happen during the actual transfer

between sensor and memory buffer. As shown in Fig.51, the only limitation in this mode is the overlapping of two following transfer periods.

In conclusion, a higher camera frame rate can be achieved (in accordance with the ethernet bandwidth limitation) at the cost of a lower frame transfer determinism.

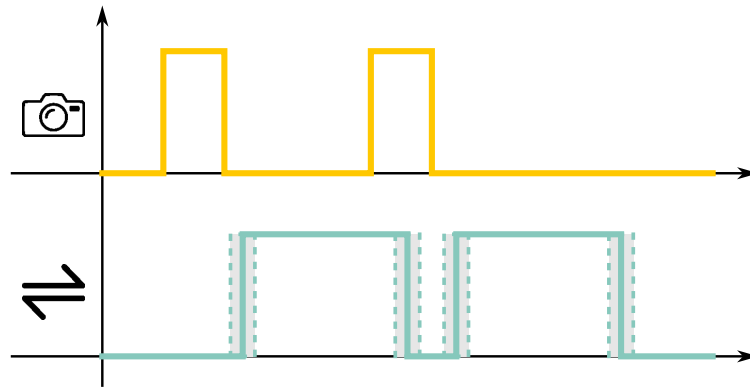


Figure 51: When TriggerOverlap is set to Readout, the following exposure time can be accepted when the current frame is being transferred to the internal memory, but the latency is affected by a higher uncertainty. From top to bottom, the sensor exposure and readout signals are depicted.

How Readout mode affects Exposure Time

ExposureTime value is bounded by:

- **ExposureTimeMin**: minimum exposure time that can be set by the user.
- **ExposureTimeMax**: maximum exposure time that can be set by the user.
- **ExposureTimeInc**: value of the discrete step for increasing/decreasing exposure time.

ExposureTimeMax is constant for both *TriggerOverlap* configurations (*Off* and *Readout*).

ExposureTimeMin differences between *TriggerOverlap* = *Off* and *TriggerOverlap* = *Readout* depend on sensor model, but are usually negligible.

TriggerOverlap mainly impacts on **ExposureTimeInc** value.

- When *TriggerOverlap* = *Off*, Exposure Time granularity is quite fine, in the order of tens of nanoseconds.
- When *TriggerOverlap* = *Readout*, Exposure Time granularity is strongly related to sensor line period, i.e. time needed by the image sensor to read a single line. In this configuration, the exposure time increment depends on sensor model, but usually is in the order of tens on microseconds.

Anyway, the **ExposureTime** set by the user is automatically adjusted by the device, depending on current camera configuration.

6.3.2 Dual Exposure

The Dual Exposure feature allows to acquire two frames as close as possible, making it easier inspecting fast moving objects or involving different sources of light. This can be achieved overlapping the first sensor readout with the second sensor exposure, as shown in Figure 52.

By default, the feature **oeDualExposureEnable** is not available. Dual Exposure is available only when **TriggerMode** is set to ON and **TriggerOverlap** is set to Readout. When **oeDualExposureEnable** is ON the **AcquisitionBurstFrameCount** disappears and it is automatically set to 2.

Dual Exposure is only available when a valid **TriggerSource** is set. The trigger starts the process explained below:

- The first exposure time *EXPOSURE TIME 1* follows **ExposureMode** settings: its duration can be Timed or TriggerWidth, as explained in subsection 6.3.4.
- Once the first exposure is completed a *WAIT TIME 1* is needed for sensor operation. This time represents the minimum achievable time span between two frames. This time is fixed.
- When it comes to *EXPOSURE TIME 2*, here takes place the trigger overlapping since the camera starts the second exposure during the first sensor readout. The *EXPOSURE TIME 2* lasts as long as the *SENSOR READOUT 1*. This time is fixed.
- After the second exposure the *WAIT TIME 2* occurs, allowing sensor operation and the sensor readout of the second frame. This time is fixed.

All fixed times are sensor dependent and are needed for the correct camera operation in dual exposure mode. These values depends on the sensor configuration (ROI, Pixel Format, Binning and Decimation) and can be read from **oeWaitTime1**, **oeExposureTime2** and **oeWaitTime2** features.

NOTE: bear in mind that a second hardware trigger can occur only after the entire process is completed, which means after the *WAIT TIME 2*. All triggering signals which occur during *WAIT TIME 1* or *EXPOSURE TIME 2* or *WAIT TIME 2* will be ignored.

NOTE: *WAIT TIME 1* should be considered when designing your machine vision system to properly select the camera model, depending on the application requirements.

NOTE: When **Dual Exposure** is enabled, **ChunkExposureTime** shows only the value set for *EXPOSURE TIME 1* (i.e. the value set in **Exposure Time** field).

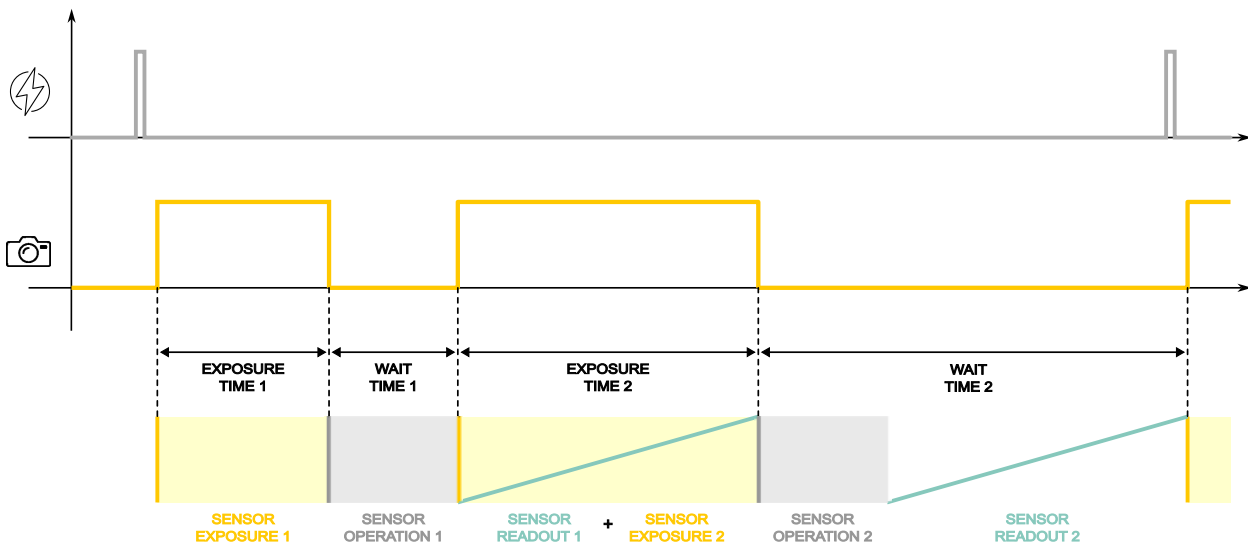


Figure 52: Dual Exposure timings and operation. From top to bottom, the trigger signal, the sensor exposure signal and the camera operation are depicted.

6.3.3 Trigger delay

Since synchronization between illumination and exposure can be challenging, the **TriggerDelay** feature can be used to properly align the exposure period of the sensor with the external illumination.

For example, consider the situation of a triggering device which enables both illumination and camera exposure. In case of bad alignment, not all the light is collected by sensor (see Figure 53), leading to an inefficient illumination.

The **TriggerDelay** feature can be used to shift the exposure time of Itala cameras in order to fully cover the illuminator on-time (see Figure 54).

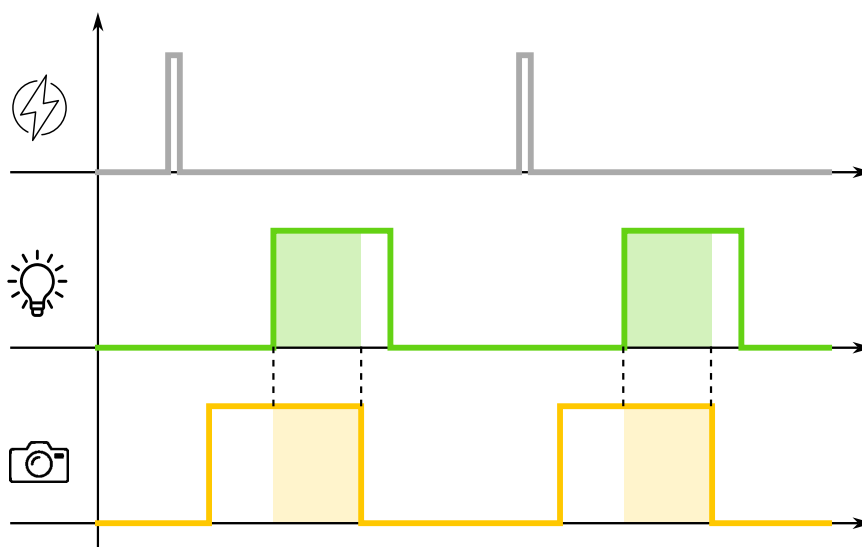


Figure 53: Incorrect alignment between illumination and camera exposure time. From top to bottom, the trigger signal, the illumination period and the camera exposure time are depicted.

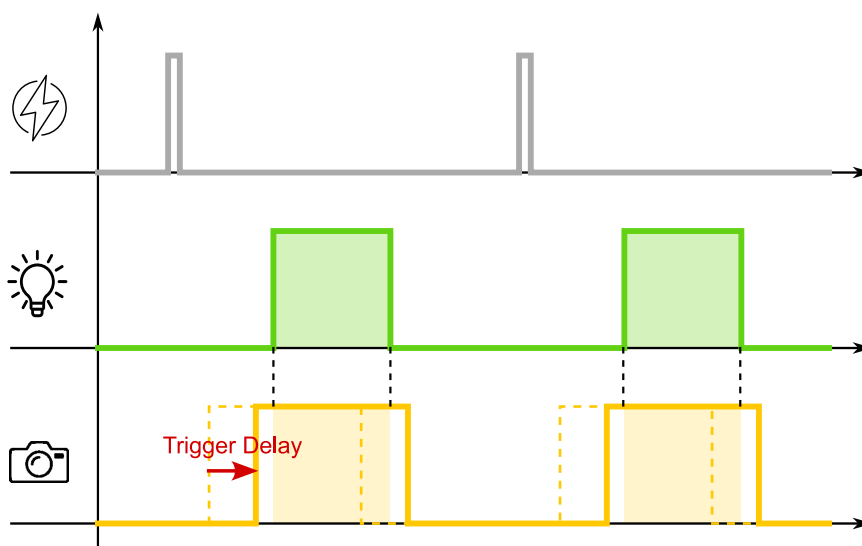


Figure 54: Correct alignment between illumination and camera exposure time. From top to bottom, the trigger signal, the illumination period and the camera exposure time are depicted.

6.3.4 Timed vs TriggerWidth Exposure Mode

Exposure mode can be either **Timed** or **TriggerWidth**.

When **Timed Exposure** is selected, the sensor exposure time is set using the *ExposureTime* or *ExposureAuto* features.

In this case the exposure time can be expressed as follows:

$$SensorExposureTime = ExposureTime \quad (1)$$

For example, if $ExposureTime = 500\mu s$, the image sensor is exposed for $500\mu s$.

When **TriggerWidth Exposure** is selected, the exposure duration is equal to the width of the current trigger signal pulse.

Actually, for some image sensors, the real exposure time can be computed as follows:

$$SensorExposureTime = TriggerPulse + ExposureOffset \quad (2)$$

where:

- *SensorExposureTime* is the overall exposure time of the image sensor.
- *TriggerPulse* is equal to the external trigger signal.
- *ExposureOffset* is an additional intrinsic period in which the image sensor is collecting light.

In this case, if the trigger pulse supplied to the camera is equal to $500\mu s$, also the intrinsic contribution of the sensor exposure offset must be taken into account, leading to an overall exposure time higher than $500\mu s$.

In general, the exposure offset depends on the image sensor and it's in the order of units/tens of microseconds.

NOTE: When **TriggerWidth Exposure** is selected, the **ExposureTime** field acts as the maximum allowed pulse width. Providing a trigger signal with a larger pulse width results in a captured frame with an exposure time clamped to the **ExposureTime** field.

NOTE: When **TriggerWidth Exposure** is selected, **ChunkExposureTime** shows the value set in **ExposureTime** field.

6.3.5 Image Compression

The **image compression** function allows the captured frame to be compressed with a lossless algorithm.

The compression ratio of the algorithm doesn't have a constant value, but is variable and depends on the image captured. In particular, the ratio is inversely proportional to the entropy of the pixels that make up the image. Therefore, the compression ratio is affected by image noise, which

is why it is recommended to use low gain levels. The typical compression ratio is between 1.5 and 2.

If the **image compression** is enabled, the image payload size can be reduced and thus the frame rate can be increased for the same bandwidth. This is because the image payload size (along with exposure time, acquisition mode and other image processing functions such as binning and decimation) controls the frame rate of the camera, assuming the link bandwidth is fixed (see DeviceLinkThroughputLimit function in the section 6.1.1).

To enable the **image compression**, set the **oeImageCompressionEnable** parameter. When compression is activated, the camera automatically compresses the frame, calculates the new image payload size and adjusts the frame rate with the new size value to occupy all available bandwidth (DeviceLinkThroughputLimit). The result is that the algorithm increases the speed of the camera as much as possible.

All these operations are performed automatically by the camera only with compression enabled; all that is left for the user to do is to configure the other camera parameters.

The maximum frame rate, achievable by the camera only when image compression is on, is shown by **oeMaxFrameRate** parameter, while the absolute maximum frame rate is defined by the speed of the image sensor.

NOTE: The **image compression** supports only pixel formats with a size of 8bit: Mono8, BayerRG8, BayerGR8, BayerGB8, BayerBG8, Polarized00Mono8, Polarized00BayerRG8, Polarized00BayerGR8, Polarized00BayerGB8 and Polarized00BayerBG8.

NOTE: Chunk data are not supported if **image compression** is enabled.

The **decompression algorithm** is implemented in the Opto Engineering® GenTL producer (.cti), so decompression is not available with third party software that does not rely on Opto Engineering® GenTL producer (.cti).

The Fig.55 shows an example of implementation: two cameras share a bandwidth of 1 Gbps and each camera has 0.5 Gbps of bandwidth available (DeviceLinkThroughputLimit = 62500000). If compression can achieve a speed of x2, the two cameras can achieve the same frame rate even though they have half the bandwidth.

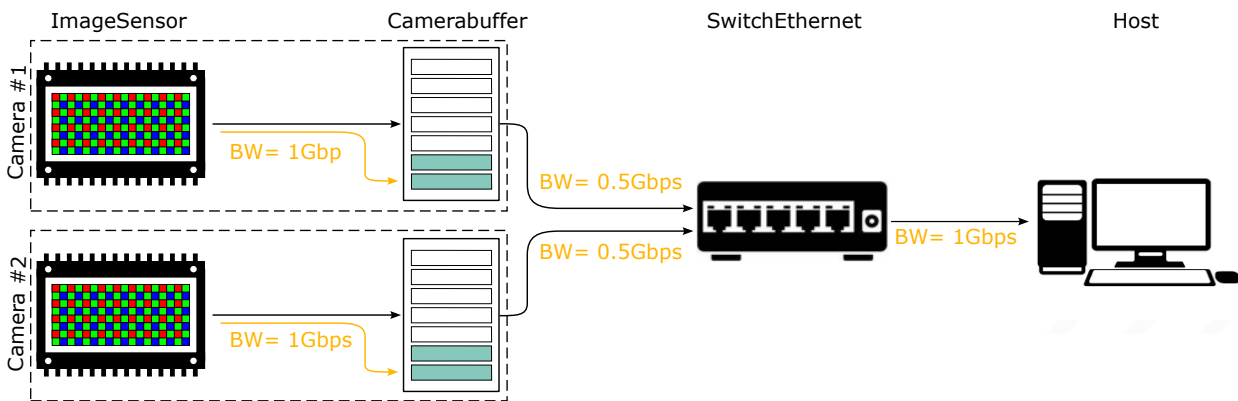


Figure 55: Example of multi-camera system with image compression enabled.

6.4 Analog Control

This sections describes how to influence the analog features of an image, such as gain, black level and gamma.

| Feature | Description | Interface | Access |
|----------------------|---|--------------|--------|
| Gain | Controls the selected gain as an absolute physical value | IFloat | RW |
| GainAuto | Sets the automatic gain control (AGC) mode | IEnumeration | RW |
| oeGainAutoMin | Set the lower limit for the auto gain algorithm | IFloat | RW |
| oeGainAutoMax | Set the upper limit for the auto gain algorithm | IFloat | RW |
| BlackLevel | Controls the analog black level as an absolute physical value | IFloat | RW |
| BalanceRatioSelector | Selects the balance ratio to control | IEnumeration | RW |
| BalanceRatio | Controls the ratio of the selected color component | IFloat | RW |
| BalanceWhiteAuto | Controls the mode for automatic white balancing between the color channels. The white balancing ratios are automatically adjusted | IEnumeration | RW |
| oeGammaEnable | Enable the gamma correction. The LUT functions will be disabled | IBoolean | RW |

| | | | |
|-------|--|--------|----|
| Gamma | Controls the gamma correction of pixel intensity | IFloat | RW |
|-------|--|--------|----|

Table 18: Analog Control Features

6.4.1 Gain

Gain is a multiplying factor applied to pixel values in order to increase the image brightness also in low-light conditions.

Sensor gain, however, affects indiscriminately useful signal and undesired noise: as can be seen in Fig.56, image brightness increases proportionally to the gain, however image quality can decrease in case of high gain values due to excessive noise.



Figure 56: Different gain images

6.4.2 White balance

The **white balance** feature allows the adjustment of the response of the three color channels (R, G, B) of color cameras.

Typically color sensors have different sensitivities for the three color coordinates: this is mainly due to the different response of the Bayer filter present on top of the image sensor.

In Fig.57 (left figure) is shown a typical characteristic of a color sensor: even in case of perfectly uniform external light (i.e. flat spectrum), the pixel responses are not uniform (different mean values of grey levels), therefore green pixels will result brighter than red and blue ones.

To solve this inhomogeneity a scaling factor can be applied to the three color channels:

$$R_{out} = K_{red} * R_{in} \quad (3)$$

$$G_{out} = K_{green} * G_{in} \quad (4)$$

$$B_{out} = K_{blue} * B_{in} \quad (5)$$

To further simplify this operation a color channel may be kept constant (typically the green one, since it's the dominant Bayer tile color). The white balance coefficient for the red and blue channels can be therefore written as:

$$R_{out} = K_{red} * R_{in} \quad (6)$$

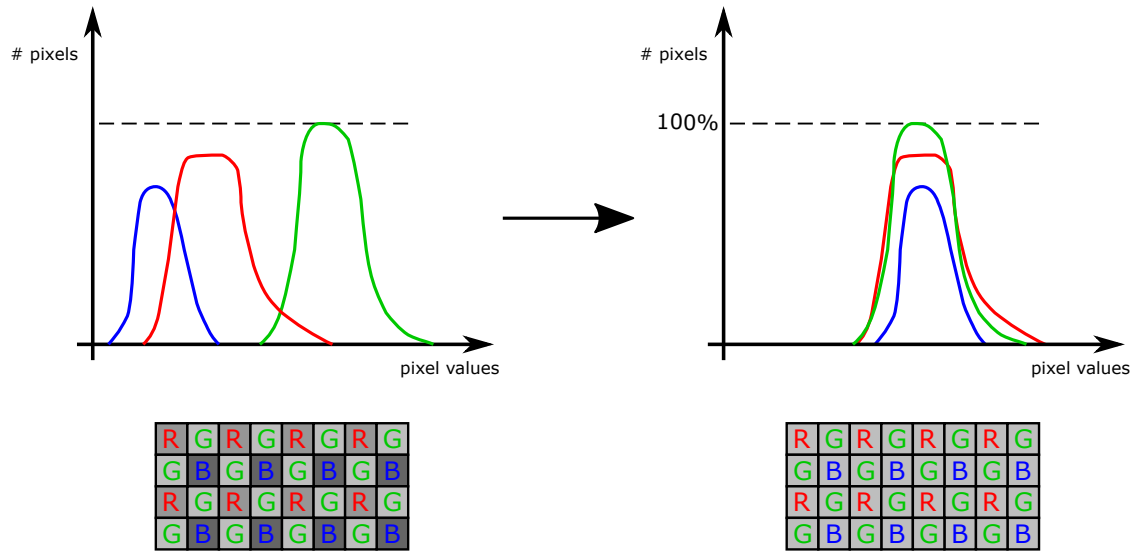


Figure 57: On the left, the histogram of a typical spectral sensitivity of a color sensor. On the right the histogram of a white balanced camera.

$$G_{out} = G_{in} \quad (7)$$

$$B_{out} = K_{blue} * B_{in} \quad (8)$$

where:

$$K_{red} = G_{in}/R_{in} \quad (9)$$

$$K_{blue} = G_{in}/B_{in} \quad (10)$$

Like in the last equations, **BalanceRatio** allows to set K_{red} and K_{blue} coefficients while K_{green} is fixed to 1.

The effect of the white balance procedure is depicted in Fig.57 (right figure): the three channels are equalized and show the same mean grey level.

Itala cameras offer the possibility to automatically balance the three color coordinates: in order to do this, the **BalanceWhiteAuto** feature must be enabled.

The BalanceWhiteAuto algorithm relies on the grey world approximation: the premise behind this assumption is that in a well balanced color image the average of all the color present is a neutral grey.

Consequently, in order to obtain a perfect white balance, perform the following steps:

- Start a free-run acquisition;
- Make sure to insert a uniform sample (for example a white neutral background) which cover all the ROI (region of interest) of the image;
- Enable the BalanceWhiteAuto (*Continuous mode* or *Once mode*);
- In case of *Continuous mode* auto white balance, after the correction is performed, disable the BalanceWhiteAuto;

- Remove the uniform neutral background;
- The camera is now balanced and ready to be used.

An example of auto white balance procedure is shown in Fig.58: on the left side an uncorrected image is displayed; on the right side a white balance corrected picture is shown.

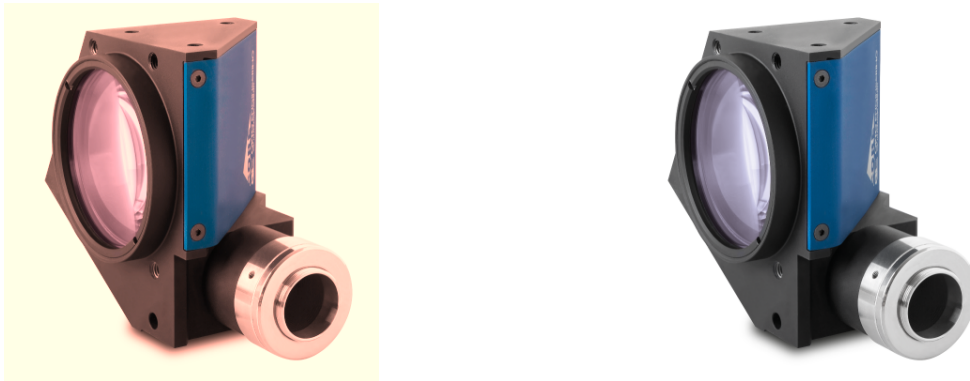


Figure 58: On the left side an uncorrected image is displayed; on the right side a white balance corrected picture is shown.

6.4.3 Gamma correction

Gamma correction is a non-linear operation which follows the formula 11:

$$V_{out} = V_{in}^{\gamma} \quad (11)$$

where V_{out} is the grey level of the pixel n after the gamma correction, V_{in} is the grey level of the pixel n and γ is the coefficient used for the non-linear transformation, set with the **Gamma** feature.

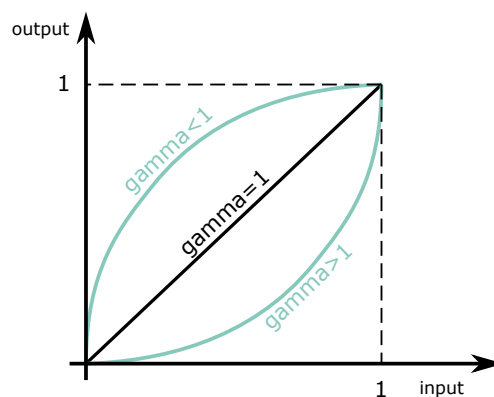


Figure 59: Gamma correction curves for $\gamma = 1$, $\gamma < 1$ and $\gamma > 1$. X-axis and Y-axis are normalized.

This operation is clearly shown in Fig.59: while "black" and "white" pixels remain the same after the correction, the different grey pixels are re-mapped on a non-linear curve, which enhances dark or bright features, depending on γ value.

In Fig.60 is shown an example of application of gamma correction.

A $\gamma < 1$ expands the range of values of dark regions and compresses the bright ones, so it's useful when looking at features in dark parts of the image. Viceversa, a $\gamma > 1$ compresses the range of values of the dark regions and expands the bright ones, so it's useful when looking at features in bright parts of the image.

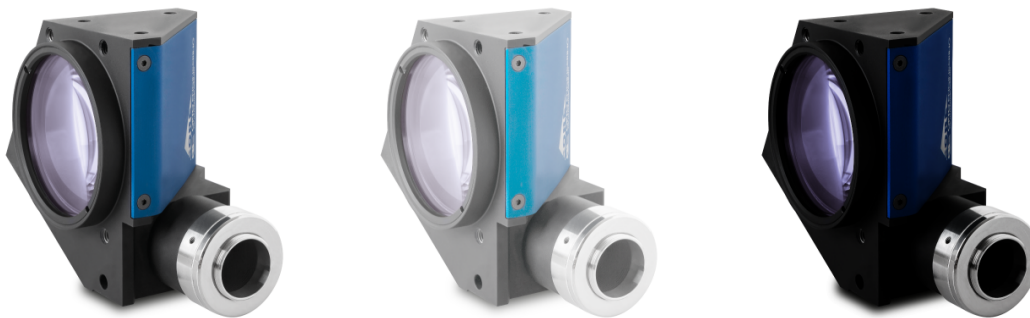


Figure 60: Images taken with different γ values: on the left $\gamma = 1$, in the center $\gamma < 1$ and on the right $\gamma > 1$

Please note that the Gamma correction can't be used if the LUT feature is enabled (see section 6.6.1).

6.4.4 Black level

The **BlackLevel** is an offset value, expressed in grey levels, that can be added to all the pixels of the image.

The effect of adding a black level value to an image is to move the pixels histogram towards the saturation level (as depicted in Fig.61).

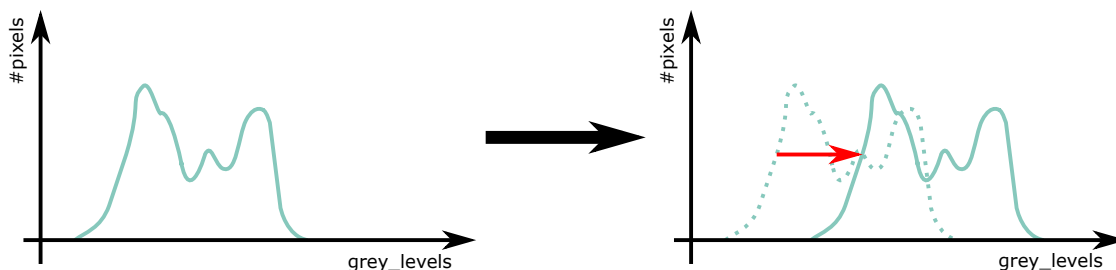


Figure 61: The black level moves the pixel histogram towards higher pixel values.

6.5 OE Auto Functions Control

This sections includes all the features related to auto exposure and auto gain control.

| Feature | Description | Interface | Access |
|------------------------|--|-----------|--------|
| oeAutoTargetBrightness | Desired brightness level (in %) of the image used by auto gain and auto exposure functions | Integer | RW |
| oeResultingBrightness | Actual brightness level (in %) of the image | Integer | R |
| oeAutoDampingFactor | Control value (in %) used by auto gain and auto exposure features to reduce algorithm oscillations | Integer | RW |
| oeAutoConfidence | Hysteresis around the target value used by auto gain and auto exposure features. Larger values improve image stability but increase the brightness error | Integer | RW |
| oeAutoAOIWidth | Width of the area used for auto functions calculations (in pixels) | Integer | RW |
| oeAutoAOIHeight | Height of the area used for auto functions calculations (in pixels) | Integer | RW |
| oeAutoAOIOffsetX | Horizontal offset from the origin to the area used for auto functions calculations (in pixels) | Integer | RW |
| oeAutoAOIOffsetY | Vertical offset from the origin to the area used for auto functions calculations (in pixels) | Integer | RW |

Table 19: OE Auto Functions Control Features

6.5.1 OE Autoexposure/Autogain

When the correct amount of exposure time needed to have a sufficient level of brightness is not known a priori, or when the illumination of the target changes during time, the **autoexposure** and **autogain** features can be used to obtain a stable brightness level even if the external light conditions are not constant.

For example, consider Fig.62: at the beginning of the acquisition (i.e. the first capture), the resulting average grey level value of the image is equal to 50. Usually, a good exposure is centered at half of the full scale range (about 127 in case of 8-bit image), so an average grey level value of 50% of the full dynamic can be set in the **oeAutoTargetBrightness** feature. Thus, as it can be seen in Fig.62, the average grey level value automatically adjusts in order to achieve the desired brightness of 127.

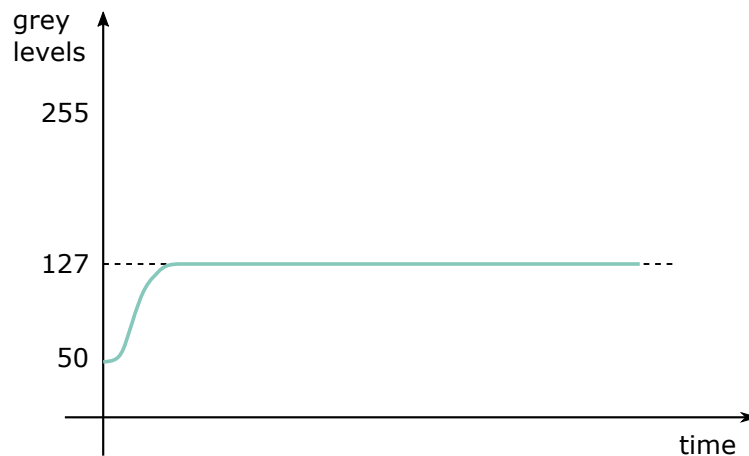


Figure 62: Evolution of the average grey level value over time when autoexposure is active.

In order to avoid the continuous adjustment of the exposure time, a certain threshold can be set with the **oeAutoConfidence** feature: in this way the algorithm becomes more insensitive to little external light fluctuations and becomes active only in case of a consistent grey level variations (see Fig.63).

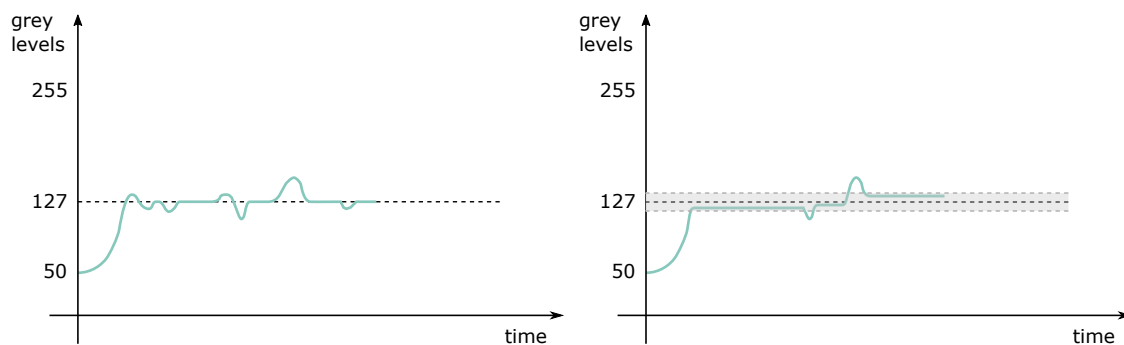


Figure 63: The continuous adjustment of the exposure time due to variations of the external light (figure on the left) can be slowed down by adding a confidence value around the target brightness (figure on the right).

The behaviour of the autogain/autoexposure algorithm can be tuned through the **oeAutoDampingFactor** node: low values of this parameter give higher stability but slower response; on the contrary, high values can speed up the algorithm but can lead to unstable behaviours (see Fig.64). If the light conditions are poor, a long exposure time is needed to achieve the desired brightness level: in some cases, this situation can lead to an undesired reduction of the camera frame rate. In order to avoid this condition, it's possible to set a minimum and maximum exposure time which can clamp and limit the exposure time computed by the autoexposure algorithm (i.e. **oeExposureAutoMin** and **oeExposureAutoMax** respectively): for example, as shown in Fig.65, in case the exposure time needed for having the desired brightness is greater than the **oeExposureAutoMax** value, the target grey level cannot be reached but the resulting frame rate won't be affected

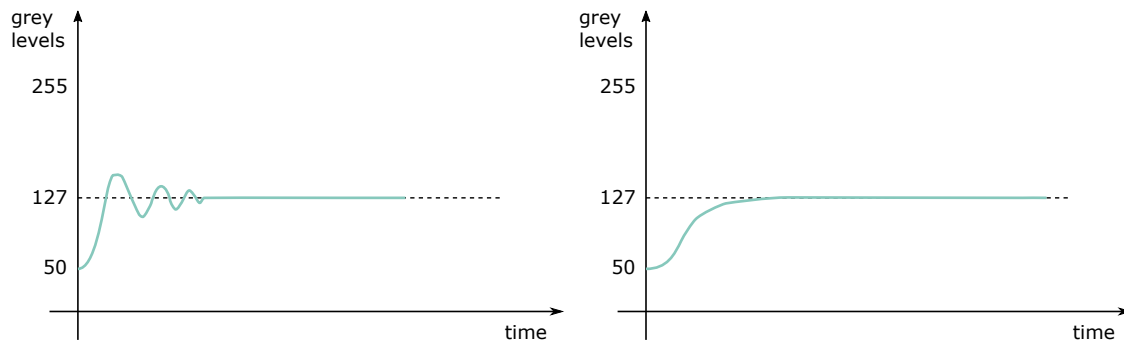


Figure 64: In case of low damping factor (figure on the left) the algorithm has a fast response but oscillations may rise; in case of high damping factor (figure on the right) the algorithm is stable but it can require long time to converge.

by an excessively long exposure time.

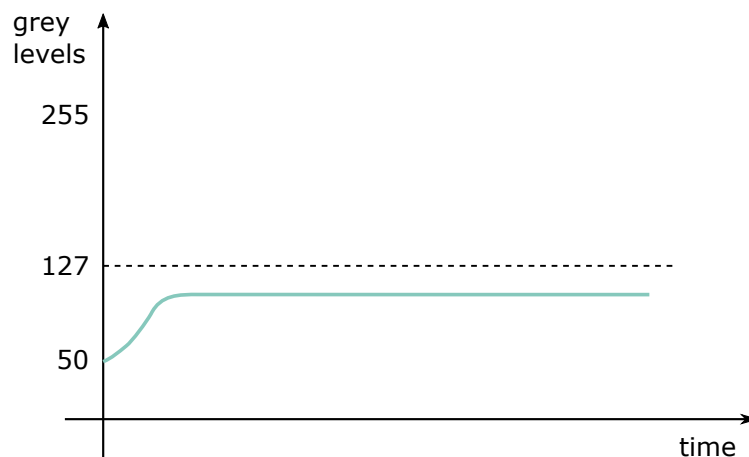


Figure 65: When an `oeExposureAutoMax` is set, the target brightness may not be achieved, but long exposures are avoided, thus preventing the reduction of the camera frame rate.

6.6 LUT Control

Features in this chapter describe the Look-up table (LUT) related features.

| Feature | Description | Interface | Access |
|-------------|------------------------------|--------------|--------|
| LUTSelector | Selects which LUT to control | IEnumeration | RW |
| LUTEnable | Activates the selected LUT | IBoolean | RW |

| | | | |
|----------|---|---------|----|
| LUTIndex | Control the index (offset) of the coefficient to access in the selected LUT | Integer | RW |
| LUTValue | Returns the Value at entry LUTIndex of the LUT selected by LUTSelector | Integer | RW |

Table 20: LUT Control Features

6.6.1 LUT

The **LUT** (Look-up-table) feature allows the user to set a transformation at pixel level: a specific grey level at the input of the LUT can be replaced by a new grey level value. All the pixels with the same grey level value are processed in the same way.

Consider the graphs shown in Fig.66: in the first graph no LUT is applied, so the output grey level is equal to the input one (e.g. grey level 127 remains 127 at the output of the LUT); in the second graph, a binary thresholding is applied: all pixels with grey level values below 127 (in case of 8-bit image) are set equal to 0 (black), the others are set to 255 (white).

In Fig.67 are shown the results of the two previous transformations.

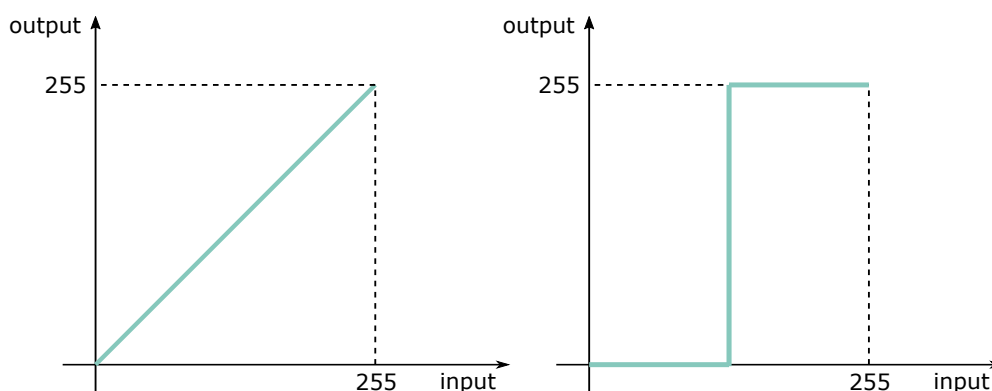


Figure 66: Two typical in-to-out transfer function: on the left no LUT is applied, on the right a binary thresholding is adopted.

Please note that the LUT can't be used if the Gamma feature is enabled (see section 6.4.3). For more information about the LUT wizard of Itala View refer to section 4.7.4.

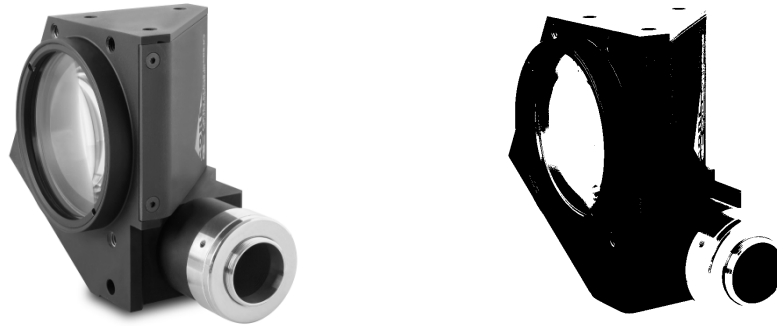


Figure 67: On the left no LUT is applied, on the right a binary thresholding is adopted.

6.7 Color transformation control

The Color Transformation section describes all features related to color transformations in the device.

| Feature | Description | Interface | Access |
|----------------------------------|--|--------------|--------|
| ColorTransformationSelector | Selects which Color Transformation module is controlled by the various Color Transformation features | IEnumeration | RW |
| ColorTransformationEnable | Activates the selected Color Transformation module | IBoolean | RW |
| ColorTransformationValueSelector | Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module | IEnumeration | RW |
| ColorTransformationValue | Represents the value of the selected Gain factor or Offset inside the Transformation matrix | IFloat | RW |

Table 21: Color Transformation Control Features

6.7.1 Color Correction Matrix (CCM)

Obtaining a good color fidelity can be challenging, this because the colors of an image depend on the camera color filter and, above all, on illumination.

Since illumination is application specific, sometimes colors need to be corrected in order to obtain a suitable color fidelity.

The **Color Correction Matrix (CCM)** allows the adjustment of the output colors of an image by

acting on gains/offset as follow:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} Offset_0 \\ Offset_1 \\ Offset_2 \end{bmatrix}$$

where R' , G' and B' are the corrected color coordinates, while R , G and B are the uncorrected ones. Gains and Offsets can be freely edited by the user but, in order to obtain an excellent calibration, a wizard has been already developed and available in Itala View. Please refer to Paragraph 4.7.6 to see the steps of this calibration procedure.

The color correction matrix is also use to make conversions between color spaces: for example, if a YUV pixel format is selected, the camera automatically load the right coefficients to switch from RGB to YUV color space:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

For the proper adjustment of the CCM coefficients refer to section 4.7.6.

6.7.2 How to perform a correct color calibration

In order to perform a correct color calibration with Itala cameras, Itala View can be used and the following steps can be performed:

1. Given a proper lighting, select the *Horizontal line profile* tab (central lower panel) and draw a ROI which includes only the grayscale values on the bottom of the color checker (Figure 68).
2. Each greyscale tile of the displayed image should match the reference value imposed by the color checker.
Therefore it's necessary to adjust *ExposureTime* and *Gamma* values in order to achieve this perfect match (Figure 69). For now, consider only the green channel (current pixel values are displayed at the bottom right of the image display panel).
3. After the green channel has been properly adjusted, use the *BalanceRatioSelector* and the *BalanceRatio* features to do the same operation for the red and blue color channels.
Avoid the *Balance white auto* feature for this step and keep monitored the *Horizontal line profile* tab. When the R, G and B curves are superimposed, the white balance is optimal (Figure 70).
4. Now the color calibration can be performed by using the dedicated wizard in the *Tool* panel of Itala View: the instructions for the proper use of the *Color Correction Wizard* are listed in Paragraph 4.7.6 (Figure 71).
Once the calibration is complete, the result will be similar to the following one depicted in Figure 72.

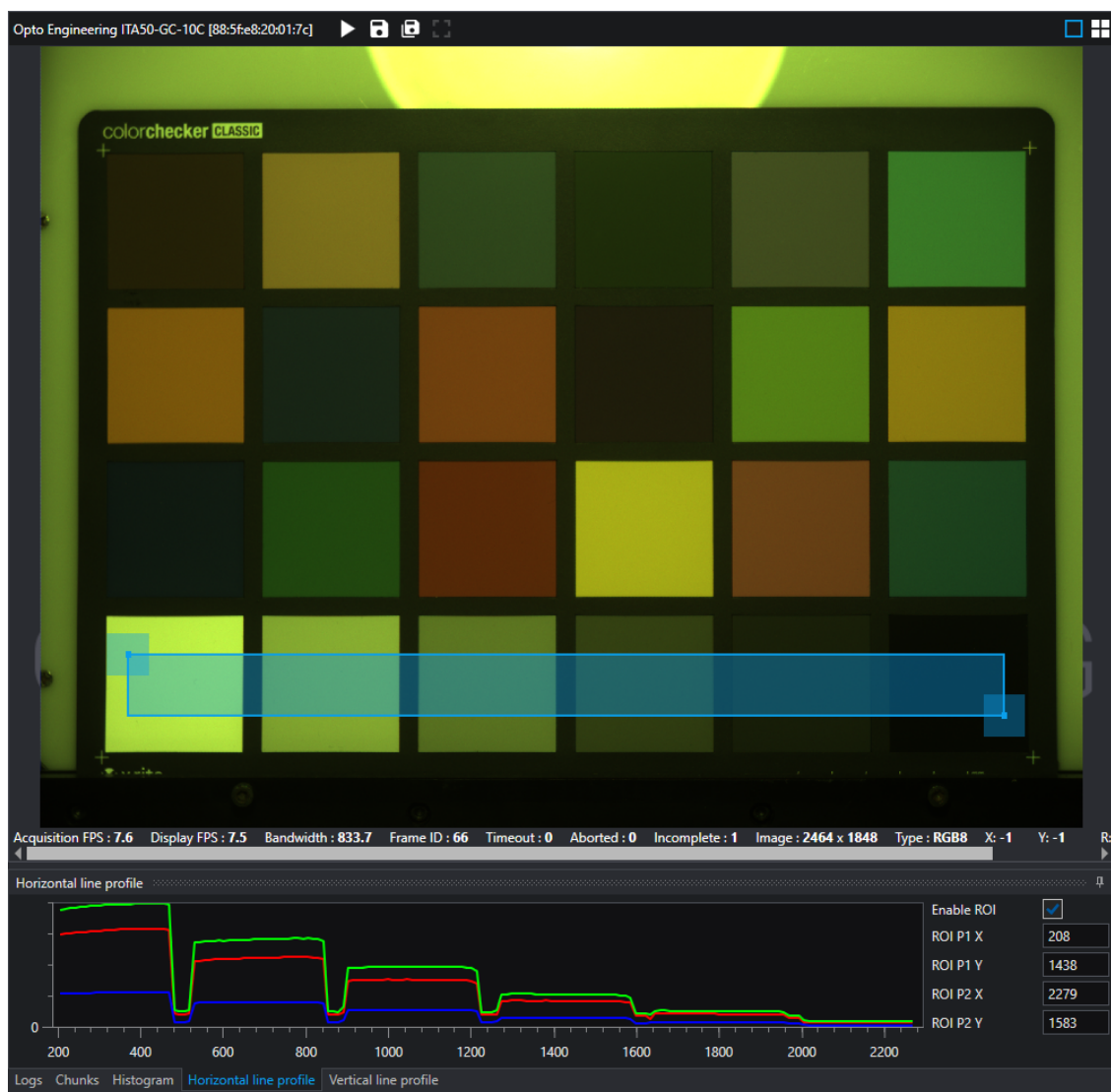


Figure 68: First steps of the color calibration procedure: setting of a proper ROI (including only the greyscale tiles of the color checker) and displaying the results on the horizontal line profiler.

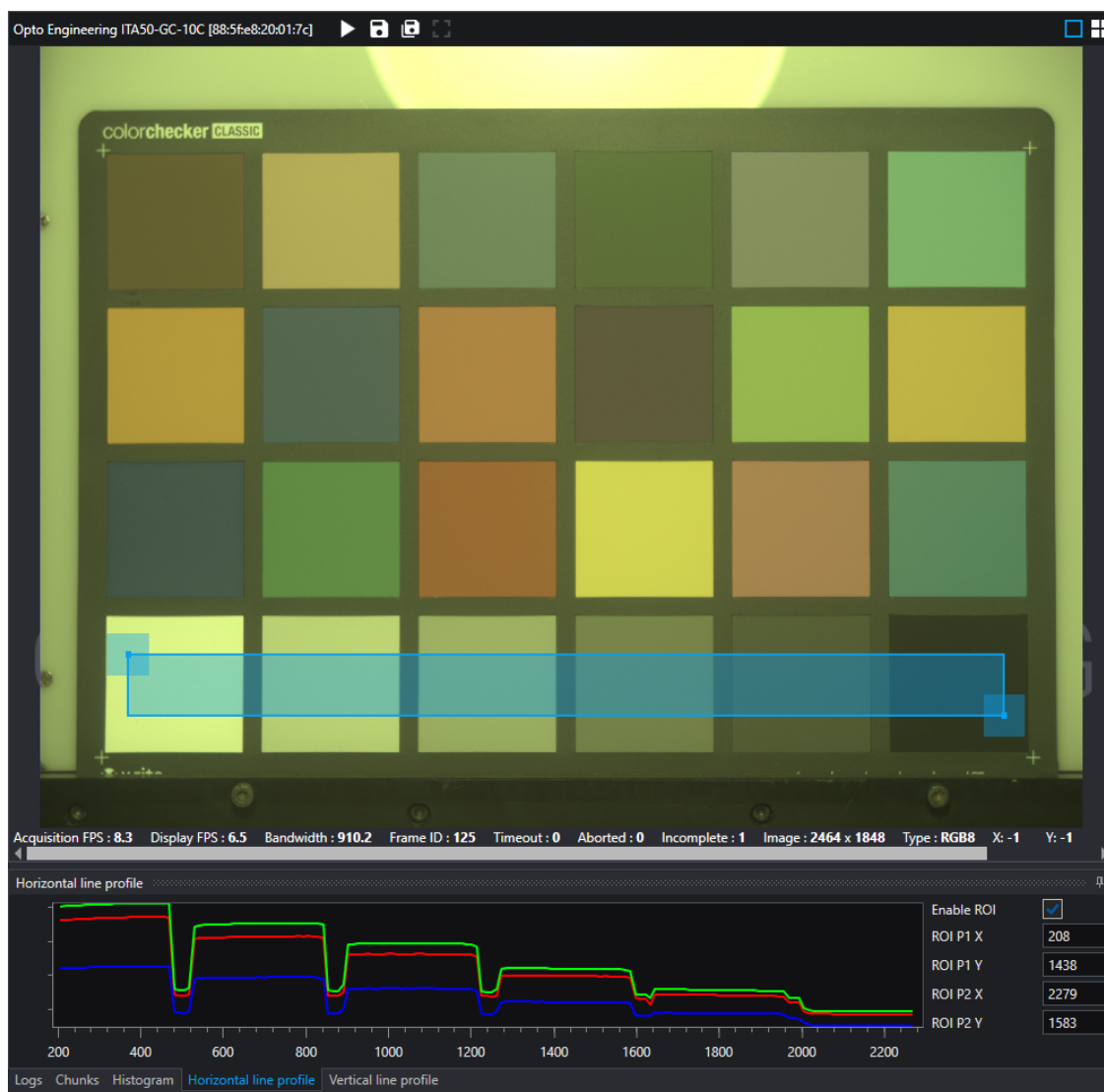


Figure 69: Second step of the color calibration procedure: *Exposure Time* and *Gamma* adjustment so that the green channel matches the value imposed by the color checker.

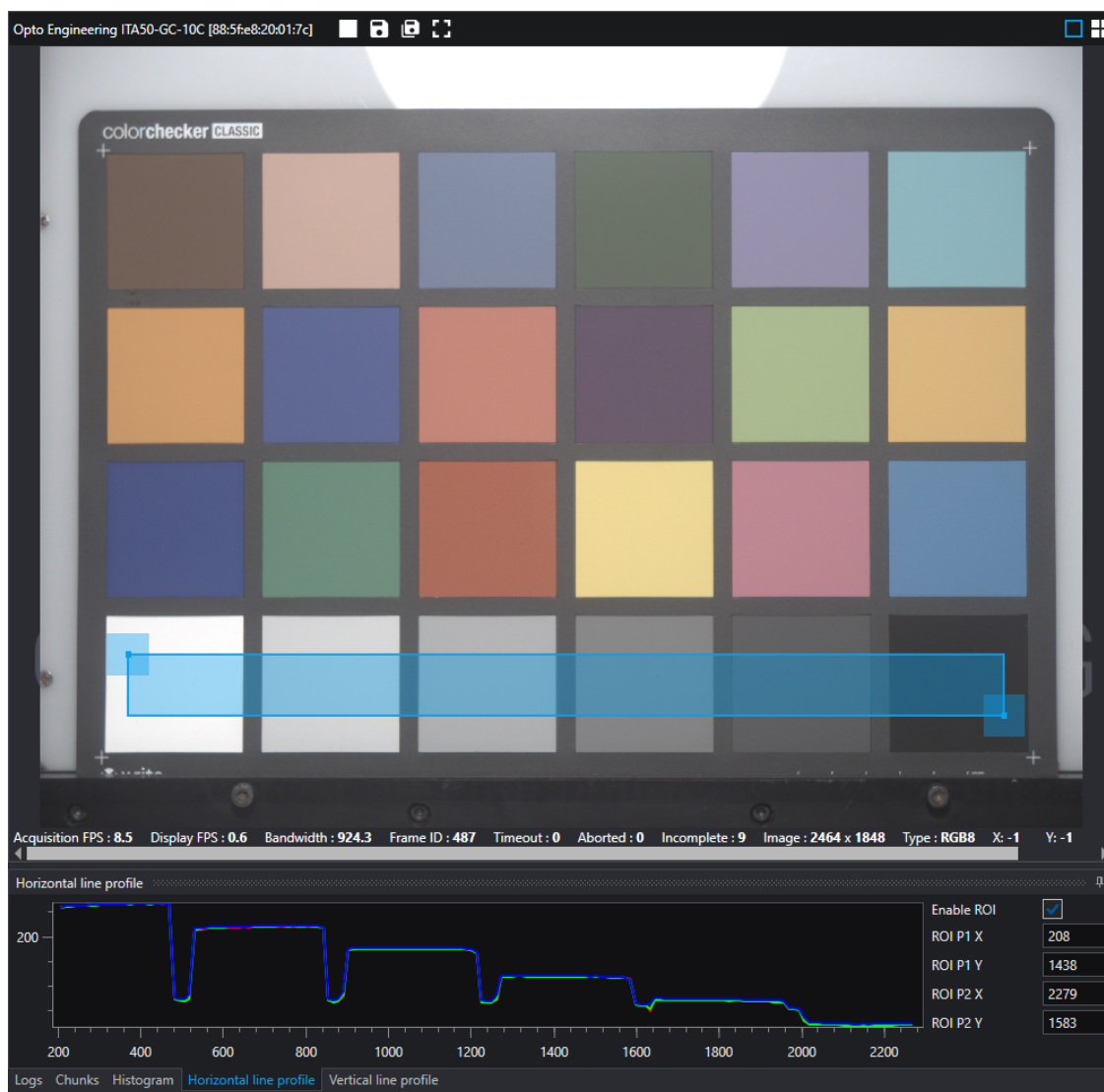


Figure 70: Third step of the color calibration procedure: using the *BalanceRatio* feature, adjust the red and blue channels in order to have all the three color curves superimposed in the horizontal line profile tab.

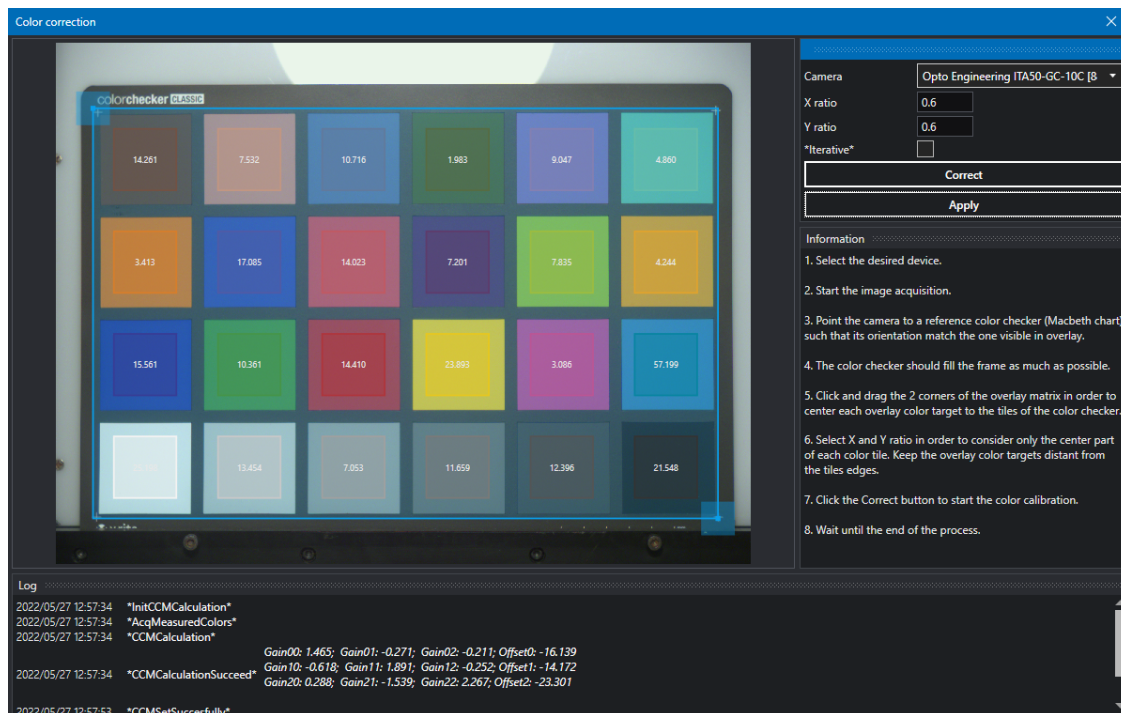


Figure 71: Fourth step of the color calibration procedure: using the *Color Correction Wizard* of Itala View, perform the color calibration following the tips of the wizard.

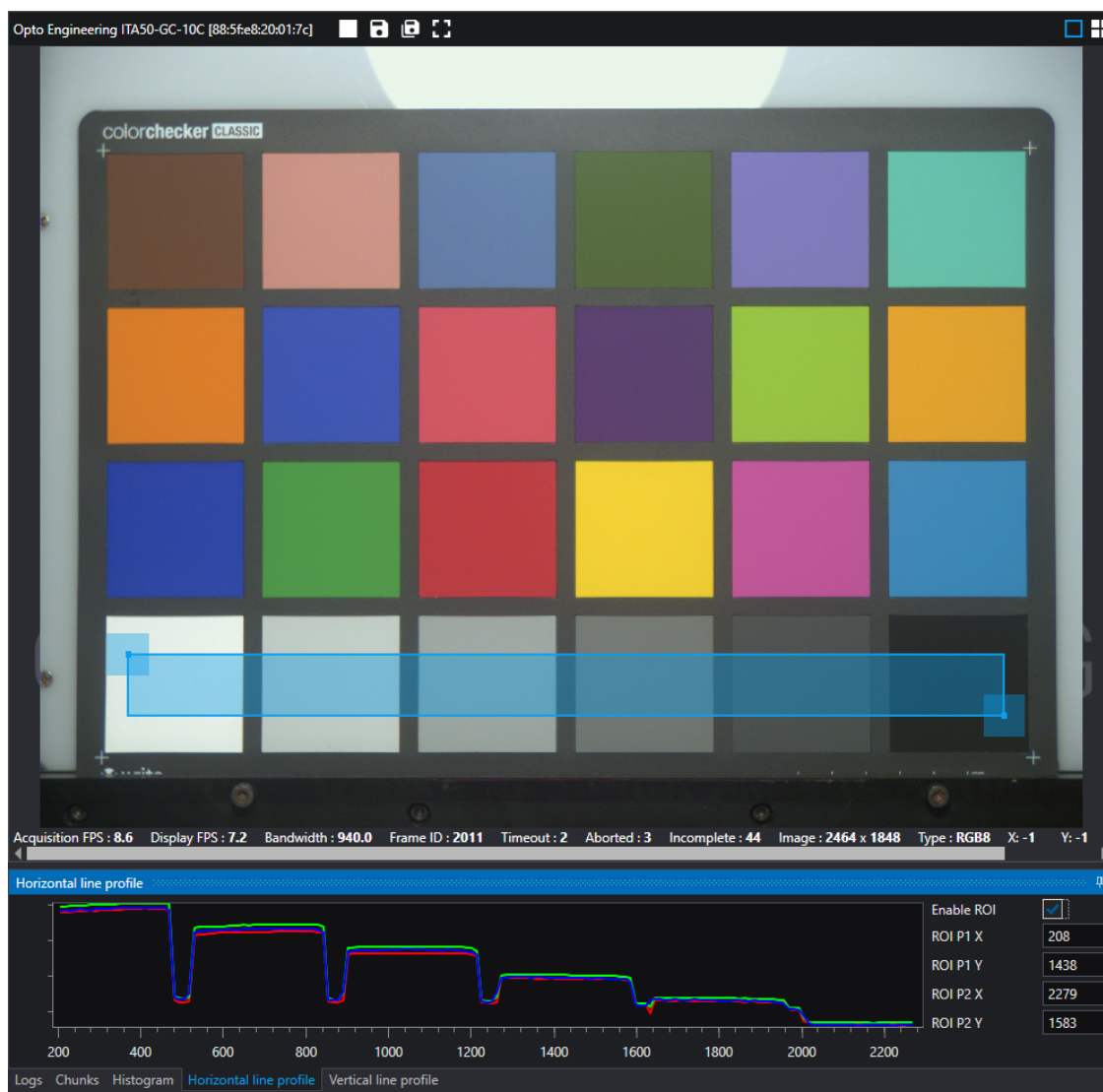


Figure 72: Color checker after Color Calibration procedure.

6.8 Digital I/O Control

The Digital I/O chapter covers the features required to control the general Input and Output signals of the device. These include input and output control signals for triggering timers, counters and also static signals such as user configurable input or output bits.

| Feature | Description | Interface | Access |
|------------------------|---|--------------|--------|
| LineSelector | Selects the physical line (or pin) of the external device connector or the virtual line of the Transport Layer to configure | IEnumeration | RW |
| LineMode | Controls if the physical Line is used to Input or Output a signal | IEnumeration | RW |
| LineInverter | Controls the inversion of the signal of the selected input or output Line | IBoolean | RW |
| LineStatus | Returns the current status of the selected input or output Line | IBoolean | R |
| LineStatusAll | Returns the current status of all available Line signals at time of polling in a single bit field | IInteger | R |
| LineSource | Selects which internal acquisition or I/O source signal to output on the selected Line | IEnumeration | RW |
| oeDebounceEnable | Enable the input debounce circuitry. This allows to filter the input signal and ignore spurious commutations. | IBoolean | RW |
| oeDebounceAmount | Amount of time for which the input signal need to stay constant in order to be recognized as a valid input | IFloat | RW |
| oePulseGeneratorEnable | Override the output signal with a pulse generated at the rising edge of the signal specified by LineSource | IBoolean | RW |
| oePulseGeneratorPeriod | Sets the duration of the output signal pulse | IFloat | RW |
| UserOutputSelector | Selects which bit of the User Output register will be set by UserOutputValue | IEnumeration | RW |

| | | | |
|------------------------|---|----------|----|
| UserOutputValue | Sets the value of the bit selected by UserOutputSelector | IBoolean | RW |
| UserOutputValueAll | Sets the value of all the bits of the User Output register | Integer | RW |
| UserOutputValueAllMask | Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register | Integer | RW |

Table 22: Digital I/O Control Features

6.8.1 I/O stage

The digital I/O block includes an onboard processing stage for input trigger signals and synchronization outputs.

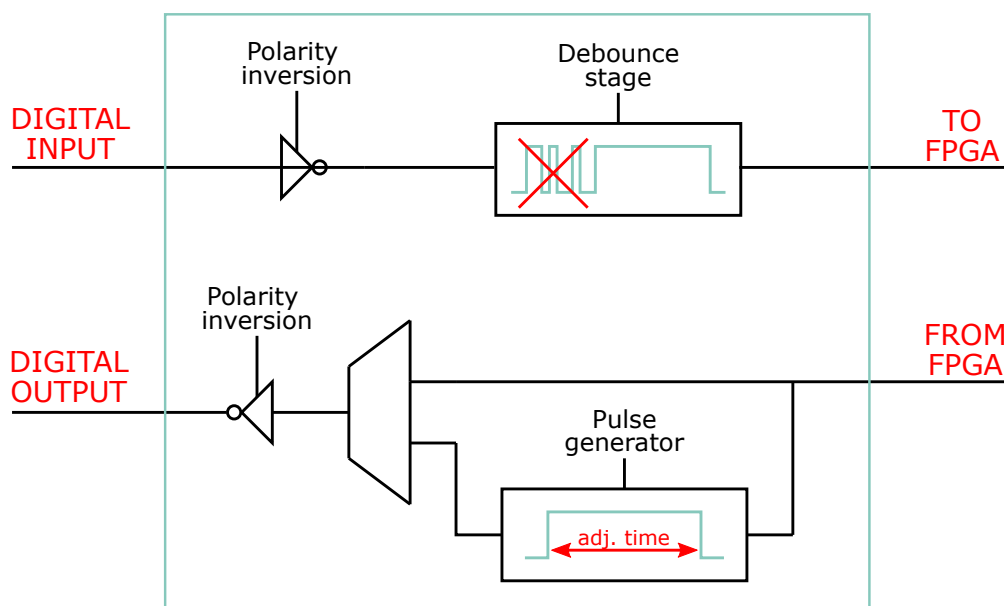


Figure 73: Digital I/O stage representation.

The **LineInverter** feature must be activated when the input trigger works with an *active-low* logic, i.e. when a falling edge must be detected. This feature applies also for synchronization outputs.

External triggers may be characterized by spurious and unwanted pulses or spikes known as *bounces*. For this reason a debounce stage has been included in the processing chain.

The **oeDebounceAmount** feature sets a threshold level: signals whose duration is lower than oeDebounceAmount are considered spurious pulses and therefore are discarded, while the other ones are considered useful signal.

Obviously, the debounce stage brings intrinsically a delay in the input trigger processing, since a

period of **oeDebounceAmount** must pass before choosing if the input signal is spurious or not.

The output synchronization signal can be chosen between an internally generated pulse which asserts when conditions are met (e.g. **TimerEnd**, **CounterEnd**, **EncoderOut**, ...) and a dedicated pulse generator: the advantage of using the second approach relies in the possibility of choosing the duration of the pulse, while in the first case the generated signals may last only one clock period and may be too fast to be detected by the slave device.

The **oePulseGeneratorEnable** feature allows the triggering of the pulse generator when **Line-Source** condition is met, while the **oePulseGeneratorPeriod** feature sets its on-time.

6.9 Counter and Timer Control

This section lists all features related to control and monitor Counters and Timers.

| Feature | Description | Interface | Access |
|--------------------|---|--------------|--------|
| CounterSelector | Selects which Counter to configure | IEnumeration | RW |
| CounterEventSource | Select the events that will be the source to increment the Counter | IEnumeration | RW |
| CounterResetSource | Selects the signals that will be the source to reset the Counter | IEnumeration | RW |
| CounterDuration | Sets the duration (or number of events) before the CounterEnd event is generated | Integer | RW |
| CounterValue | Reads or writes the current value of the selected Counter | Integer | RW |
| CounterReset | Does a software reset of the selected Counter and starts it | ICommand | RW |
| TimerSelector | Selects which Timer to configure | IEnumeration | RW |
| TimerTriggerSource | Selects the source of the trigger to start the Timer | IEnumeration | RW |
| TimerDuration | Sets the duration (in microseconds) of the Timer pulse | IFloat | RW |
| TimerDelay | Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer | IFloat | RW |
| TimerValue | Reads or writes the current value (in microseconds) of the selected Timer | IFloat | RW |

| | | | |
|------------|---|----------|----|
| TimerReset | Does a software reset of the selected timer and starts it | ICommand | RW |
|------------|---|----------|----|

Table 23: Counter and Timer Control Features

6.10 Encoder Control

This section lists all features for the control and the monitoring of quadrature encoders. Quadrature encoders are also known as incremental, rotary and shaft encoders.

| Feature | Description | Interface | Access |
|--------------------|---|--------------|--------|
| EncoderSelector | Selects which Encoder to configure | IEnumeration | RW |
| EncoderSourceA | Selects the signal which will be the source of the A input of the Encoder | IEnumeration | RW |
| EncoderSourceB | Selects the signal which will be the source of the B input of the Encoder | IEnumeration | RW |
| EncoderMode | Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering | IEnumeration | RW |
| EncoderDivider | Sets how many Encoder increment/decrements are needed to generate an Encoder output pulse signal | Integer | RW |
| EncoderOutputMode | Selects the conditions for the Encoder interface to generate a valid Encoder output signal | IEnumeration | RW |
| EncoderValue | Reads or writes the current value of the position counter of the selected Encoder | Integer | RW |
| EncoderResetSource | Selects the signals that will be the source to reset the Encoder | IEnumeration | RW |
| EncoderReset | Does a software reset of the selected Encoder and starts it | ICommand | RW |

Table 24: Encoder Control Features

6.10.1 Encoder interface

Itala can be used to monitor incremental encoders.
 A-B pins of quadrature encoders can be connected to the opto-isolated input pins of Itala cameras.
 The electrical specifications are listed in Table 7 and Table 5.1.

The maximum input frequency of Itala encoder interface is approximately 30KHz.
 This is mainly due to the response times of the opto-isolated circuitry.

NOTE: In case the input frequency is higher than the nominal specifications, sampling errors may occur and, consequently, errors may arise during the encoder monitoring.

6.10.2 Encoder output mode

The **EncoderOutputMode** feature selects one of two following driving modes (see Fig.74):

- **direction mode:** encoder position is incremented only in one direction.
 When moving direction changes, the encoder counter stops counting until the moving direction changes again.
 In this configuration, motion reverse is not considered.
- **position mode:** encoder position is incremented only in one direction.
 When moving direction changes, the encoder counter starts decrementing until the moving direction changes again.
 In this configuration, motion reverse is therefore considered.

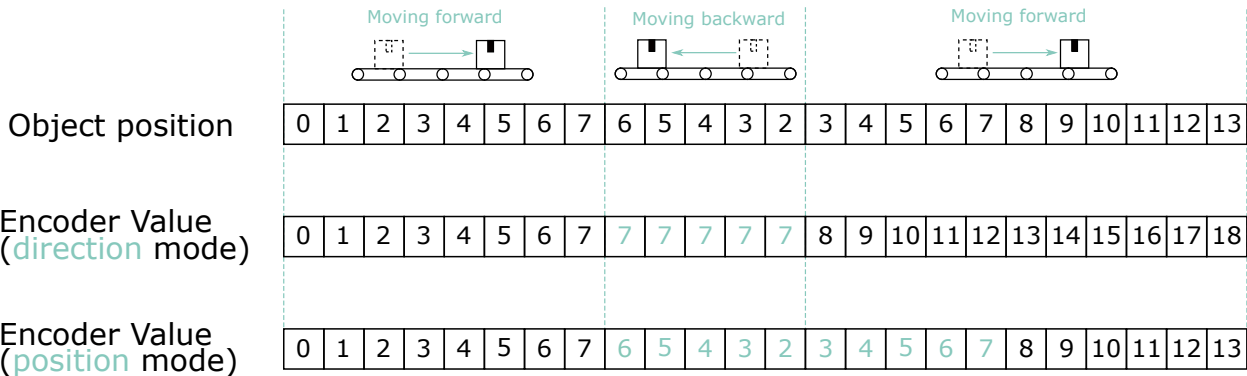


Figure 74: The two different driving modes of the encoder block: (figure above) **direction mode** and (figure below) **position mode**.

6.10.3 Encoder mode

The encoder counter can be incremented/decremented with two different approaches:

- **High resolution mode:** encoder counter is updated (incremented or decremented) for every commutation of *Encoder A* or *Encoder B* signal.

- **4-phase mode:** encoder counter is updated (incremented or decremented) for every full cycle (i.e. for every sequence 00 - 10 - 11 - 01)

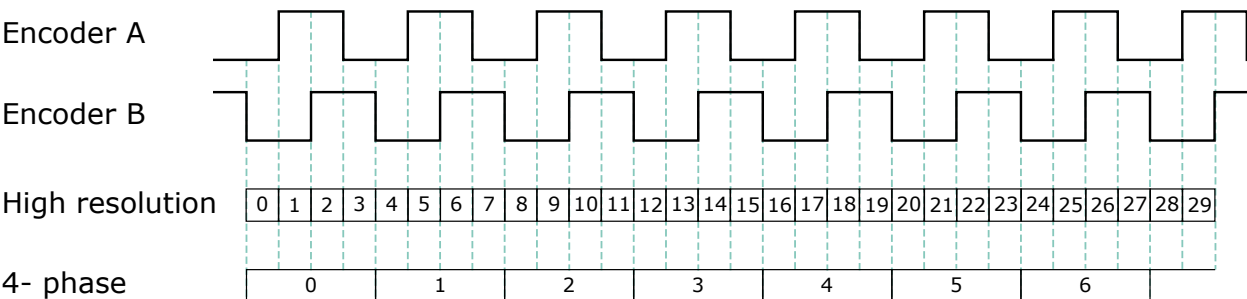


Figure 75: Encoder modes: High resolution (above) vs 4-phase (below).

6.10.4 EncoderValue wrapping management

EncoderValue is a positive value, ranging from 0 to 65535.

When encoder output mode is set to **position mode**, reverse motion is considered in **EncoderValue** computation: since this value cannot be negative, backward movements cause the encoder to reverse count, from 65535 to 0, as depicted in Fig.76.

In this scenario, if reverse motion is not correctly managed, spurious pulses could arise.

Consider the following exemple, where **EncoderDivider** is set equal to 20000.

Consider also a number of backward steps that lead **EncoderValue** to be equal 15000, i.e. a value smaller than **EncoderDivider**. When forward motion restarts, a spurious encoder pulse arises when **EncoderValue** equals **EncoderDivider** (i.e. 20000).

In order to avoid unwanted encoder pulses, the following condition must be satisfied:

$$\text{Maximum backward steps} < (2^{16} - 1) - \text{EncoderDivider} \quad (12)$$

In case Eq.11 could not be guaranteed, spurious encoder pulses must be managed by user application.

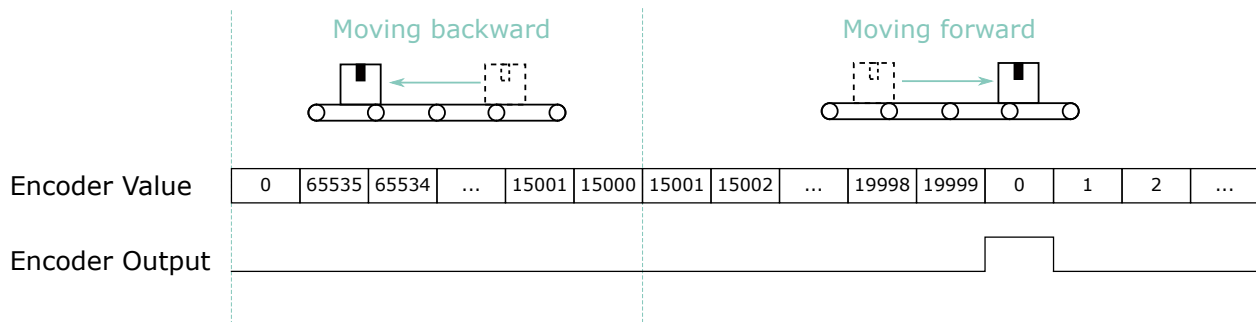


Figure 76: Example of spurious pulse generation: first, reverse motion is applied so that **EncoderValue** indicator goes lower than **EncoderDivider**. Then, when forward motion is restored, a spurious pulse is generated when **EncoderValue** equals **EncoderDivider**

6.11 Logic Block Control

The Logic Block Control section describes the model and features related to the control and the generation of signals by Logic Block elements.

| Feature | Description | Interface | Access |
|-------------------------|---|--------------|--------|
| LogicBlockSelector | Specifies the Logic Block to configure | IEnumeration | RW |
| LogicBlockFunction | Selects the combinational logic Function of the Logic Block to configure | IEnumeration | RW |
| LogicBlockInputNumber | Specifies the number of active signal inputs of the Logic Block | Integer | R |
| LogicBlockInputSelector | Selects the Logic Block's input to configure | Integer | RW |
| LogicBlockInputSource | Selects the source signal for the input into the Logic Block | IEnumeration | RW |
| LogicBlockInputInverter | Selects if the selected Logic Block Input source signal is inverted | Boolean | RW |
| LogicBlockLUTIndex | Controls the index of the truth table to access in the selected LUT | Integer | RW |
| LogicBlockLUTValue | Read or Write the Value associated with the entry at index LogicBlockLUTIndex of the selected LUT | Boolean | RW |

| | | | |
|-----------------------|--|---------|----|
| LogicBlockLUTValueAll | Sets the values of all the output bits of the selected LUT in one access ignoring LogicBlockLUTIndex | Integer | RW |
|-----------------------|--|---------|----|

Table 25: Logic Block Control Features

6.11.1 Logic block module

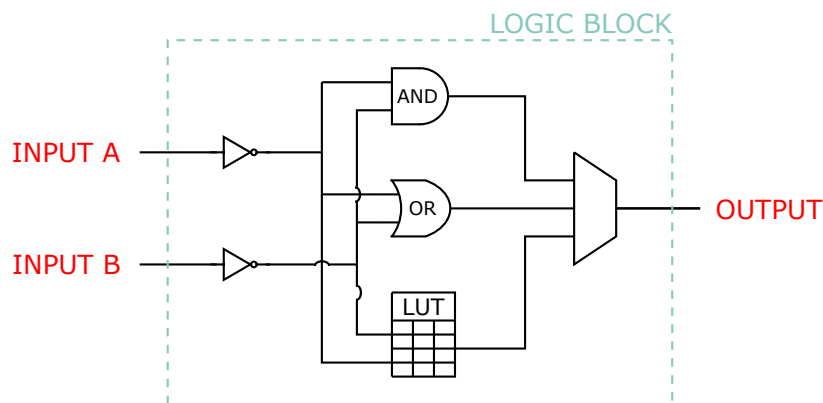


Figure 77: Simplified scheme of the logic block cell.

The **logic block module** is mainly used to generate an output signal depending on two input conditions.

This block is characterized by three different logic functions (see Fig.77):

- **AND**: the output of logic block is HIGH if both the inputs are HIGH;
- **OR**: the output of logic block is HIGH if at least one the inputs is HIGH;
- **LUT**: the user can freely compile the truth table of the lut:

| LogicBlockLUTIndex | Input A | Input B | LogicBlockLUTValue |
|--------------------|---------|---------|-----------------------|
| 0 | 0 | 0 | LogicBlockLUTValue[0] |
| 1 | 0 | 1 | LogicBlockLUTValue[1] |
| 2 | 1 | 0 | LogicBlockLUTValue[2] |
| 3 | 1 | 1 | LogicBlockLUTValue[3] |

Table 26: Example of LUT compilation.

In order to have the maximum flexibility, also an **inverting stage** has been included at the input of this block.

6.12 Action Control

The Action chapter describes all features related to Action Signals in the device.

| Feature | Description | Interface | Access |
|-------------------------|---|--------------|--------|
| ActionUnconditionalMode | Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed. | IEnumeration | RW |
| ActionDeviceKey | Provides the device key that allows the device to check the validity of action commands | Integer | W |
| ActionQueueSize | Indicates the size of the scheduled action commands queue. This number represents the maximum number of scheduled action commands that can be pending at a given point in time. | IEnumeration | R |
| ActionSelector | Selects to which Action Signal further Action settings apply | Integer | RW |
| ActionGroupKey | Provides the key that the device will use to validate the action on reception of the action protocol message | Integer | RW |
| ActionGroupMask | Provides the mask that the device will use to validate the action on reception of the action protocol message | Integer | RW |

Table 27: Action command Control Features

6.12.1 Action Command

Action Command allows the user to trigger actions on one or more GigE cameras at roughly the same time on the ethernet network. Action signals are not synchronous on all devices like hardware triggers because the system is affected by ethernet network latencies, therefore the signal does not reach the devices at the same time. Anyway, using the action command the user can avoid other hardware connections to trigger the camera and use only the ethernet line. Also, action command trigger is better than software trigger on multiple cameras, because only one command is forwarded to all of them.

The Itala camera support one action command, so the user can configure an ActionDeviceKey, an ActionGroupKey and an ActionGroupMask. The camera checks that the command information

matches the configured action command and then generates the trigger. Commands can be sent in unicast or broadcast mode, depending on whether one or more devices should be triggered. To improve the synchronism of the trigger on camera the **Scheduled Action Commands** must be used (6.12.2).

6.12.2 Scheduled Action Command

Scheduled Action Commands allows a trigger action to be generated on one or more devices at a specific time in the future, with nanosecond granularity. The only two requirements are enabling of **PTP** feature and synchronizing the camera with PTP master clock (see 6.20.1), otherwise the scheduled action request is not processed.

The block diagram to explain the architecture of the scheduled action control is shown in Fig.78 and it is discussed in detail below.

When the user sends a scheduled action, the camera checks via hardware that the command information matches the configured action command. Then, if the scheduled action has a time tag in the past, the action is processed immediately, otherwise is added into the hardware action time queue; the queue has a depth of 4, so up to 4 scheduled actions can be handled. If the queue is full, additional commands will be ignored. When the action timestamp becomes less than or equal to the reference time, it is removed from the queue and an hardware trigger toward the Exposure Control is asserted.

If the ACK message is requested, the possible status codes are:

- **GEV_STATUS_NO_REF_TIME:** if the camera has not a reference time synchronized to any master clock. The scheduled action request is not processed.
- **GEV_STATUS_ACTION_LATE:** when the camera received a scheduled action command with time tag in the past (relative to the device timestamp).
- **GEV_STATUS_OVERFLOW:** if the timestamp queue is full and the camera received an action command to be scheduled.

A hardware delay, measured when the scheduled action is configured to trigger a sensor exposure and caused by hardware processing, is removed in the timestamp in order to compensate for it and improve the accuracy of the trigger time.

If TriggerOverlap is ON, the latency between exposure time and frame transfer is affected by a higher uncertainty, therefore the jitter on frame acquisition is higher than when TriggerOverlap is OFF, even if Scheduled Action is used.

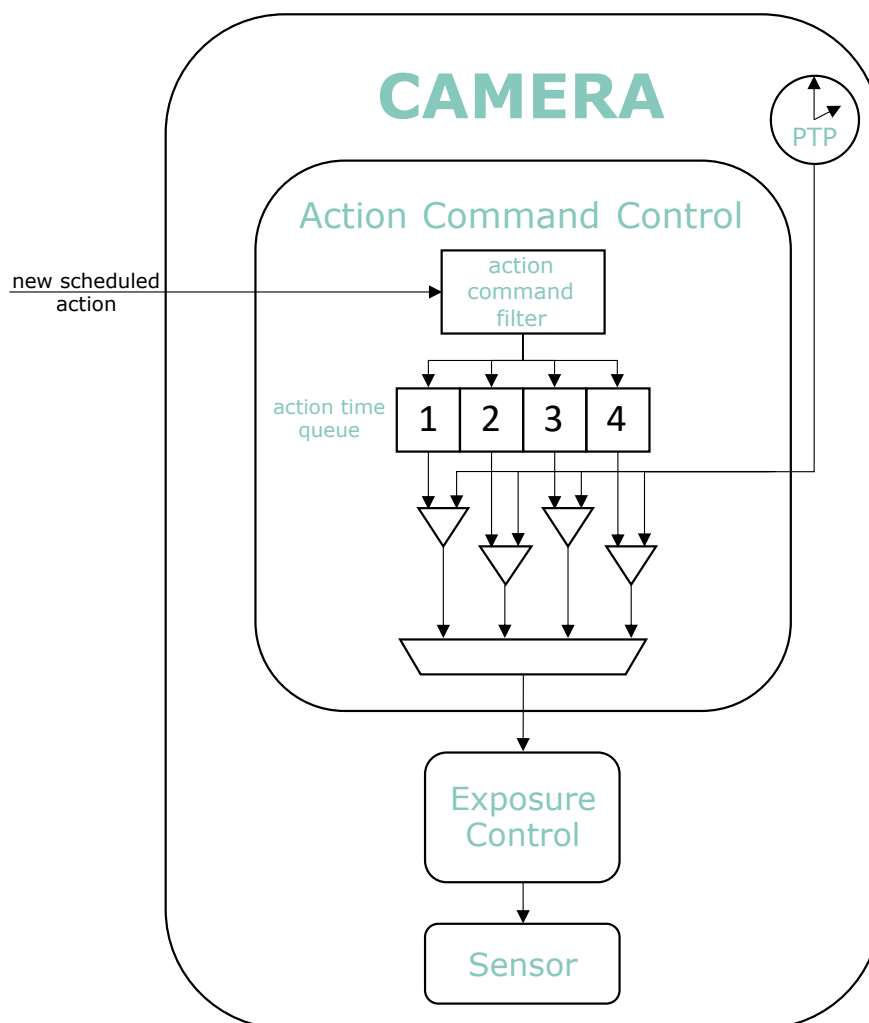


Figure 78: Scheduled Action Command block diagram.

6.13 Event Control

This section describes how to control the generation of events to the host application. An Event is a message that is sent to the host application to notify it of the occurrence of an internal event. Events are typically used to synchronize the host application with some Events happening in the device.

| Feature | Description | Interface | Access |
|---------------|---|--------------|--------|
| EventSelector | Selects which Event to signal to the host application | IEnumeration | RW |

| | | | |
|-----------------------------|---|--------------|----|
| EventNotification | Activate or deactivate the notification to the host application of the occurrence of the selected Event | IEnumeration | RW |
| EventExposureEndData | Category which contains all the attributes related to the ExposureEnd event | ICategory | - |
| EventFrameTriggerMissedData | Category which contains all the attributes related to the FrameTriggerMissed event | ICategory | - |
| EventLine0RisingEdgeData | Category which contains all the attributes related to the Line0RisingEdge event | ICategory | - |
| EventLine1RisingEdgeData | Category which contains all the attributes related to the Line1RisingEdge event | ICategory | - |
| EventTestData | Category which contains all the attributes related to the Test event | ICategory | - |
| EventAutofocusDoneData | Category which contains all the attributes related to the AutofocusDone event | ICategory | - |

Table 28: Event Control Features

| Feature | Description | Interface | Access |
|---------------------------|--|-----------|--------|
| EventExposureEnd | Returns the unique Identifier of the Exposure End type of Event | Integer | R |
| EventExposureEndTimestamp | Returns the Timestamp of the ExposureEnd Event | Integer | R |
| EventExposureEndFrameID | Returns the unique Identifier of the Frame (or image) that generated the ExposureEnd Event | Integer | R |

Table 29: Event Exposure End Data features

| Feature | Description | Interface | Access |
|---------|-------------|-----------|--------|
|---------|-------------|-----------|--------|

| | | | |
|----------------------------------|---|---------|---|
| EventFrameTriggerMissed | Returns the unique Identifier of the Frame Trigger Missed type of Event | Integer | R |
| EventFrameTriggerMissedTimestamp | Returns the Timestamp of the Frame Trigger Missed Event | Integer | R |
| EventFrameTriggerMissed-FramelD | Returns the unique Identifier of the Frame (or image) that generated the Frame Trigger Missed Event | Integer | R |

Table 30: Event Frame Trigger Missed Data features

| Feature | Description | Interface | Access |
|-------------------------------|---|-----------|--------|
| EventLine0RisingEdge | Returns the unique Identifier of the Line 0 Rising Edge type of Event | Integer | R |
| EventLine0RisingEdgeTimestamp | Returns the Timestamp of the Line 0 Rising Edge Event | Integer | R |
| EventLine0RisingEdgeFrameID | Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event | Integer | R |

Table 31: Event Line 0 Rising Edge Data features

| Feature | Description | Interface | Access |
|-------------------------------|---|-----------|--------|
| EventLine1RisingEdge | Returns the unique Identifier of the Line 1 Rising Edge type of Event | Integer | R |
| EventLine1RisingEdgeTimestamp | Returns the Timestamp of the Line 1 Rising Edge Event | Integer | R |
| EventLine1RisingEdgeFrameID | Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event | Integer | R |

Table 32: Event Line 1 Rising Edge Data features

| Feature | Description | Interface | Access |
|---------|-------------|-----------|--------|
|---------|-------------|-----------|--------|

| | | | |
|--------------------|---|---------|---|
| EventTest | Returns the unique Identifier of the Test type of event generated using the TestEventGenerate command | Integer | R |
| EventTestTimestamp | Returns the Timestamp of the Test Event | Integer | R |

Table 33: Event Test Data features

| Feature | Description | Interface | Access |
|-----------------------------|--|-----------|--------|
| EventAutofocusDone | Returns the unique Identifier of the Event Autofocus type of Event | Integer | R |
| EventAutofocusDoneTimestamp | Returns the Timestamp of the Autofocus Done Event | Integer | R |

Table 34: Event Autofocus Done Data features

6.14 User Set Control

This section describes the features for global control of the device settings. It allows loading or saving factory or user-defined settings.

Loading the factory default User Set guarantees a state where a continuous acquisition can be started using only the mandatory features.

| Feature | Description | Interface | Access |
|-----------------|--|--------------|--------|
| UserSetSelector | Selects the feature User Set to load, save or configure | Integer | RW |
| UserSetLoad | Loads the User Set specified by UserSetSelector to the device and makes it active | ICommand | RW |
| UserSetSave | Save the User Set specified by UserSetSelector to the non-volatile memory of the device | ICommand | RW |
| UserSetDefault | Selects the feature User Set to load and make active by default when the device is reset | IEnumeration | RW |

Table 35: User Set Control Features

6.15 Chunk Data Control

This section describes all the features related to the chunk data.

| Feature | Description | Interface | Access |
|-------------------------|---|--------------|--------|
| ChunkModeActive | Activates the inclusion of Chunk data in the payload of the image | IBoolean | RW |
| ChunkSelector | Selects which Chunk to enable or control | IEnumeration | RW |
| ChunkEnable | Enables the inclusion of the selected Chunk data in the payload of the image | IBoolean | RW |
| ChunkWidth | Returns the Width of the image included in the payload | Integer | R |
| ChunkHeight | Returns the Height of the image included in the payload | Integer | R |
| ChunkOffsetX | Returns the OffsetX of the image included in the payload | Integer | R |
| ChunkOffsetY | Returns the OffsetY of the image included in the payload | Integer | R |
| ChunkPixelFormat | Returns the PixelFormat of the image included in the payload | IEnumeration | R |
| ChunkExposureTime | Returns the exposure time used to capture the image | IFloat | R |
| ChunkGain | Returns the gain used to capture the image | IFloat | R |
| ChunkBlackLevel | Returns the black level used to capture the image included in the payload | IFloat | R |
| ChunkTimestamp | Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event | Integer | R |
| ChunkFrameID | Returns the unique Identifier of the frame (or image) included in the payload | Integer | R |
| ChunkSequencerSetActive | Returns the index of the active set of the running sequencer included in the payload | Integer | R |
| ChunkEncoderValue | Returns the value of the Encoder 0 at the time of the FrameStart event | Integer | R |

| | | | |
|-------------------|--|---------|---|
| ChunkCounterValue | Returns the value of the Counter 0 at the time of the FrameStart event | Integer | R |
|-------------------|--|---------|---|

Table 36: Chunk mode Control Features

6.16 OE Serial Interface Control

This section deals with the serial communication features.

| Feature | Description | Interface | Access |
|--------------------------|--|--------------|--------|
| oeSerialEnable | Enable the serial interface | IBoolean | RW |
| oeSerialBaudRate | Select the serial interface baud rate | IEnumeration | RW |
| oeSerialMode | Select the serial interface mode of operation | IEnumeration | RW |
| oeSerialProtocol | Select the protocol to use on the serial interface | IEnumeration | RW |
| oeSerialSlewRate | Select the slew rate of the serial interface data | IEnumeration | RW |
| oeSerialASCIIWriteBuffer | Character write buffer of the serial interface | IString | RW |
| oeSerialASCIIWrite | Start a write operation on the serial interface | ICommand | RW |
| oeSerialASCIIReadCount | Number of bytes to read from serial input buffer | Integer | RW |
| oeSerialASCIIReadBuffer | Character read buffer of the serial interface | IString | R |
| oeSerialASCIIRead | Read the serial input buffer | ICommand | RW |
| oeSerialModbusSlaveID | Modbus Slave ID of target device | Integer | RW |
| oeSerialModbusAddress | Slave Register Address for read/write requests | Integer | RW |
| oeSerialModbusWriteValue | Value to write in Slave Register Address | Integer | RW |
| oeSerialModbusWrite | Send a "Write Single Register" request (0x06) | ICommand | RW |
| oeSerialModbusReadValue | Value to read from Slave Register Address | Integer | R |

| | | | |
|---------------------------|---|-----------|----|
| oeSerialModbusRead | Send a "Read Holding Register" request (0x03) | ICommand | RW |
| oeSerialBinaryWriteBuffer | Binary data buffer to write over the serial interface | IRegister | RW |
| oeSerialBinaryWriteCount | Length of data to write | Integer | RW |
| oeSerialBinaryWrite | Command to write data to the serial interface | ICommand | RW |
| oeSerialBinaryReadBuffer | Binary data read from the serial interface | IRegister | R |
| oeSerialBinaryReadCount | Number of serial bytes to read | Integer | RW |
| oeSerialBinaryRead | Command to read data from the serial interface | ICommand | RW |

Table 37: OE Serial Interface Control Features

6.16.1 Serial interface

The **serial interface** allows to communicate with an external device through a serial connection. This is a dual-mode peripheral which can be configured as a **RS232** or **RS485** transceiver via the **oeSerialMode** feature, as shown in Fig.79. Select the appropriate mode of operation according to the external device you want to connect to the camera. The communication channel in RS232 mode is full-duplex while in RS485 mode is half-duplex.

The serial interface is configured as follows:

- **Baud Rate:** from 9600 to 115200;
- **Data Bits:** 8 bit;
- **Stop Bits:** 1 bit;
- **Parity:** none.



CAUTION: always check the external device specification **before** connecting it to the camera and set the serial mode accordingly. Failing to do so may result in damaging the camera or the external device.

The **slew rate** control affect the shape of the output signal waveform. The **slow** setting yields smooth transitions and may reduce EMI radiation. The **fast** setting yields steep transitions, enabling the use of the highest baud rates. This feature is only available for **RS485** mode.

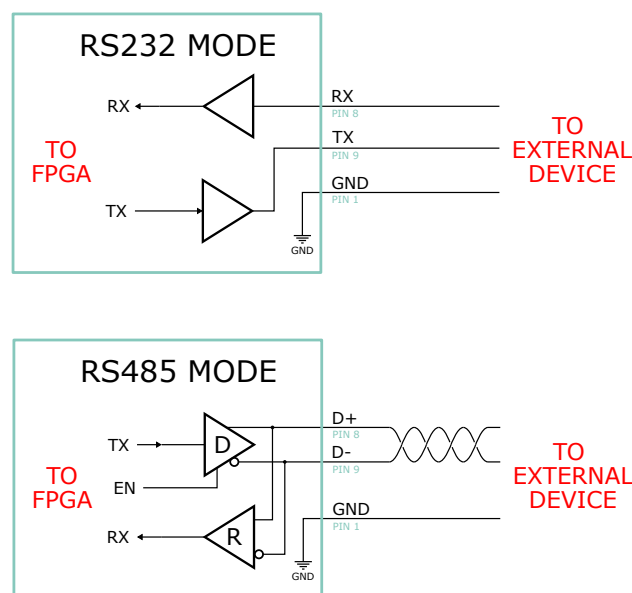


Figure 79: Equivalent circuit of the serial interface in RS232 and RS485 mode of operation.

ASCII Protocol:

The ASCII Protocol allows to send and receive ASCII characters (NULL terminated) over the serial interface. Available both for RS232 and RS485 mode.

Modbus RTU Protocol:

Modbus RTU is a request/reply protocol which provides client/server communication between devices connected on a serial bus. The camera acts like a client and implements a subset of the Modbus RTU function codes such as **Read Holding Registers** and **Write Single Register**. Only available for RS485 mode.

Binary Protocol:

The Binary Protocol allows to send and receive plain binary data over the serial interface. Available both for RS232 and RS485 mode.

6.17 OE Liquid Lens Control

This section describes all the features related to the liquid lens control.

| Feature | Description | Interface | Access |
|--------------------|-----------------------------------|-----------|--------|
| oeLiquidLensEnable | Enable the liquid lens controller | IBoolean | RW |

| | | | |
|-----------------------------------|--|--------------|----|
| oeLiquidLensConfigurationData | Get the configuration for the liquid lens | IEnumeration | R |
| oeLiquidLensSerialNumber | Liquid lens serial number. This string is a unique identifier of the liquid lens | IString | R |
| oeLiquidLensTemperature | Temperature read by the sensor integrated in the liquid lens (available on specific models only) | IFloat | R |
| oeLiquidLensMode | Select the lens control mode | IEnumeration | RW |
| oeLiquidLensMaxPositiveCurrent | Maximum positive current which can be applied to the lens | IFloat | RW |
| oeLiquidLensMaxNegativeCurrent | Maximum negative current which can be applied to the lens | IFloat | RW |
| oeLiquidLensCurrent | Set the liquid lens coil current | IFloat | RW |
| oeLiquidLensPower | Set the liquid lens focal power | IFloat | RW |
| oeLiquidLensResultingCurrent | Resulting liquid lens coil current | IFloat | R |
| oeLiquidLensResultingPower | Resulting liquid lens focal power | IFloat | R |
| oeLiquidLensAutofocusEnable | Enable the autofocus controller | IBoolean | RW |
| oeLiquidLensAutofocusAOISize | Select the autofocus area size | IEnumeration | RW |
| oeLiquidLensAutofocusAOIOffsetX | Horizontal offset from the origin to the area used for autofocus calculations (in pixels) | Integer | RW |
| oeLiquidLensAutofocusAOIOffsetY | Vertical offset from the origin to the area used for autofocus calculations (in pixels) | Integer | RW |
| oeLiquidLensAutofocusStartCurrent | Start current value for autofocus | IFloat | RW |
| oeLiquidLensAutofocusStopCurrent | Stop current value for autofocus | IFloat | RW |

| | | | |
|------------------------------------|---|--------------|----|
| oeLiquidLensAutofocusStartPower | Start power value for autofocus | IFloat | RW |
| oeLiquidLensAutofocusStopPower | Stop power value for autofocus | IFloat | RW |
| oeLiquidLensAutofocusFrameCount | Number of frames to acquire for autofocus | Integer | RW |
| oeLiquidLensAutofocusStart | Start autofocus | ICommand | RW |
| oeLiquidLensAutofocusTriggerSource | Specifies the internal signal or physical input Line to use as the autofocus trigger source | IEnumeration | RW |
| oeLiquidLensAutofocusStatus | Returns the autofocus status | IEnumeration | R |

Table 38: OE Liquid Lens Control Features

6.17.1 Liquid Lens interface

The **liquid lens interface** allows to control an Opto Engineering® product with liquid lens technology directly from the camera device. This ensure maximum integration with the camera SDK and compatibility with third party software thanks to *GigE Vision* and *GenTL* standards. The interface can operate in two different modes:

- EEPROM mode;
- manual mode;

In **EEPROM mode** the camera automatically detects the connected lens and read the calibration data from the embedded EEPROM. Through the *GenICam* feature tree is possible to read the lens attributes and directly set the lens focal power. The EEPROM also includes a temperature sensor used by the controller for thermal compensation of the lens current. This ensures a costant focal power across a wide range of operating temperature. It's also possible to directly control the lens current and check for the actual resulting focal power, which depends on the lens temperature. This mode is automatically selected when a compatible lens is connected.

In **manual mode** is possible to control a lens without an embedded EEPROM, directly setting the current of the actuation coil. In this case the user is responsible to set the correct values and to not exceed the limits reported in the lens specifications.

oeLiquidLensConfigurationData shows if the lens is equipped with a calibration EEPROM or if the peripheral is running in manual mode.



CAUTION: always check the lens specification before connecting it to the camera. If the lens is not equipped with a calibration EEPROM, check and set current limits **before** connecting the lens. Failing to do so may result in damaging the camera or the liquid lens.

6.17.2 Autofocus

The autofocus is a completely in-camera feature that allows to automatically find the best focus controlling an Opto Engineering® product with liquid lens technology directly from the camera device.

The autofocus algorithm is not applied to the entire frame, so a smaller AOI must be selected via the **oeLiquidLensAutofocusAOISize** selector and the **oeLiquidLensAutofocusAOIOffsetX** and **oeLiquidLensAutofocusAOIOffsetY** features.

Depending on the **oeLiquidLensMode** feature, a current or power range must be set.

Finally, with the **oeLiquidLensAutofocusFrameCount** feature, a number of frame must be set.

NOTE: Once the **oeLiquidLensAutofocusStart** command is triggered, the camera waits for the last frame to be transferred. Then the autofocus starts.

To determine if autofocus is complete, check **oeLiquidLensAutofocusStatus**.

The autofocus performance is affected by its configuration:

1. The lower the current/power range, the better the accuracy.
2. The smaller the AOI, the faster the focusing time.
3. The shorter the exposure time, the better the accuracy and the faster the focusing time.
4. The higher the number of frames, the better the accuracy but the slower the focusing time.

The following formula determines the expected minimum focusing time:

$$MinAutofocusTime[ms] = 1000 \cdot \frac{N_{frame} - 1}{FPS[s^{-1}]} + t_{exp}[ms] \quad (13)$$

where, the FPS value must be evaluated setting a sensor ROI equal to the **oeLiquidLensAutofocusAOISize** and setting the **oeAcquisitionFrameRateLimitMode** to **oe Sensor Throughput**.

The **defocus** parameter is a figure of merit that helps determine whether the current autofocus configuration will produce an accurate result. The higher the defocus value, the worst the focusing accuracy:

$$defocus[dpt] = slope[dpt/ms] \cdot t_{exp}[ms] \quad (14)$$

where,

$$slope[dpt/ms] = \frac{PowerRange[dpt]}{MinAutofocusTime[ms]} \quad (15)$$

NOTE: The focusing time is variable and always greater than the Min Autofocus Time. It depends on the camera settings and the time it takes to download the last frame before the autofocus starts.

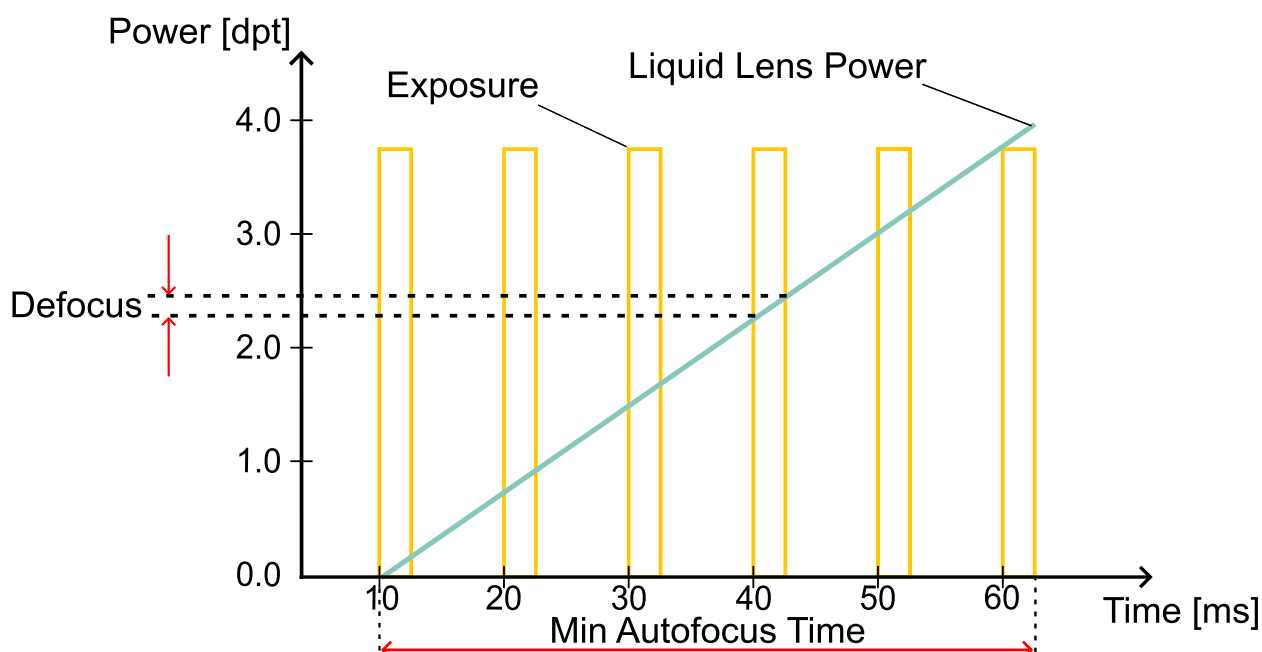


Figure 80: Autofocus working principle: exposure time and liquid lens behavior are depicted. The liquid lens performs a sweep across the current/power range..

NOTE: Focusing accuracy may be affected by exposure start jitter when **TriggerOverlap** is set to **ReadOut**.

NOTE: For higher accuracy, a **dual-pass** behavior can be implemented via the SDK. First, a coarse pass helps determine the region of best focus, then a fine pass provides a more accurate result. This can be accomplished by acting on the current/power range.

6.18 OE Defective Pixel Correction Control

This section describes all the features related to the correction of the defective pixels.

| Feature | Description | Interface | Access |
|--------------------------|--|-----------|--------|
| oeDefectivePixelCount | Shows the number of the defective pixels | Integer | RW |
| oeDefectivePixelSelector | Represents the index of the defective pixel inside the defective pixel map | Integer | RW |

| | | | |
|-----------------------------|--|----------|----|
| oeDefectivePixelXCoordinate | Represents the horizontal coordinate of the actual defective pixel | Integer | RW |
| oeDefectivePixelYCoordinate | Represents the vertical coordinate of the actual defective pixel | Integer | RW |
| oeDefectivePixelWriteMap | Write the defective pixel map in the camera non-volatile memory | ICommand | RW |

Table 39: OE Defective Pixel Correction Control Features

6.18.1 Defective Pixel Correction

Image sensors can be affected of pixel degradation for multiple causes (temperature, aging, cosmic rays, ionizing radiation and so on..).

A possible way to overcome these effects is to adopt a defective pixel correction strategy. This consists in replacing the defective pixel value with the one of a near good pixel. This algorithm is executed real-time in the camera acquisition pipeline and rely on a defective pixel coordinates table.

NOTE: The automatic procedure for the pixels defects detection and correction is explained in section 4.7.5. Here there is the explanation of the single defective pixel correction only.

oeDefectivePixelCount is the indicator of the actual defective pixels corrected in camera. The defective pixels coordinates can be displayed on the nodes **oeDefectivePixelXCoordinate** and **oeDefectivePixelYCoordinate** after selecting the pixel index (**oeDefectivePixelSelector**).

The following example shows how to manually add a new defective pixel to the defective pixels list. Let's consider a defective pixel at coordinates (4,2) (see Fig.81). To correct this pixel:

1. Increase by 1 the number of **oeDefectivePixelCount**;
2. Select the first available index in **oeDefectivePixelSelector** node: the correct index is the one with the un-initialized **oeDefectivePixelXCoordinate** and **oeDefectivePixelYCoordinate** values (be aware that pixel enumeration starts from 0);
3. Insert the coordinates of the defective pixel in the **oeDefectivePixelXCoordinate** and **oeDefectivePixelYCoordinate** fields;
4. Save the new map in the onboard-memory with **oeDefectivePixelWriteMap** command;

In case of color cameras, the color correction algorithm takes into account that the adjacent pixel has a different chroma information, therefore the correction is performed with the following pixel value, as depicted in Fig.82.

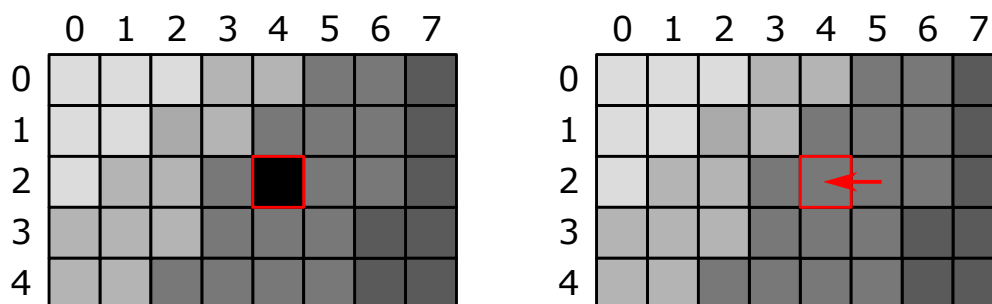


Figure 81: (On the left) Presence of a dead pixel at coordinates (4,2). (On the right) Error correction through the nearest neighbor algorithm.

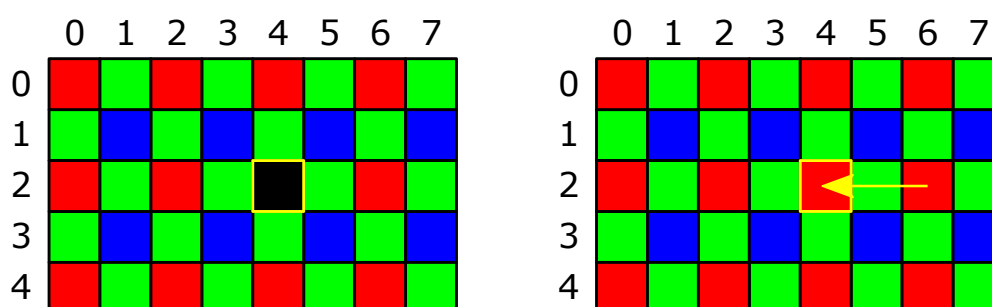


Figure 82: (On the left) Presence of a dead pixel at coordinates (4,2). (On the right) Error correction through the nearest neighbor (but with the same chroma info) algorithm.

6.19 Test Control

Contains the features related to the control of the test features.

| Feature | Description | Interface | Access |
|-------------------|------------------------|-----------|--------|
| TestEventGenerate | Generates a Test Event | ICommand | W |

Table 40: Test Control Features

6.20 Transport Layer Control

This section provides the Transport Layer control features.

| Feature | Description | Interface | Access |
|-------------|--|-----------|--------|
| PayloadSize | Provides the number of bytes transferred for each image or chunk on the stream channel | Integer | R |

| | | | |
|---------------------------------------|--|--------------|----|
| PtpEnable | Enables the Precision Time Protocol (PTP) | IBoolean | RW |
| oePtpOffsetFromUtc | Enables applying current IEEE 1588 UTC offset to the output time | IBoolean | RW |
| PtpDataSetLatch | Latches the current values from the device's PTP clock data set | ICommand | W |
| PtpStatus | Returns the latched state of the PTP clock | IEnumeration | R |
| PtpServoStatus | Returns the latched state of the clock servo | IEnumeration | R |
| PtpOffsetFromMaster | Returns the latched offset from the PTP master clock in nanoseconds | Integer | R |
| PtpClockID | Returns the latched clock ID of the PTP device | Integer | R |
| PtpParentClockID | Returns the latched parent clock ID of the PTP device | Integer | R |
| PtpGrandmasterClockID | Returns the latched grandmaster clock ID of the PTP device | Integer | R |
| GevSupportedOptionSelector | Selects the GEV option to interrogate for existing support | IEnumeration | RW |
| GevSupportedOption | Returns if the selected GEV option is supported | IBoolean | R |
| GevInterfaceSelector | Selects which logical link to control | Integer | RW |
| GevMACAddress | MAC address of the logical link | Integer | R |
| GevCurrentIPConfigurationLLA | Controls whether the Link Local Address IP configuration scheme is activated on the given logical link | IBoolean | RW |
| GevCurrentIPConfigurationDHCP | Controls whether the DHCP IP configuration scheme is activated on the given logical link | IBoolean | RW |
| GevCurrentIPConfigurationPersistentIP | Controls whether the PersistentIP configuration scheme is activated on the given logical link | IBoolean | RW |
| GevCurrentIPAddress | Reports the IP address for the given logical link | Integer | R |

| | | | |
|-----------------------------|--|---------|----|
| GevCurrentSubnetMask | Reports the subnet mask of the given logical link | Integer | R |
| GevCurrentDefaultGateway | Reports the default gateway IP address of the given logical link | Integer | R |
| GevPersistentIPAddress | Controls the Persistent IP address for this logical link | Integer | RW |
| GevPersistentSubnetMask | Controls the Persistent subnet mask associated with the Persistent IP address on this logical link | Integer | RW |
| GevPersistentDefaultGateway | Controls the persistent default gateway for this logical link | Integer | RW |
| GevDiscoveryAckDelay | Indicates the maximum randomized delay the device will wait to acknowledge a discovery command | Integer | R |
| GevMCPHostPort | Controls the port to which the device must send messages | Integer | R |
| GevMCDA | Controls the destination IP address for the message channel | Integer | RW |
| GevMCTT | Provides the transmission timeout value in milliseconds | Integer | RW |
| GevMCRC | Controls the number of retransmissions allowed when a message channel message times out | Integer | RW |
| GevMCSP | This feature indicates the source port for the message channel | Integer | R |
| GevStreamChannelSelector | Selects the stream channel to control | Integer | RW |
| GevSCPIInterfaceIndex | Index of the logical link to use | Integer | RW |
| 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 | Integer | R |
| GevSCPSFireTestPacket | Sends a test packet. When this feature is set, the device will fire one test packet | Boolean | RW |

| | | | |
|----------------------|---|---------|----|
| GevSCPSDoNotFragment | The state of this feature is copied into the "do not fragment" bit of IP header of each stream packet. It can be used by the application to prevent IP fragmentation of packets on the stream channel | Boolean | RW |
| GevSCPSPacketSize | This GigE Vision specific feature corresponds to DeviceStreamChannelPacketSize and should be kept in sync with it | Integer | RW |
| GevSCPD | Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel | Integer | R |
| 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 | Integer | RW |
| GevSCSP | Indicates the source port of the stream channel | Integer | R |

Table 41: Transport Layer Control Features

6.20.1 Precision Time Protocol (PTP)

PTP (Precision Time Protocol) is a clock synchronization protocol of IEEE 1588 standard. It allows to precisely synchronize clocks of multiple GigE cameras on an ethernet network. PTP procedure establishes that the device with the most accurate clock in the network is elected as grandmaster clock and other devices becomes slaves. Slaves periodically and automatically synchronize their clock directly with the master's clock. The result is that the timestamp values are aligned with the master over the entire network. This protocol is closely described in the IEEE standard document.

NOTE: *Itala cameras can become only slave (master mode is not implemented).*

The PTP feature must be enabled if the **scheduled action commands** are used (6.12.2).

6.21 Sequencer Control

This section describes all the features related to the Sequencer Control.

| Feature | Description | Interface | Access |
|----------------------------|---|--------------|--------|
| SequencerMode | Controls if the sequencer mechanism is active | IEnumeration | RW |
| SequencerConfigurationMode | Controls if the sequencer configuration mode is active | IEnumeration | RW |
| SequencerFeatureSelector | Selects which sequencer features to control | IEnumeration | RW |
| SequencerFeatureEnable | Enables the selected feature and make it active in all the sequencer | IBoolean | RW |
| SequencerSetSelector | Selects the sequencer set to which further feature settings applies | Integer | RW |
| SequencerSetSave | Saves the current device state to the sequencer set selected by the SequencerSetSelector | ICommand | W |
| SequencerSetLoad | Loads the sequencer set selected by SequencerSetSelector in the device | ICommand | W |
| SequencerSetActive | Contains the currently active sequencer set | Integer | R |
| SequencerSetStart | Sets the initial/start sequencer set, which is the first set used within a sequencer | Integer | RW |
| SequencerPathSelector | Selects to which branching path further path settings applies | Integer | RW |
| SequencerSetNext | Specifies the next sequencer set | Integer | RW |
| SequencerTriggerSource | Specifies the internal signal or physical input line to use as the sequencer trigger source | IEnumeration | RW |
| SequencerTriggerActivation | Specifies the activation mode of the sequencer trigger | IEnumeration | RW |

Table 42: Sequencer Control Features

6.21.1 Sequencer overview

The purpose of the Sequencer Control is to allow the user to define a series of feature sets which can consecutively be activated during the acquisition. Each *sequencer set* change is triggered by an

event configured by the user.

The execution of the sequencer is completely controlled by the device.

6.21.2 Configuration of a Sequencer set

The index of the *sequencer set* is given by the **SequencerSetSelector**. Up to 16 sequencer sets can be configured.

The features which are actually part of a *sequencer set* are defined in Table 43. These features can be selected by **SequencerFeatureSelector** and activated by **SequencerFeatureEnable**. If a feature is enabled it is for all *sequencer sets*.

To configure a *sequencer set* the camera has to be switched into configuration mode by **Sequencer-ConfigurationMode**. Then the user has to select the desired *sequencer set* he wants to modify with the **SequencerSetSelector**. After the user has changed all the needed camera settings it is possible to store all these settings within a selected *sequencer set* by **SequencerSetSave**. The user can also read back this settings by **SequencerSetLoad**.

To permit a flexible usage, up to two paths are available to go from one *sequencer set* to another. Such a path is selected by **SequencerPathSelector**. Each path and therefore the transition between different *sequencer sets* is based on a defined trigger and an upcoming targeted *sequencer set* which is selectable by **SequencerSetNext**. After the trigger occurs the settings of the next set become active.

The trigger is defined by the features **SequencerTriggerSource** (Table 44) and **SequencerTriggerActivation**.

NOTE: *SequencerTriggerActivation* is set to "RisingEdge" by default and cannot be changed.

A *sequencer set* should contain the following values:

- Camera data which should be controlled by the device
- **SequencerPathSelector** with at least one path
- **SequencerSetNext**, **SequencerTriggerSource** and **SequencerTriggerActivation** for every path which is selectable by the **SequencerPathSelector**.

NOTE: *If two paths are configured, Path 0 has a higher priority than Path 1. If two different SequencerTriggerSource occur at the same time, the trigger source associated with Path 0 wins.*

The following features are currently available to be used inside the Sequencer Control:

| Feature | Note |
|--------------|------|
| ExposureTime | |

| | |
|---------------------|---|
| CounterDuration | Only Counter0 is configured |
| OffsetX | A proper ROI must be pre-configured |
| OffsetY | A proper ROI must be pre-configured |
| Gain | |
| oeLiquidLensCurrent | oeLiquidLensMode must be set in CurrentMode |
| oeLiquidLensPower | oeLiquidLensMode must be set in PowerMode |

Table 43: Available features for Sequencer Control operations

SequencerTriggerSource values supported by Itala cameras are:

| Feature | Note |
|-------------|--|
| Off | Disables the sequencer trigger |
| ExposureEnd | Starts with the reception of the ExposureEnd |
| Counter0End | Starts with the reception of the Counter0End |
| Timer0End | Starts with the reception of the Timer0End |
| Encoder0 | Starts with the reception of the Encoder output signal |

Table 44: Available trigger sources for Sequencer Control operations

NOTE: Configured sequencer programs are stored as part of the UserSets like any other feature.

NOTE: Actual settings of the camera are overwritten when a sequencer set is loaded.

7 USE CASES

7.1 Wiring connection examples

7.1.1 Triggering the camera by an external device

To trigger Itala cameras in a machine vision system, suitable connections must be performed. Considering the circuitry of the opto-isolated input pin (Section 5.7), possible connections are the depicted in Figure 83.

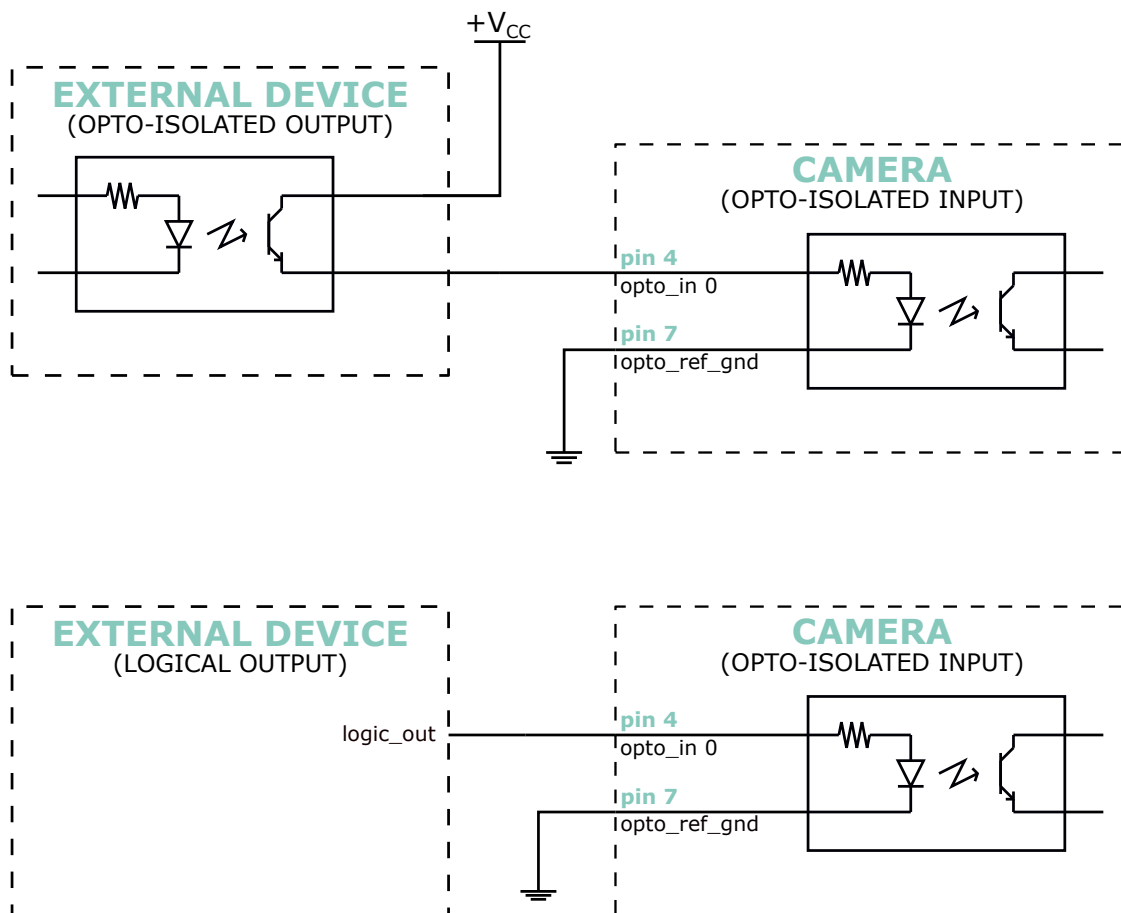


Figure 83: Figure above: camera is triggered by an opto-isolated external device. Figure below: camera is triggered by a logic output pin.

When Itala camera is triggered by an external **opto-isolated** device, the input pin of the camera can be directly wired to the source output pin of the triggering device.

In this case the triggering output stage acts as a switch: when the synchronization signal is generated, the switch closes and the external power supply ($+V_{CC}$) is delivered to the camera input pin, toggling the actual state and therefore triggering the camera.



CAUTION: Please be careful not to exceed the maximum voltage specification of the opto-isolated input pins.
As mentioned in Section 5.2, $+V_{CC}$ must not exceed 30V.

When Itala camera is triggered by an external **logic** pin (e.g. TTL), the output pin can still be wired to the opto-isolated input pin of the camera: in this case the logic output pin must be capable of triggering the opto-isolated input stage, i.e. the high logic level must be greater than the threshold voltage of the opto-coupler (see Section 5.2).

Moreover, the output pin must have a suitable drive strength in order to toggle the opto-isolated input stage.

7.1.2 Synchronizing an external device with Itala cameras

When Itala camera is used to trigger external devices, suitable connections must be performed. Considering the circuitry of the opto-isolated output pin (Section 5.7), possible connections are the depicted in Figure 84.

When Itala camera triggers an external **opto-isolated** device, the output pin of the camera can be directly wired to the input pin of the triggered device.

In this case the output stage acts as a switch: when the synchronization signal is generated, the switch closes and the external power supply ($+V_{CC}$) is delivered to the external device, toggling the actual state and therefore triggering the device.



CAUTION: Please be careful not to exceed the maximum voltage specification of the opto-isolated input pins.
As mentioned in Section 5.2, $+V_{CC}$ must not exceed 30V.

On the opposite, when Itala camera triggers an external **logic** pin (e.g. TTL), the output pin can still be wired to the opto-isolated input pin of the camera with some cautions: an external resistor is required in order to tie the input pin to ground when the opto-isolated output is not active.



CAUTION: Please be careful not to exceed the maximum voltage specification of the opto-isolated input pins.
As mentioned in Section 5.2, $+V_{CC}$ must not exceed 30V.



CAUTION: Always check the compatibility between $+V_{CC}$ and the logic pin maximum voltage ratings.

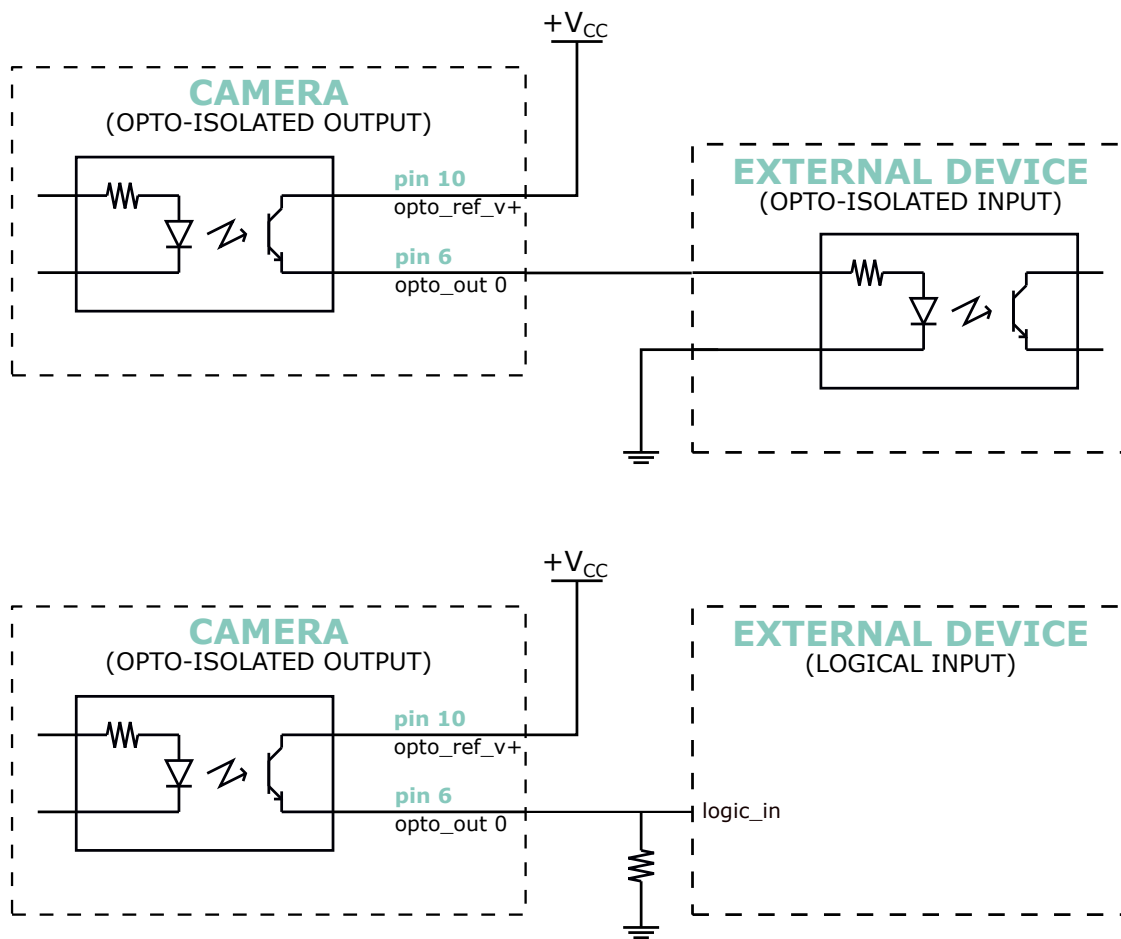


Figure 84: Figure above: camera triggers an opto-isolated external device. Figure below: camera triggers a logic input pin.

7.2 How to add a delay on the Output Lines of the camera

In this section is shown how to add a user defined delay for Itala output lines.

For example, let's consider to generate an output pulse on **Line2**, whose duration is equal to 1ms and delayed by $100\mu\text{s}$, and triggered by the *Exposure End* feature.

This scenario is depicted in Figure 85.

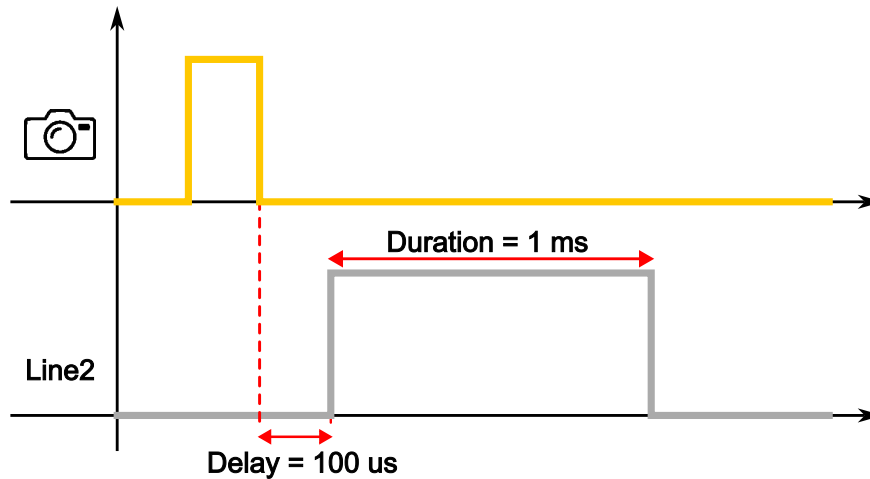


Figure 85: Example scenario: a 1ms output pulse (with $100\mu\text{s}$ delay) is generated after the exposure time.

In order to obtain this waveform on **Line2**, the following settings can be adopted:

1. Timer configuration

- Select one of the timers using the *Timer Selector* feature (i.e. *Timer 0*).
- Select the *Exposure End* entry for the *Timer Trigger Source* feature.
- Set the desired pulse delay in the *Timer Delay* field (i.e. $100\mu\text{s}$).
- Set the desired pulse duration in the *Timer Duration* field (i.e. $1000\mu\text{s}$).

2. Digital IO configuration

- Select one of the output lines using the *Line Selector* feature (i.e. *Line 2*).
- Select the *Timer 0 Active* entry for the *Line Source* feature.

If Itala View is used, the mentioned configuration is shown in Figure 86.

In particular, the Timer configuration is highlighted by green boxes, while the Digital IO configuration by the yellow ones.

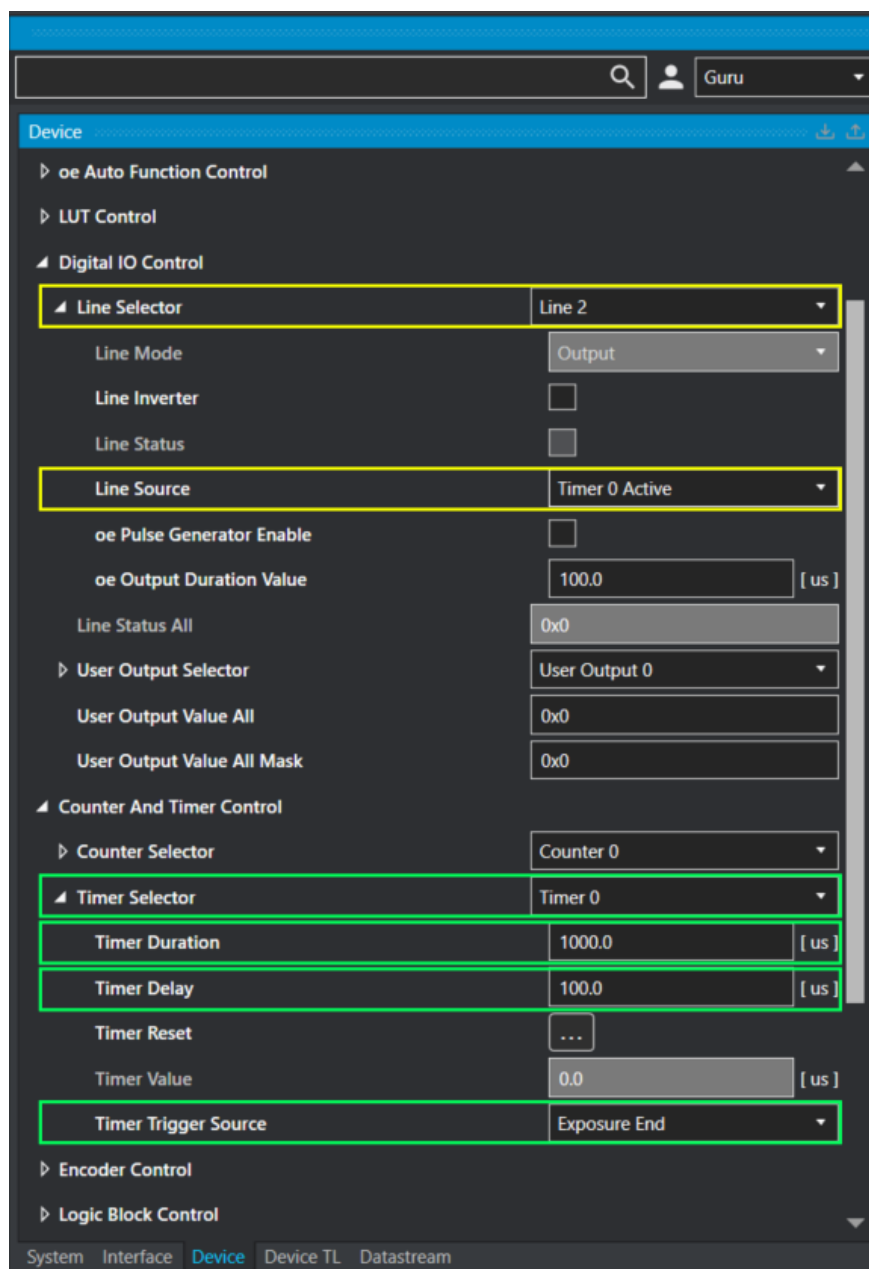


Figure 86: Implementation of an output delayed pulse using Itala View

7.3 Streaming bandwidth management

Itala has the possibility to select (and to adjust) the overall acquisition frame rate from the image sensor: in particular, the GenICam custom feature involved in the sensor throughput settings is **oeAcquisitionFrameRateLimitMode**.

By default, **oeAcquisitionFrameRateLimitMode** is set to **oeLinkThroughput**.

In this configuration, the acquisition frame rate is limited by the gigabit link bandwidth.

Sensors settings (like triggering logic, blanking periods and so on...) are automatically computed in order to match the throughput of the gigabit connection, i.e. 1 Gbps.

This scenario is depicted in Figure 87: frames, represented by turquoise blocks, are captured from the image sensor and stored in the internal camera buffer; then are read by the user application. Figure 87 also shows the behaviour of the on-board image buffer: since *read* data rate is always equal to *write* data rate, there isn't the possibility for the internal memory to be completely full.

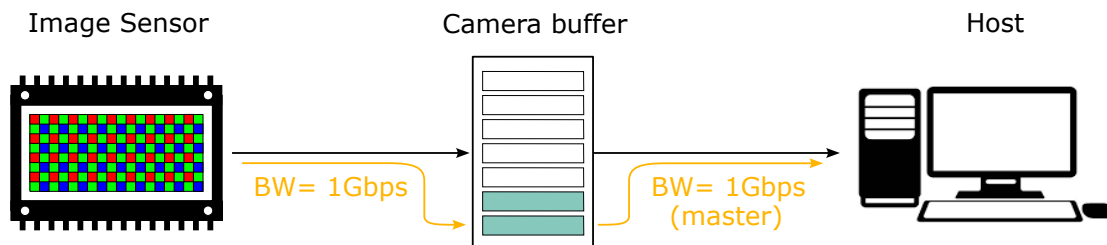


Figure 87: When *oeAcquisitionFrameRateLimitMode* = *oeLinkThroughput*, sensor throughput is automatically adjusted in order to match the gigabit bandwidth.

On the other hand, when *oeAcquisitionFrameRateLimitMode* is set to **oeSensorThroughput**, the driver of the acquisition bandwidth becomes the image sensor, as depicted in Fig 88.

In this case, the overall bandwidth between camera and host is still limited by the ethernet interface, however, acquisition data rate is no more related to the link bandwidth and its value can be higher or lower than the gigabit link speed, depending on the image sensor model and its mode of operation.

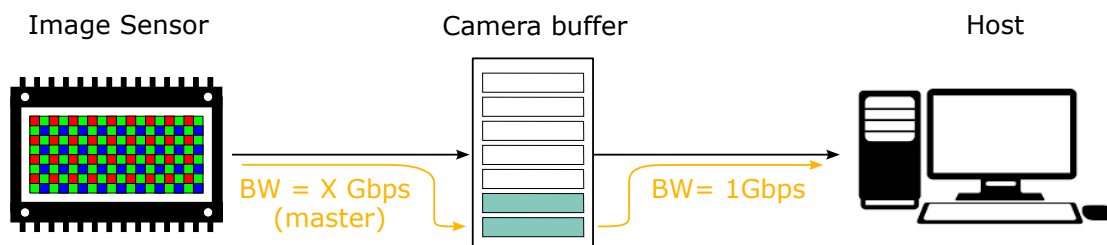


Figure 88: When *oeAcquisitionFrameRateLimitMode* = *oeSensorThroughput*, acquisition bandwidth is completely unrelated to the link bandwidth.

Since *write* data rate can be higher than *read* data rate, the internal image buffer of the camera can saturate, as depicted in Fig 89. When this condition is met, a lagging effect can arise during the visualization of the captured frames.

This scenario is almost always present when Itala is in free-run acquisition mode and *oeAcquisitionFrameRateLimitMode* is set to *oeSensorThroughput*.

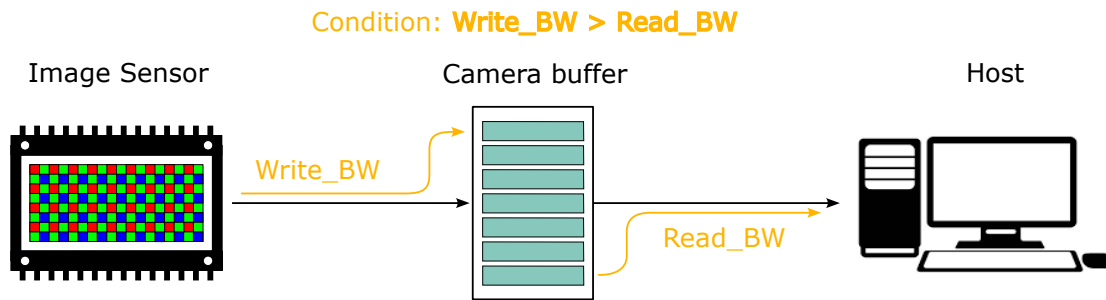


Figure 89: When *oeAcquisitionFrameRateLimitMode* = *oeSensorThroughput*, the *buffer-full* condition can occur if the write bandwidth is higher than read bandwidth.

oeSensorThroughput configuration really shows its benefits when used in conjunction with trigger mode and burst acquisitions, i.e. when **TriggerMode** is set to ON and **AcquisitionBurstFrame-Count** is higher than 1.

In this case, it's possible to grab images from the image sensor at very high frame rate and to store them in the internal memory buffer (Fig 90a), while the readout can happen subsequently, with the lower gigabit bandwidth (Fig 90b).

An important factor to consider is the finite size of the internal buffer: the number of frames which can be stored in memory are strictly related and dependent on two parameters: *image resolution* and *pixel format*.

The maximum number of frames which can be captured before buffer saturation is expressed by the following formula:

$$MaxBurstSize = \frac{BufferSize[Mbit]}{Resolution[Mpixel] * BitPerPixel} \quad (16)$$

where:

- *Buffer Size* is the internal memory buffer size (see Section 5.1 for buffer size info).
- *Resolution* is the sensor resolution (width x height), expressed in Megapixels.
- *Bit Per Pixel* is the number of bit associated to each single pixel of the image.

Table 45 briefly summarize the *Bit per Pixel* value for each pixel format supported by Itala cameras.

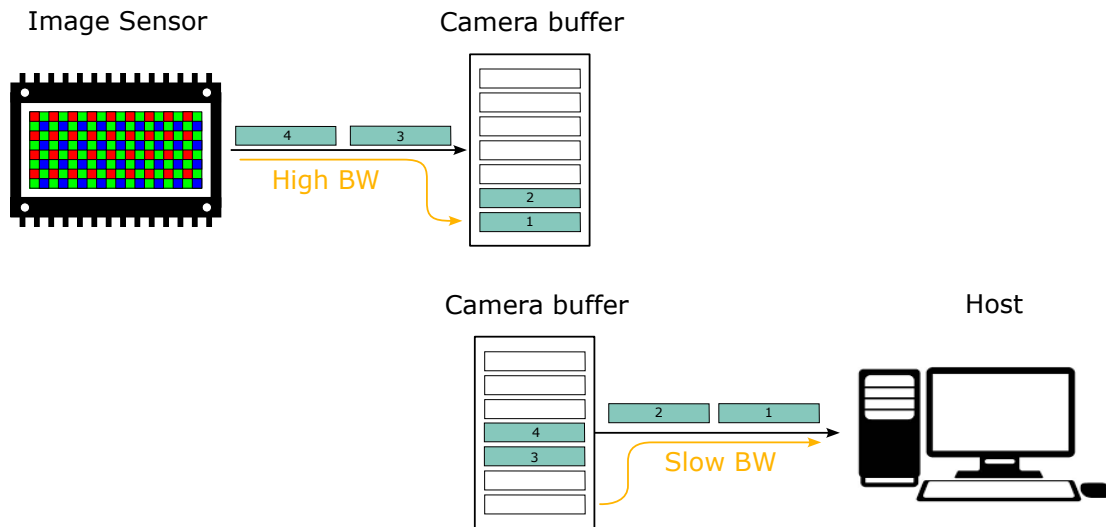


Figure 90: (a) First, burst acquisition is enabled in order to grab multiple frames with only one input trigger signal. (b) Second, images can be transferred from the camera to the host with a slower frame rate.

| Pixel Format | Bit per Pixel |
|----------------------------|---------------|
| Mono8 | 8 |
| BayerRG8 | 8 |
| PolarizedXXMono8 | 8 |
| PolarizedXXBayerRG8 | 8 |
| Mono10p | 10 |
| BayerRG10p | 10 |
| PolarizedXXMono10p | 10 |
| PolarizedXXBayerRG10p | 10 |
| Mono10Packed | 12 |
| BayerRG10Packed | 12 |
| PolarizedXXMono10Packed | 12 |
| PolarizedXXBayerRG10Packed | 12 |
| Mono12p | 12 |
| BayerRG12p | 12 |
| PolarizedXXMono12p | 12 |
| PolarizedXXBayerRG12p | 12 |
| Mono12Packed | 12 |
| BayerRG12Packed | 12 |
| PolarizedXXMono12Packed | 12 |

| | |
|----------------------------|----|
| PolarizedXXBayerRG12Packed | 12 |
| YUV411 | 12 |
| YUV422 | 16 |
| RGB8 | 24 |

Table 45: Recap of the Bit per Pixel value for each pixel format supported by Itala cameras

The following example will show a rough computation of a maximum burst size estimation without buffer saturation.

NOTE: In the following example the worst case scenario is considered, in which the download of the frames to the host application starts **after** the end of the burst acquisition.

NOTE: In the following example a buffer size equal to 384 Mb is considered (i.e. 3072 Mbit). Actual buffer size can be found in Section 5.1.

Considering a **12Mpixel** image sensor and a pixel format set to **Mono8**, the maximum number of frames which can be acquired before saturation will be equal to:

$$MaxBurstSize = \frac{3072Mbit}{12Mpixel * 8} = 32frames \quad (17)$$

A camera with the same sensor, set to work with **RGB8** pixel format, will have a maximum burst size equal to:

$$MaxBurstSize = \frac{3072Mbit}{12Mpixel * 24} = 10frames \quad (18)$$

7.4 Cognex Vision Pro compatibility

This is a small technical guide to briefly explain how to setup Cognex Vision Pro with Itala. Please follow these steps:

1. Connect the camera..
2. Ensure IPv4 configuration and camera IP mode is correct.
3. Open *Cognex GigE Vision Configuration*.
4. Ensure that the *Performance driver* is installed.

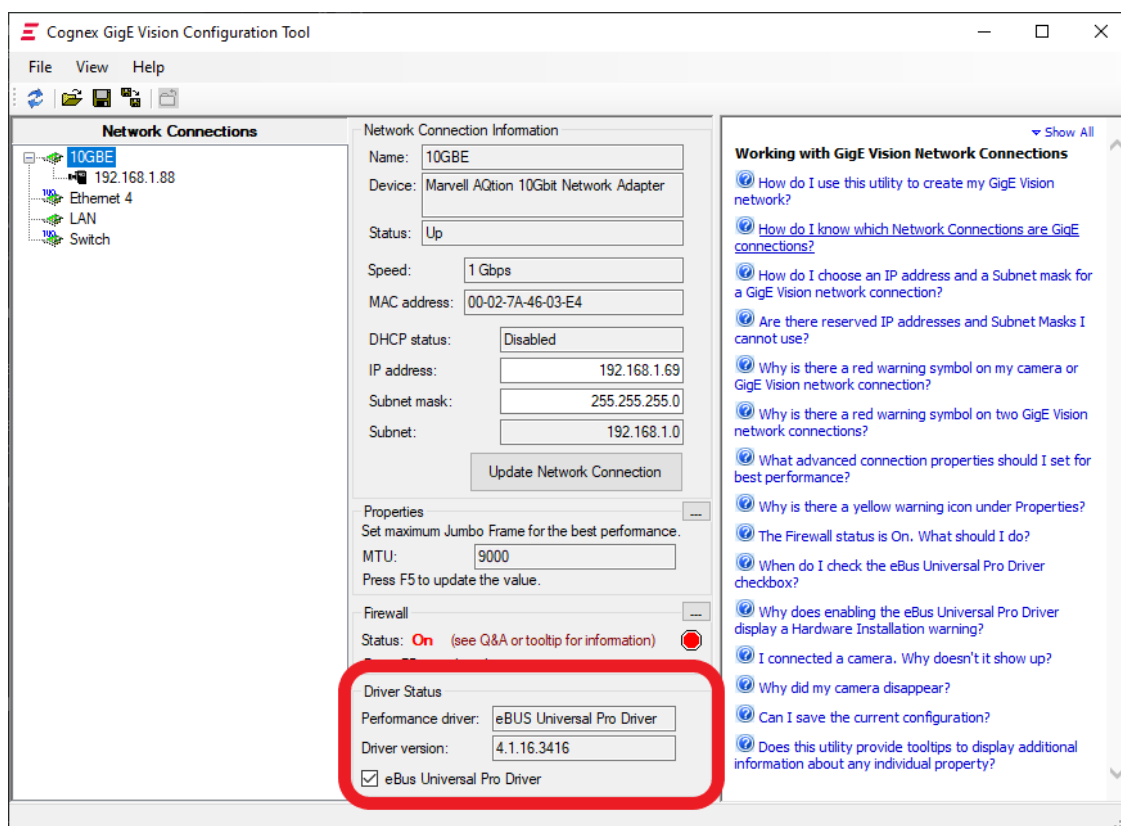


Figure 91: STEP 4 - *Cognex GigE Vision Configuration Tool* window.

5. Open *VisionPro QuickBuild*.

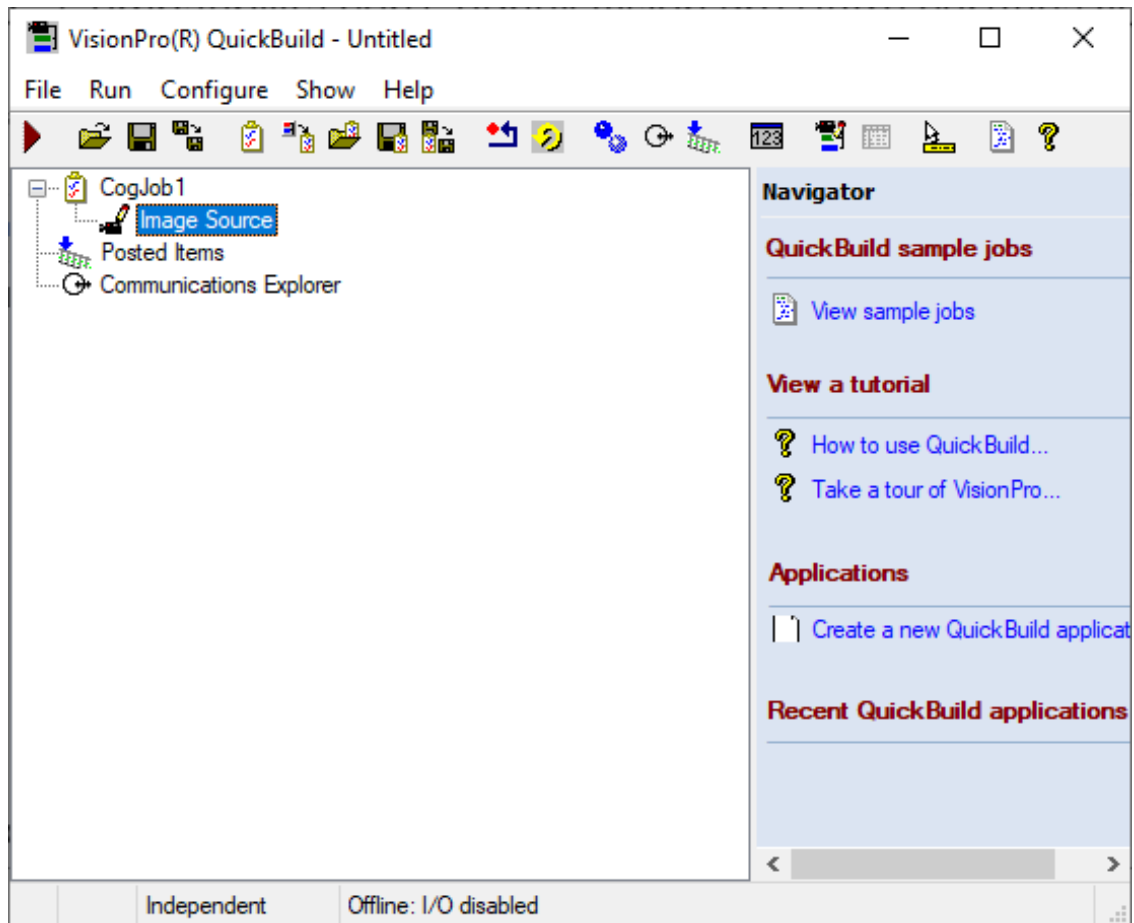


Figure 92: STEP 5 - *VisionPro QuickBuild* window.

6. Double click on *Image Source*.
7. Two windows will open up. On the *Image Source* window, click on the *Camera* button.

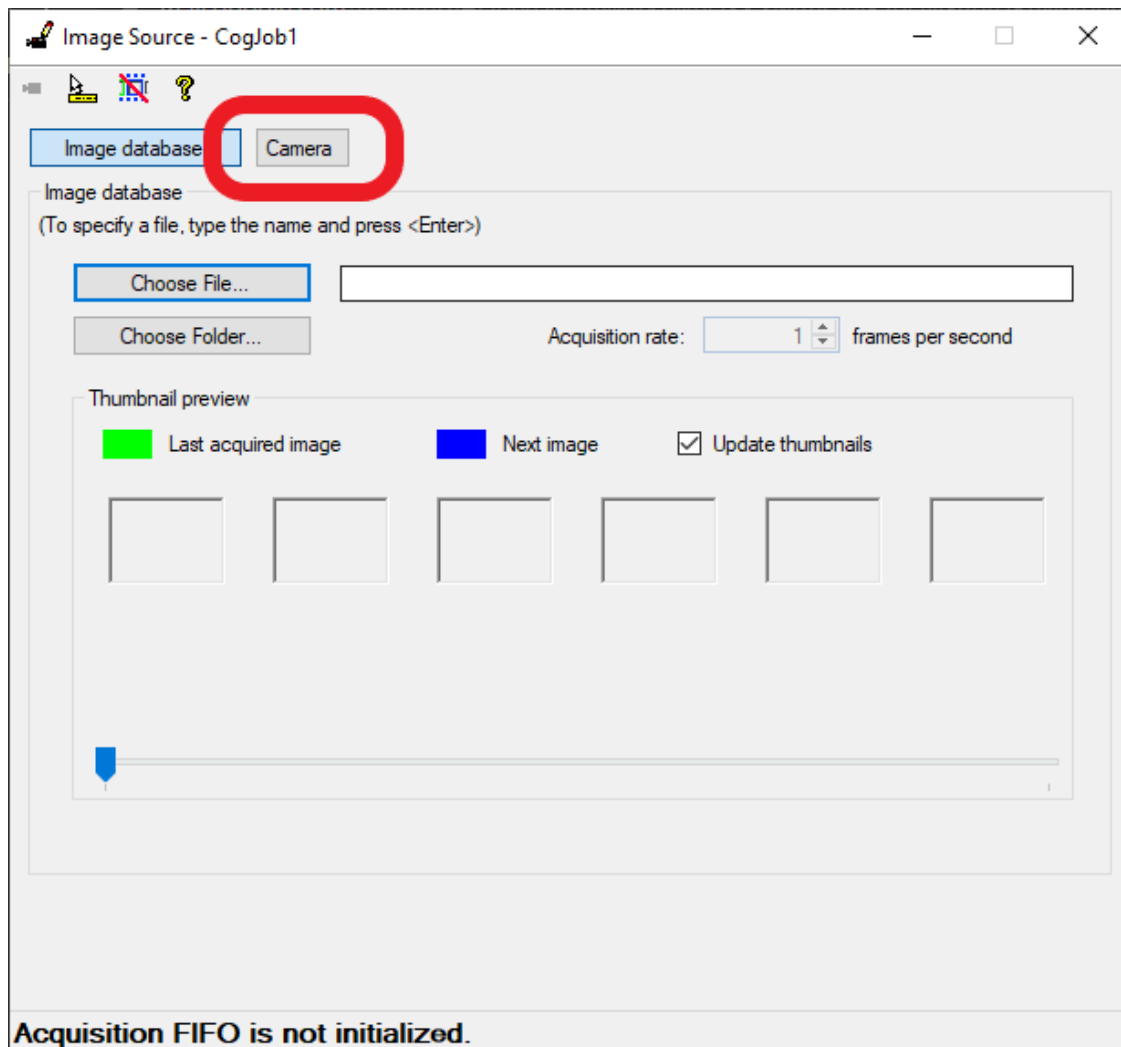


Figure 93: STEP 7 - *Image Source* window.

8. From the combo box select the entry that starts with *GigEVision*.

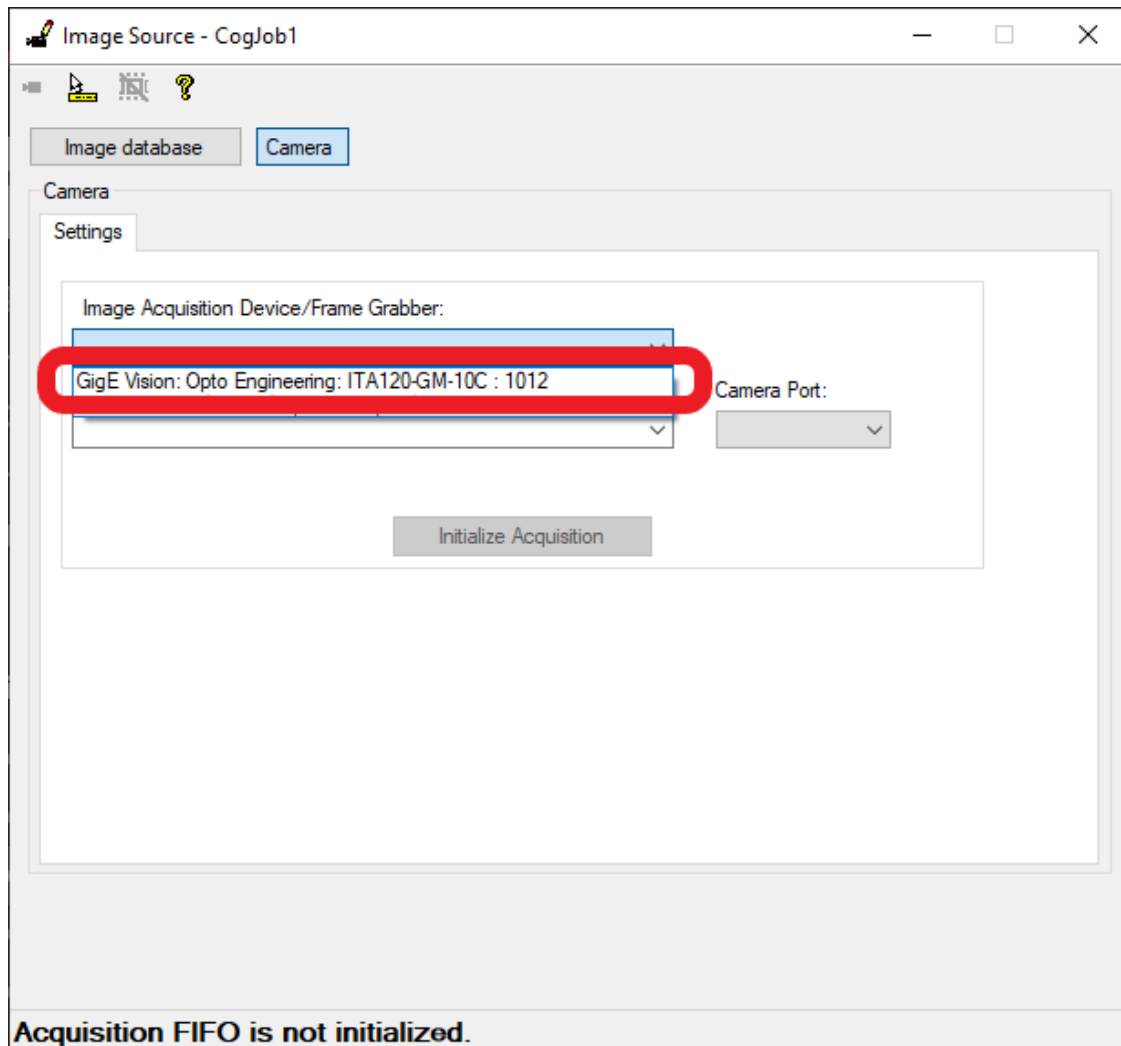


Figure 94: STEP 8 - *Image Source* window, *Image Acquisition Device* selection.

9. Click on the descending arrow on the *Video Formats* field and select the desired pixel format.

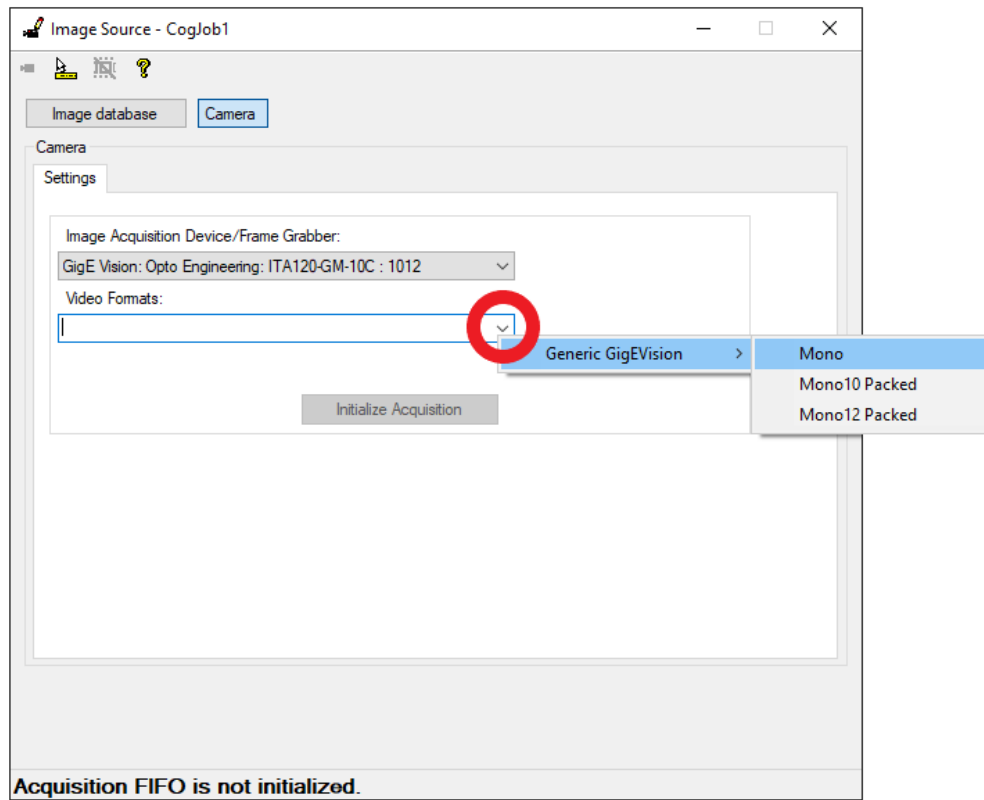


Figure 95: STEP 9 - *Image Source* window, *Video Formats* selection.

10. Click on *Initialize Acquisition*.

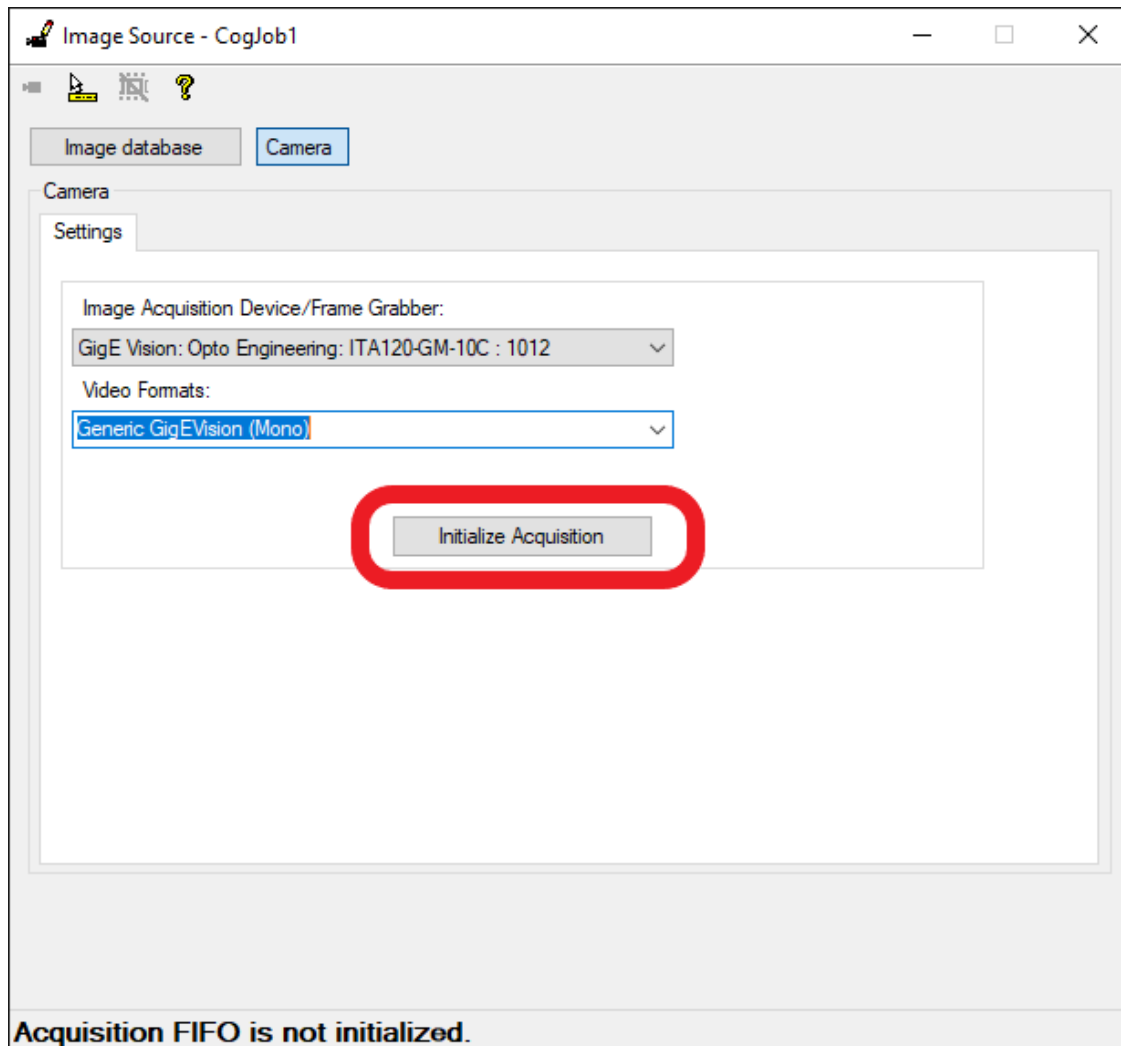


Figure 96: STEP 10 - *Image Source* window, *Initialize Acquisition*.

11. For a live preview click on the camera icon on the top left corner of the window. It is possible to set also the exposure time, trigger mode and other settings such as ROIs.

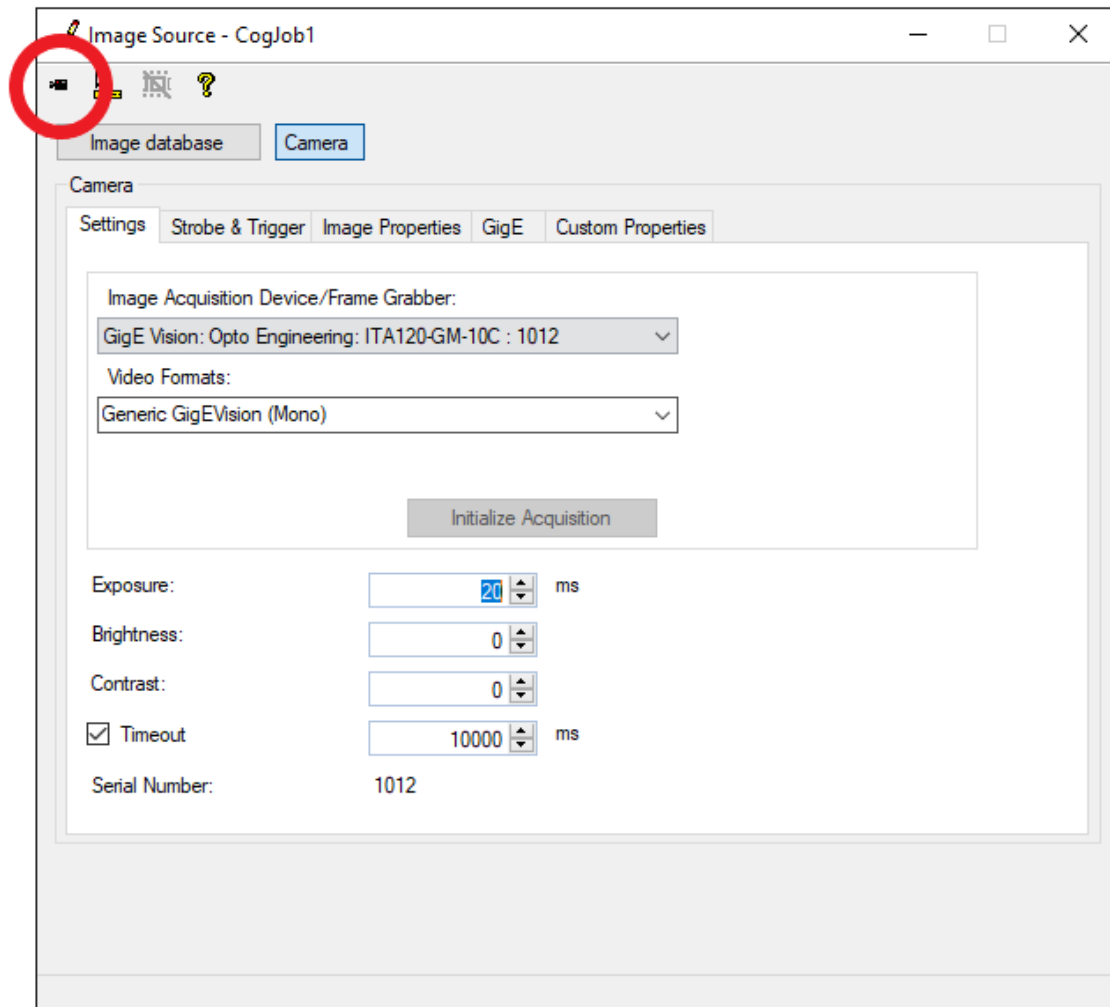


Figure 97: STEP 11 - *Image Source* window, *Live Preview*.

7.5 Sequencer Control configuration example

In this example, we define an acquisition sequence with four different exposure times on the device, in which the last step is repeated five times.

All configuration is done on the device itself, so after configuration is complete and acquisition is started, the device itself will apply the parameter changes if necessary. The host application then only needs to acquire the images. This results in a much faster overall frame rate than if these changes were applied frame by frame by the host application.

This will result in the following flow diagram:

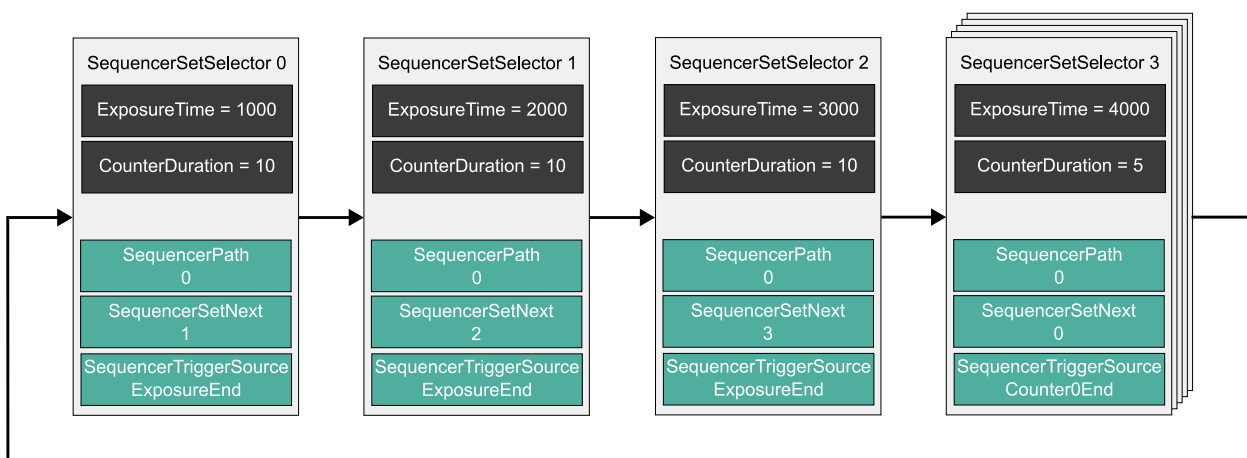


Figure 98: Sequencer Control example flow diagram.

There are some specifications concerning the sequencer path feature:

- A path is inactive as soon as the **SequencerTriggerSource** is Off.
- If both paths are inactive or none of the **SequencerTriggerSource** is triggered, the sequencer will remain in the current set.
- If the **SequencerTriggerSource** of both paths is triggered, the path with the trigger that occurred first will be followed.

7.5.1 Working with Sequencer Paths

It is possible to define sets with a maximum of two active paths. The following diagram shows an example using the **ExposureTime** and **oeLiquidLensPower** features, where two paths are defined in "Set 0" and "Set 1":

- "Set 0" and "Set 1" loops at each "ExposureEnd" following Path 0. When Timer0 ends, the next set is "Set 2", following Path 1.
- "Set 2" returns to "Set 0" after "ExposureEnd"

Additional settings:

- **TimerDuration** (Timer 0) = 2000000 μs
- **oeLiquidLensMode** = Power Mode

This will result in the following flow diagram:

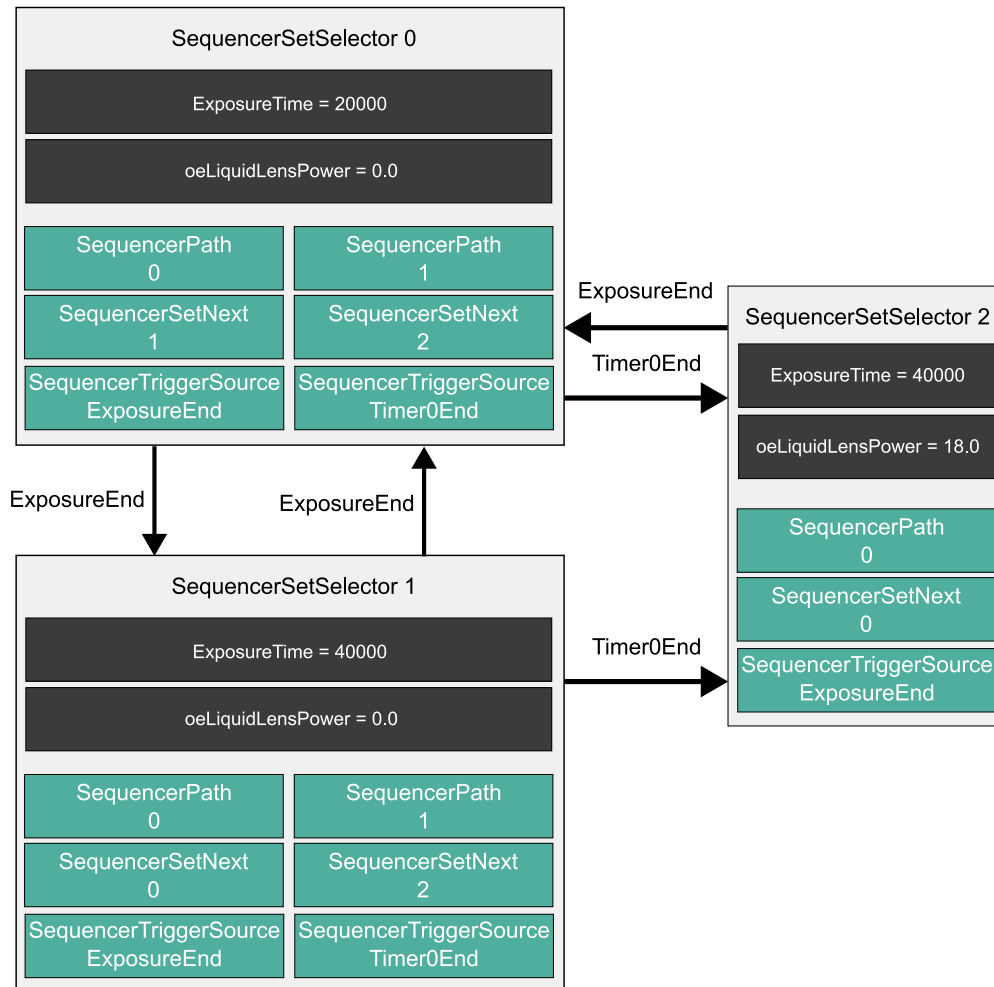


Figure 99: Sequencer Control paths example flow diagram.

8 TROUBLESHOOTING

8.1 The camera cannot be found in the available device list

When the camera is not detected and it's not available in the device list, check the following steps:

1. Check that the camera is correctly powered. When the camera is turned on, the status LED becomes yellow after few seconds.
2. Check if a firewall is currently blocking the communication between the host and the device.
3. Check the configuration of the NIC (network interface controller).
By default, the camera is configured to have an IP address assigned by a DHCP server.
The user, however, can assign a static IP address to the camera: in this case check that the interface card has a suited IP address, compatible to the camera one.
As alternative, use the *IP Configurator tool* to properly configure the IP of the camera.
4. Check that network card's drivers are correctly installed (and updated to the last version).
5. If the camera is currently used by another application, the camera results unavailable for the actual process. In this case, disconnect the camera from the other application and connect the camera to the desired application again.
6. Check that the cable is not damaged.

8.2 Why some features are not present in the GenICam tree of the camera viewer?

When some feature are missing check the following points:

1. Check that the feature is actually available for the selected camera model.
For example, typical color features (like RGB pixel formats) are not available for monochrome cameras.
2. Check the visibility mode of the viewer.
Some features are not visible in *Beginner mode*, but can be displayed only in *Expert mode* or *Guru mode*.
3. Some new features may be added in following firmware releases: check that the camera FW is always up to date.

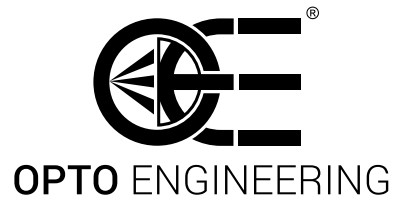
8.3 Why does the camera give frame losses?

When the camera and/or the network card are not properly configured, some frames may be lost. When this is happening, check the following potential causes:

1. Check the the GigE Vision capture driver is correctly installed.
2. Check the network interface drivers are currently up to date.
3. Check that the *jumbo packet* option of the network interface card is enabled.
Jumbo packets support frames larger than 1500 bytes and give optimal performance on high-bandwidth cameras.
4. Check that the network interface card receive buffer is correctly dimensioned. In case of frame losses, try to increase the receive buffer size.
5. Check that the PC is not in *power saving mode*. In this working regime, CPU performances are strongly reduced and may cause frame losses.
6. Check that the bandwidth doesn't exceed the supported rate of 1 Gigabit.
As a rule of thumb, BW can be approximated quite well by the equation 19:

$$BW[Mbps] = Resolution[Mpixel] * BitPerPixel[bit/pixel] * FrameRate[fps] \quad (19)$$

Along with this check, the *DeviceLinkThroughputLimit* feature may 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.



Contact us

EUROPE

Opto Engineering Headquarters

str. Circonvallazione Sud, 15
46100 Mantova, IT
phone: +39 0376 699111
eu@opto-e.com

Opto Engineering Germany

Marktplatz 3,
82031 Grünwald, DE
phone: +49 (0)89 693 9671-0
de@opto-e.com

UNITED STATES

Opto Engineering USA

11321 Richmond Ave
Suite M-105, Houston, TX 77082, USA
phone: +1 832 2129391
us@opto-e.com

ASIA

Opto Engineering China

Room 1903-1904, No.885, Renmin RD
Huangpu District 200010
Shanghai, CN
phone: +86 21 61356711
cn@opto-e.com

www.opto-e.com