BU series

CMOS Camera

Instruction Manual

Model

B/W Camera	: BU040M / BU040MG / BU160M / BU160MG
Color Camera	: BU040MCG / BU040MCF / BU160MCG / BU160MCF

Thank you for purchasing our product.
Before using this CMOS camera, please read through this instruction manual carefully in order to use this product correctly and safely.
After reading, keep this instruction manual handy so that you can refer to, whenever you need it.

Toshiba Teli Corporation

Information contained in this document is subject to change without prior notice.

Standard name might be trade mark of each company.

DAA00195A

http://www.toshiba-teli.co.jp/en/

Contents

Safety Precautions

Before using this product, read these safety precautions carefully. Important information is shown in this Instruction Manual to protect users from bodily injuries and property damages, and to enable them to use the product safely and correctly.

Please be sure to thoroughly understand the meanings of the following signs and symbols before reading the main text that follow, and observe the instructions given herein.

[Definition of Safety Signs]

Safety Signs	Description
⚠ WARNING	Indicates a potentially hazardous situation that may result in death or serious injury (*1) in the event of improper handling.
⚠ CAUTION	Indicates a potentially hazardous situation that may result in light to moderate injuries (*2) or only in property damage (*3)in the event of improper handling.

Notes *1:“Serious injury” refers to cases of loss of eyesight, wounds, burns (high or low temperature), electric shock, broken bones, poisoning, etc., which leave after-effects or which require hospitalization or a long period of outpatient treatment of cure.

*2: "Light to moderate injuries" refers to injuries, burns, electric shock etc. that do not require hospitalization or long-term treatment.

*3: "Property damage" refers to cases of extensive damage involving damage to buildings, equipment, farm animals, pet animals and other belongings.

[Explanation of Safety Symbols]

Safety Symbols	Description
🛇 PROHIBITED	This sign indicates **PROHIBITION** (Do not). The content of prohibition is shown by a picture or words beside the symbol.
● MANDATORY	This sign indicates **MANDATORY ACTION** (You are required to do). The content of action is shown by a picture or words beside the symbol.

General Handing

⚠ WARNING

Unplug	● **Stop operation immediately when any abnormality or defect occurs.** If abnormal conditions are present, such as smoke, a burning smell, ingress of water or foreign matter, or if the equipment is dropped or malfunctions, fire or electric shock may result. Be always sure to disconnect the power cable from the wall socket at once and contact your dealer.
Do not get wet	● **Do not use the equipment in locations subject to water splashes.** Otherwise, fire or electric shock may result.
Never pull apart	● **Do not disassemble, repair, or modify the equipment.** Otherwise, fire or electric shock may result. For internal repair, inspection, or cleaning, contact your sales representative.
Avoid	● **Do not place anything on the equipment.** If metallic objects, liquid, or other foreign matter enters the equipment, fire or electric shock may result.
Avoid	● **Do not install the equipment in an unstable or inclined location or locations subject to vibration or impact.** Otherwise, the equipment may topple over and cause personal injury.
Do not touch	● **During an electrical storm, do not touch the power cable and the connection cable.** Otherwise, an electric shock may result.
Instruction	● **Use the specified voltage.** Use of an unspecified voltage may result in fire or electric shock.
Avoid	● **Do not be handled roughly, damaged, fabricated, bent forcefully, pulled, twisted, bundled, placed under heavy objects or heated the power cable and the connection cable.** Otherwise, fire or electric shock may result.

⚠ CAUTION

Instruction

- **Observe the following when installing the equipment:**
  ·Do not cover the equipment with a cloth, etc.
  ·Do not place the equipment in a narrow location where heat is likely to accumulate.
  Otherwise, heat will accumulate inside the equipment, possibly resulting in a fire.

Avoid

- **Do not place the equipment in locations subject to high moisture, oil fumes, steam, or dust.**
  Otherwise, fire or electric shock may result.

Avoid

- **Do not install the equipment in locations exposed to direct sunlight or humidity.**
  Otherwise, the internal temperature of the equipment will rise, which may cause a fire.

Instruction

- **Use only specified the power cable and the connection cables.**
  Otherwise, fire or electric shock may result.

Avoid

- **Do not give strong impact against the equipment.**
  It may cause the trouble.

Instruction

- **When performing connection, turn off power.**
  When connecting the power cable and the connection cable, turn off the equipment power.
  Otherwise, fire or electric shock may result.

Avoid

- **Do not expose its camera head to any intensive light (such as direct sunlight).**
  Otherwise, its inner image pickup device might get damaged.

Avoid

- **Avoid short-circuiting signal output.**
  Otherwise, a malfunction may occur.

Avoid

- **Avoid giving a strong shock against the camera body.**
  It might cause a breakdown or damage. If your camera is used in a system where its connector is subjected to strong repetitive shocks, its connector is possible to break down. If you intend to use your camera in such a situation, if possible, bundle and fix a cable in the place near the camera, and do not transmit a shock to the connector.

Instruction

- **Contact your sales representative to request periodic inspection and cleaning (every approx five years).**
  Accumulation of dust inside the equipment may result in fire or electric shock.
  For inspection and cleaning costs, contact your sales representative.

CASES FOR INDEMNITY (LIMITED WARRANTY)

We shall be exempted from taking responsibility and held harmless for damage or losses incurred by the user in the following cases.

- In the case damage or losses are caused by natural disasters, such as an earthquake and thunder, fire, or other acts of God, acts by a third party, deliberate or accidental misuse by the user, or use under extreme operating conditions.

- In the case of indirect, additional, consequential damages (loss of business interests, suspension of business activities) are incurred as result of malfunction or non-function of the equipment, we shall be exempted from responsibility for such damages.

- In the case damage or losses are caused by failure to observe the information contained in the instructions in this instruction manual and specifications.

- In the case damage or losses are caused by use contrary to the instructions in this instruction manual and specifications.

- In the case damage or losses are caused by malfunction or other problems resulting from unintended use of equipment or software etc. that are not specified.

- In the case damage or losses are caused by repair or modification conducted by the customer or any unauthorized third party (such as an unauthorized service representative).

- Expenses we bear on this product shall be limited to the individual price of the product.

- The item that is not described in specifications of this product is out of the guarantee.

- The case of damages or losses which are caused by incorrect connection of the cable is out of the guarantee.

RESTRICTION FOR USE

- Should the equipment be used in the following conditions or environments, give consideration to safety measures and inform us of such usage:
  1. Use of the equipment in the conditions or environment contrary to those specified, or use outdoors.
  2. Use of the equipment in applications expected to cause potential hazard to people or property, which require special safety measures to be adopted.

- This product can be used under diverse operating conditions. Determination of applicability of equipment or devices concerned shall be determined after analysis or testing as necessary by the designer of such equipment or devices, or personnel related to the specifications. Such designer or personnel shall assure the performance and safety of the equipment or devices.

- This product is not designed or manufactured to be used for control of equipment directly concerned with human life (*1) or equipment relating to maintenance of public services/functions involving factors of safety (*2). Therefore, the product shall not be used for such applications.

  (*1): Equipment directly concerned with human life refers to.

  - Medical equipment such as life-support systems, equipment for operating theaters.
  - Exhaust control equipment for exhaust gases such as toxic fumes or smoke.
  - Equipment mandatory to be installed by various laws and regulations such as the Fire Act or Building Standard Law
  - Equipment related to the above

  (*2): Equipment relating to maintenance of public services/functions involving factors of safety refers to.

  - Traffic control systems for air transportation, railways, roads, or marine transportation
  - Equipment for nuclear power generation
  - Equipment related to the above

Notes on using this product

● Handle carefully

Do not drop the equipment or allow it to be subject to strong impact or vibration, as such action may cause malfunctions. Further, do not damage the connection cable, since this may cause wire breakage.

● Environmental operating conditions

Do not use the product in locations where the ambient temperature or humidity exceeds the specifications. Otherwise, image quality may be degraded or internal components may be adversely affected. In particular, do not use the product in areas exposed to direct sunlight. Moreover, during shooting under high temperatures, vertical stripes or white spots (noise) may be produced, depending on the subject or camera conditions (such as increased gain). However, such phenomena are not malfunctions.

● Check a combination with the lens

Depending on the lens and lighting you use, an image is reflected as a ghost in the imaging area. However, this is not because of a fault of the camera.

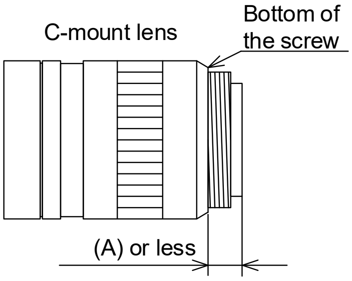
In addition, depending on the lens you use, the performance of the camera may not be brought out fully due to deterioration in resolution and brightness in the peripheral area, aberration and others.

Be sure to check a combination with the camera by using the lens and lightning you actually use.

When installing a lens in the camera, make sure carefully that it is not tilted.

In addition, use a mounting screw free from defects and dirt. Otherwise, the camera may be unable to be removed.

As for the C-mount lens used combining this product, the projection distance from bottom of the screw should use dimensions (A) or less.



(A)	Model
10mm	BU040M, BU160M
9mm	BU040MG, BU040MCG, BU040MCF BU160MG, BU160MCG, BU160MCF

● Mounting to pedestal

When mounting this product to a pedestal, make sure carefully that lens doesn’t touch with the pedestal.

● Do not expose the camera's image-pickup-plane to sunlight or other intense light directly

Its inner CMOS sensor might be damaged.

● Occurrence of moiré

If you shoot thin stripe patterns, moiré patterns (interference fringes) may appear. This is not a malfunction.

● Occurrence of noise on the screen

If an intense magnetic or electromagnetic field is generated near the camera or connection cable, noise may be generated on the screen. If this occurs, move the camera or the cable.

● Handling of the protective cap

If the camera is not in use, attach the lens cap to the camera to protect the image pickup surface.

● If the equipment is not to be used for a long duration

Turn off power to the camera for safety.

● Maintenance

Turn off power to the equipment and wipe it with a dry cloth.

If it becomes severely contaminated, gently wipe the affected areas with a soft cloth dampened with diluted neutral detergent. Never use alcohol, benzene, thinner, or other chemicals because such chemicals may damage or discolor the paint and indications.

If the image pickup surface becomes dusty, contaminated, or scratched, consult your sales representative.

● Disposal

When disposing of the camera, it may be necessary to disassemble it into separate parts, in accordance with the laws and regulations of your country and/or municipality concerning environmental contamination.

Following information is only for EU-member states:
The use of the symbol indicates that this product may not be treated as household waste. By ensuring this product is disposed of correctly, you will help prevent potential negative consequences for the environment and human health, which could otherwise be caused by inappropriate waste handling of this product. For more detailed information about the take-back and recycling of this product, please contact your supplier where you purchased the product.

	“This symbol is applicable for EU member states only”

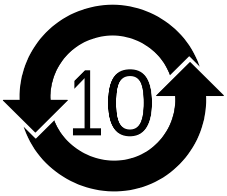
This equipment has been tested and found to comply with the limits for a class A digital device, pursuant to Part 15 of the FCC Rules.
These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.
This equipment generates,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

[Phenomena specific to CMOS sensor]

- **Defective pixels**

  A CMOS image sensor is composed of photo sensor pixels in a square grid array. Due to the characteristics of CMOS image sensors, over- or under-driving of the pixels results in temporary white or black areas (as if these are noises) appearing on the screen. This phenomenon which is not a defect is exacerbated under higher temperatures and long exposure time.
- **Image shading**

  The brightness of the upper part of the screen may be different from that of the lower part. Note that this is a characteristic of a CMOS image sensor and is not a fault.

 中华人民共和国 环保使用期限	环保使用期限标识，是根据电子信息产品污染控制管理办法以及，电子信息产品污染控制标识要求(SJ/T11364-2014)、电子信息产品环保使用期限通则，制定的适用于中国境内销售的电子信息产品的标识。 电子信息产品只要按照安全及使用说明内容，正常使用情况下，从生产月期算起，在此期限内，产品中含有的有毒有害物质不致发生外泄或突变，不致对环境造成严重污染或对其人身、财产造成严重损害。 产品正常使用后，要废弃在环保使用年限内或者刚到年限的产品时，请根据国家标准采取适当的方法进行处置。 另外，此期限不同于质量/功能的保证期限。 The Mark and Information are applicable for People's Republic of China only.

＜产品中有毒有害物质或元素的名称及含量＞

部件名称	有毒有害物质或元素					
	铅（Pb）	汞（Hg）	镉（Cd）	六价铬 （Cr(VI)）	多溴联苯 （PBB）	多溴二苯醚 （PBDE）
相机本体	×	○	○	○	○	○

本表格依据SJ/T 11364的规定编制

○：表示该有毒有害物质在该部件所有均质材料中的含量均在电子信息产品中有毒有害物质的限量要求标准规定的限量要求(GB/T26572)以下

×：表示该有毒有害物质至少在该部件的某一均质材料中的含量超出电子信息产品中有毒有害物质的限量要求标准规定的限量要求(GB/T26572)

This information is applicable for People's Republic of China only.

リサイクルに関する情報（包装物）
有关再利用的信息(包装物)
Information on recycling of wrapping composition

箱 / 箱子 / Box

ペーパーボード
纸板
Paper board

内部緩衝材料・袋
内部缓冲材料・袋
Internal buffer materials・Bag

Installation

Before using this product, you shall install application software to display image and control registers of camera, and IP configuration tool for network setting.

You can download the SDK for our USB camera products (TeliCamSDK) from the Service & Support section of our website.

User registration is necessary to use downloading service. Please make a user registration, or contact your sales representative.

● TOSHIBA TELI CORPORATION Top Page
http://www.toshiba-teli.co.jp/en/

● Service & Support
https://www.toshiba-teli.co.jp/cgi/ss/en/service.cgi

Please refer to the TeliCamSDK startup guide, about Operation environment, Installation, and Setup.

Specifications

Overview

BU040M/BU160M series is an integrated-(one-body)-type camera that adopts a global shutter CMOS sensor. These are BU040M (0.4M type1/2.9) and BU160M (1.6M type1/2.9). For video output and camera control, the USB 3.0 interface standard is adopted for high transfer rate, and it is easy to integrate into industrial equipment.

Features

● High frame rate

Supporting high frame rate, BU040M 437fps (Mono8), BU160M 226fps (Mono8).

● Global shutter

As it employs a global electronic shutter similar to a CCD image sensor, clear images of even fast-moving object are obtainable with less blur.

● USB3.0 interface

Video output and camera control are performed via the USB 3.0 standard interface. Data transfer is up to 5Gbps (Maximum) that enables to output uncompressed video data at high frame rate.

● USB3 Vision

This product is based on USB3 Vision Ver.1.0.

● GenICam Ver.2.4 and Ver.3.0

This product is based on GenICam (Generic Interface for Cameras) Ver.2.4 and Ver.3.0.

● IIDC2 Digital Camera Control Specification Ver.1.1.0

This product is based on IIDC2 Digital Camera Control Specification Ver.1.1.0.

● e-CON Connector adoption

The e-CON connector adoption enables to assemble the cable easily without using special tools.

- Random Trigger Shutter

  The Random Trigger Shutter function provides images in any timing by input of an external trigger signal. Trigger control from PC is available as well.

- Scalable

  Selectable video output area. This mode achieves higher frame rate by reducing vertical output area. And reduces occupied data rate of USB bus by reducing horizontal output area.

- Binning

  Pixel data is combined by vertical and horizontal.

- Decimation

  Camera reads all effective areas at high speed by skipping lines.

- Color processing

  Color models have built in color processing.
  There are RGB, BGR, YUV 4:2:2, YUV 4:1:1, Bayer and Mono output modes.

- Optical glass

  Build-in optical glass models are optional.
  Suffix [G] is attached to the model name of built-in optical glass model. (e.g. BU040MG)
  * Suffix [G] is not shown in the common part of instruction manual.

- IR-cut filter

  Build-in IR-cut filter models are optional for color models.
  Suffix [F] is attached to the model name of built-in IR-cut filter model. (e.g. BU040MCF)
  * Suffix [F] is not shown in the common part of specifications.

- Compact and lightweight

  This camera is compact and lightweight; it is easy to integrate into industrial equipment.

- EU RoHS & Chinese RoHS

Configuration

The system configuration of this camera series is as follows;
This camera has no accessories, please prepare other equipment separately.

● Camera: This product.

● Camera mounting kit
CPTBU, CPTBUBG (*1): To fix a camera to a tripod; attach this to the bottom of the camera.

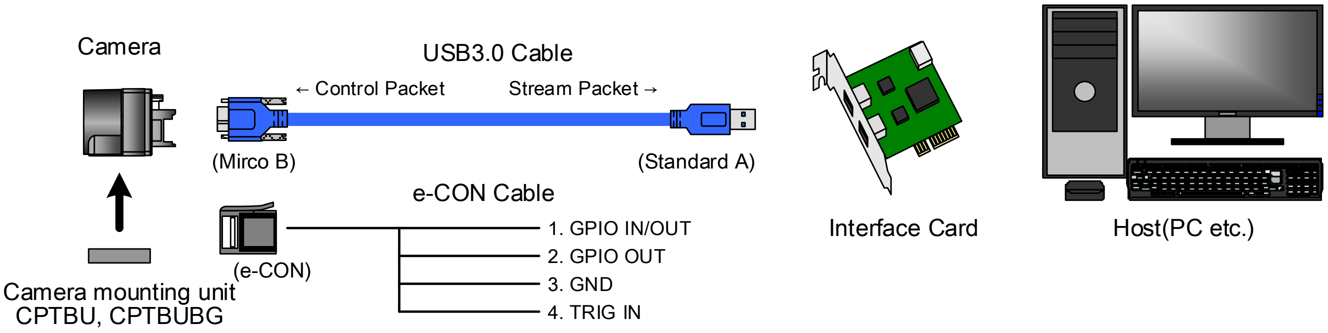
● USB3.0 Cable (*2): This cable is used to connect the camera to host PC. Please use a USB3.0 cable of Standard A - Micro B. This product is able to connect a USB cable equipped with screw lock mechanism. Please use it as needed.

● USB3.0 Interface Card (*2): This is the interface card to connect to the camera. Usually this card is installed to expansion slot of PC etc.

● e-CON Cable. (*2): This cable is used to input external trigger signal and output GPIO signal.
We recommend using shielded cable, because there is likely to be affected by the noise depending on the operating environment of the camera.

*1: Optional part. Contact your sales representative for details of option units.
*2: Commercial items.

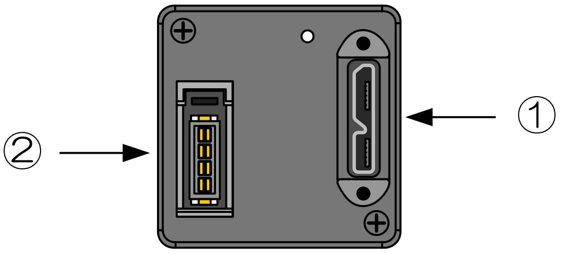
Connection



Notes on Connection:

- Please confirm the power supply of the camera off when plugging in or pulling out the I/O Connector. It causes the breakdown.
- If your camera is used in a system where its connectors are subjected to strong repetitive shocks, its connectors are possible to break down. If you use your camera in such a situation, use an USB3.0 cable with a lock screw, and secure the camera cable as close as possible to the camera body for avoid physical shock to the camera connector.
- About e-CON cable: In the case that electric-wire is long or thin, input and output voltage may not satisfy specifications of the camera or your system by voltage drop. Please confirm wires' specifications before use them.
- Lost packets may occur by an electrical characteristic of the transmission line of USB3.0. (USB3.0 Interface Card, USB3.0 Cable, and USB3.0 HUB).

Connector Pin Assignment



Rear View

1. USB3.0 Interface Connector

Connector model: WMUR-10F6L1PH5N (WIN WIN PRECISION INDUSTRIAL)

Pin No.	I/O	Signal	Function
1	-	VBUS	Power
2	I/O	D-	USB2.0 differential pair
3	I/O	D+	
4	-	NC	Not connected
5	-	GND	Ground for power return
6	O	SSTX-	SuperSpeed transmitter differential pair
7	O	SSTX+	
8	-	GND_DRAIN	Ground for SuperSpeed signal return
9	I	SSRX-	SuperSpeed receiver differential pair
10	I	SSRX+	

2. I/O Connector

Connector (Camera side)	37204-62B3-004PL (3M Japan Limited) or equivalent
Matching connector (Cable side)	Connectors which conformed to e-CON e.g. 37104 series (3M Japan Limited), RITS 4P series (TE Connectivity Ltd.)

* Matching connector is not an accessory of this product.

Pin assignment

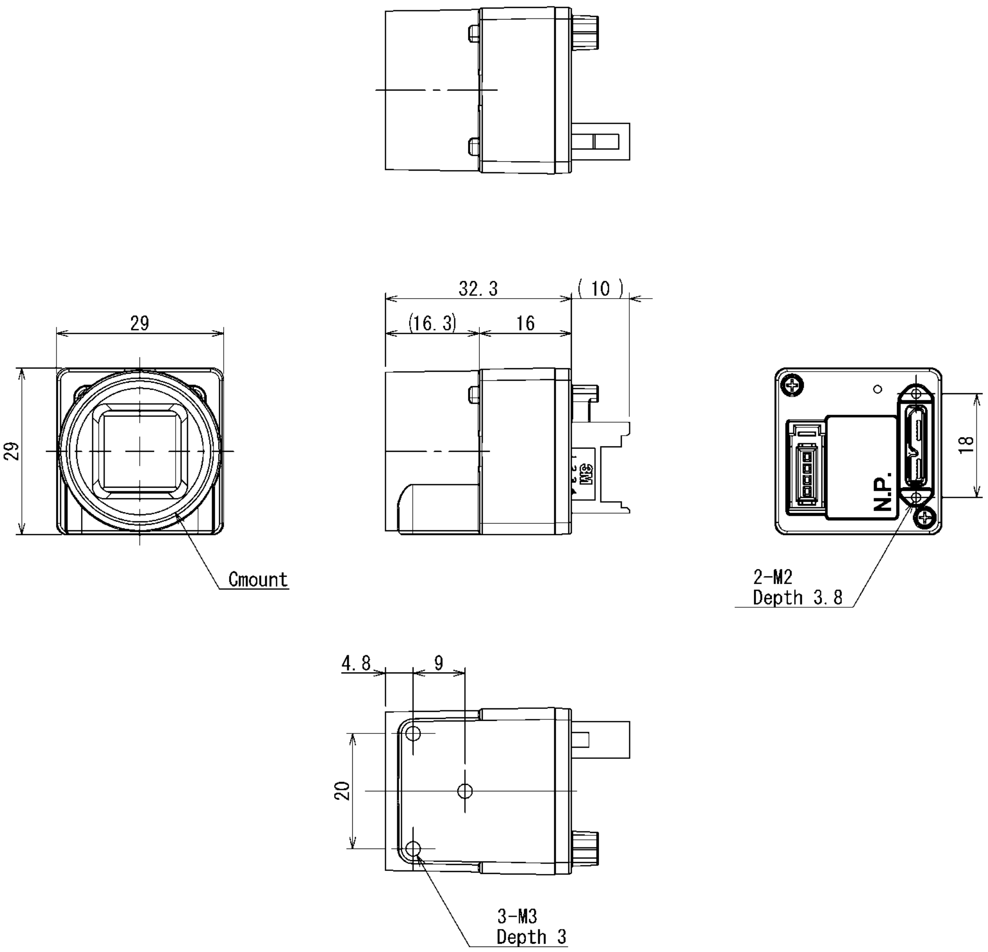
①
② ↑ TOP
③
④

*Above figure is connector view from insert side.

Pin No.	I/O	Signal	Function
1	I/O	Line2	GPIO Input / Output
2	O	Line1	GPIO Output
3	-	GND	Ground
4	I	Line0	GPIO Input

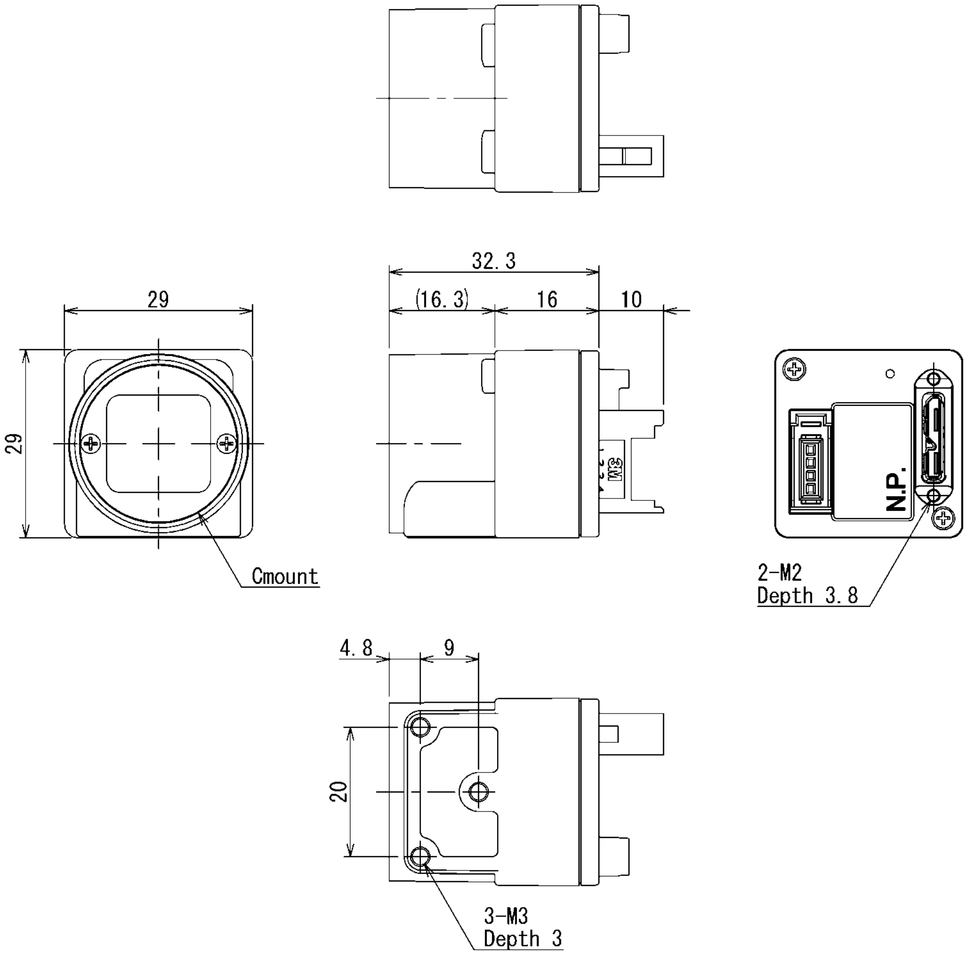
Outline Drawing

<BU040M/BU160M>



Specification
  Main material : Aluminum die-cast metal
  Processing   : Cation coating (black)

<BU040MG/BU040MCG/BU040MCF/BU160MG/BU160MCG/BU160MCF>



Specification
  Main material : Aluminum die-cast metal
  Processing   : Cation coating (black)

General Specifications

Model Name		
Without optical glass	BU040M	BU160M
With optical glass	BU040MG	BU160MG
Imager	CMOS image sensor	
Number of Video out pixels (H) × (V)	720 x 540	1440 x 1080
Optical Size	type1/2.9	
Scanning area (H) × (V)[mm]	5.02 x 3.82	5.00 x 3.75
Pixel size (H) × (V)[μm]	6.90 × 6.90	3.45 × 3.45
Scan method	Progressive	
Electronic shutter method	Global shutter	
Aspect ratio	4 : 3	
Sensitivity	2800lx, F5.6, 1/500s	3000lx, F4, 1/250s
Minimum illuminance (*1)	6lx	12lx
Power supply	DC +5V±5% (from USB connector)	
Power consumption (*2)	2.2W (maximum)	2.4W (maximum)
Interface	USB 3.0 (Only SuperSpeed is supported)	
Transmission speed	5Gbps (maximum)	
Protocol	USB3 Vision	
Image format	Mono8, Mono10, Mono12	
Maximum Frame rate (*2)		
Mono8	437 fps	226 fps
Mono10, Mono12	320 fps	128 fps
Dimensions	29 mm(W) x 29 mm (H) x 16 mm (D) (Not including protrusion)	
Mass	Approximately 33g	
Lens mount	C-mount	
Flange back	17.526mm	
Camera body grounding: insulation status	Conductive between circuit GND and camera body	

*1 F1.4, Gain +24dB, Video Level 50%

*2 at the all pixel readout

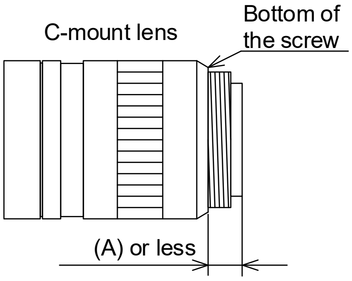
Model Name		
With Optical glass	BU040MCG	BU160MCG
With IR-cut filter	BU040MCF	BU160MCF
Imager	CMOS image sensor	
Number of Video out pixels (H) × (V)	720 x 540	1440 x 1080
Optical Size	type1/2.9	
Scanning area (H) × (V)[mm]	5.02 x 3.82	5.00 x 3.75
Pixel size (H) × (V)[μm]	6.90 × 6.90	3.45 × 3.45
Scan method	Progressive	
Electronic shutter method	Global shutter	
Aspect ratio	4 : 3	
Sensitivity		
With Optical glass	2300lx, F5.6, 1/500s	2150lx, F2.8, 1/250s
With IR-cut filter	2500lx, F5.6, 1/500s	2350lx, F2.8, 1/250s
Minimum illuminance (*1)		
	1lx	1lx
	3lx	3lx
Power supply	DC +5V±5% (from USB connector)	
Power consumption (*2)	3.2W (maximum)	3.4W (maximum)
Interface	USB 3.0 (Only SuperSpeed is supported)	
Transmission speed	5Gbps (maximum)	
Protocol	USB3 Vision	
Image format	RGB, BGR, YUV4:2:2, YUV4:1:1, Bayer8, Bayer10, Bayer12, Mono8	
Maximum Frame rate (*2)		
Bayer8, Mono8 (HighFramerateMode=ON)	436 fps (523 fps)	227 fps (240 fps)
YUV4:1:1	320 fps	162 fps
YUV4:2:2	320 fps	122 fps
Bayer10, Bayer12	320 fps	122 fps
RGB, BGR	320 fps	81 fps
Dimensions	29 mm(W) x 29 mm (H) x 16 mm (D) (Not including protrusion)	
Mass	Approximately 33g	
Lens mount	C-mount	
Flange back	17.526mm	
Camera body grounding: insulation status	Conductive between circuit GND and camera body	

*1 F1.4, Gain +36dB, Video Level 50%

*2 at the all pixel readout

Notes on combination of C-mount lens:

- Depending on the lens you use, the performance of the camera may not be brought out fully due to the deterioration in resolution and brightness in the peripheral area, occurrence of a ghost, aberration and others. When you check the combination between the lens and camera, be sure to use the lens you actually use.
- In addition, use a mounting screw free from defects and dirt. Otherwise, the camera may be unable to be removed.
- As for the C-mount lens used combining this camera, the projection distance from bottom of the screw should use dimensions (A) or less.



(A)	Model
10mm	BU040M, BU160M
9mm	BU040MG, BU040MCG, BU040MCF BU160MG, BU160MCG, BU160MCF

LED Status

Camera state	Lamp indication
No power	Off
Link detection in progress	Fast flash green (ON:20ms, OFF:60ms)
Connection Error	Flash alternate red / green
SuperSpeed connected, but no data being transferred	Flash green (ON: 200ms, OFF: 800ms)
SuperSpeed connected, waiting for trigger	Flash orange (ON: 200ms, OFF: 800ms)
Data being transferred	Fast flash green (ON:60ms, OFF:20ms)
Error during data transfer	Solid Red (Time period: 500ms)
Stand-by	Super slow flash orange (ON:200ms, OFF: 2800ms)

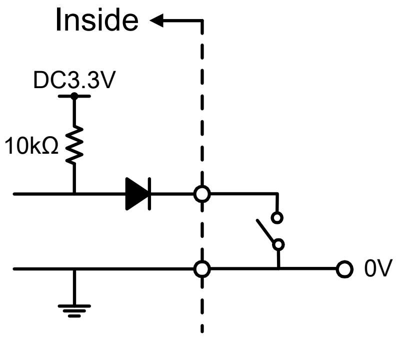
I/O Specification

● Signal Specification

- Line0 (I/O connector : 4 pin)

Input / Output Specification	: Input only
Input Circuit	: LVTTL
Level	: Low 0 ~ 0.5V, High 2.0 ~ 24.0V
Polarity	: High active / Low active (initial factory setting: Low active)
Pulse Width	: Minimum 50μs

Input circuit diagram



Notes of external trigger signal:

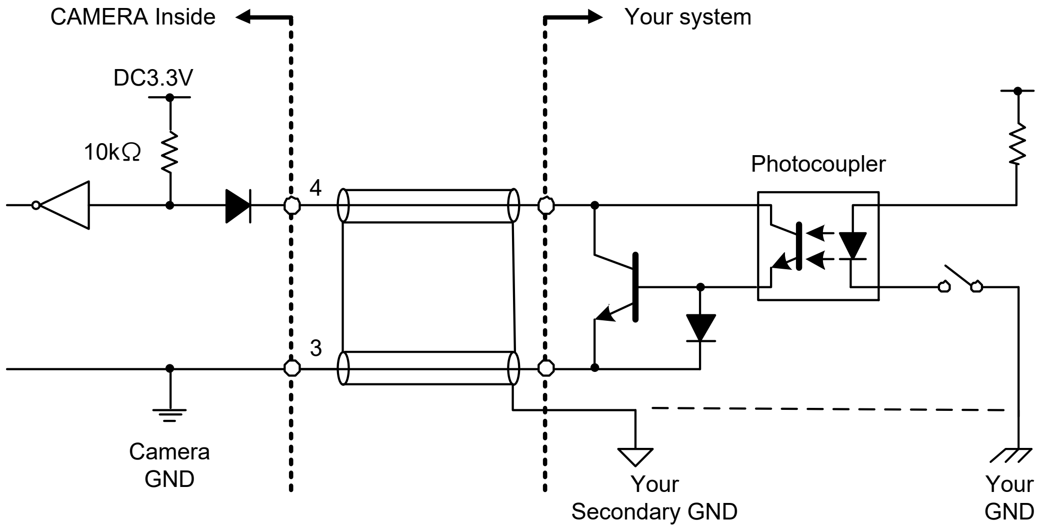
Depending on cable length, cable kinds and input current of trigger input line, Random Trigger Shutter operation may not satisfy timing specification or camera may not receive EXT_TRIG signal. Please confirm it before use.

Notes of input level:

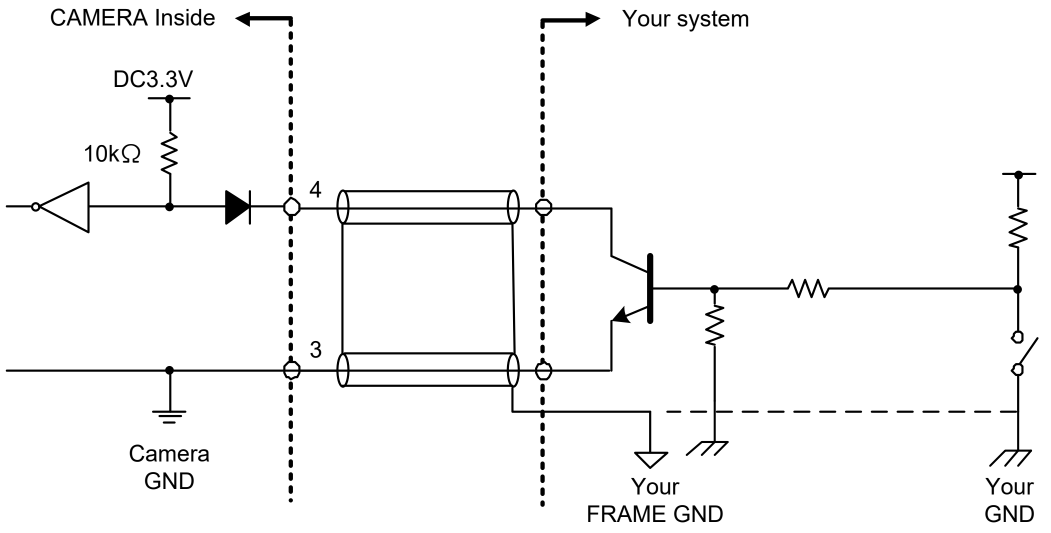
Line0 and Line2 have different input level. Please use input level within the voltage described in this specification.

- External trigger input recommended circuit

- Isolated I/F



- Non-Isolated I/F



Notes of trigger input cable:

- The recognition of the trigger signal depends on the length, characteristic or driving current of the cable. Therefore please confirm your system about those conditions.
- Pin 3 is conducted with camera frame.
  Using shield cable, terminal processing of the shield is referred as above.
- Please confirm the EMC adaptability in whole of your system.

- Line2 (I/O connector : 1 pin)

Input / Output specification : Input / Output (switch by LineMode is possible)
Factory setting : Output

- Input signal specification

Level : Low 0 ~ 0.5V, High 4.0 ~ 5.0V
Polarity : High active / Low active (initial factory setting: Low active)
Pulse Width : Minimum 50µs

Notes of external trigger signal:

Depending on cable length, cable kinds and input current of trigger input line, Random Trigger Shutter operation may not satisfy timing specification or camera may not receive EXT_TRIG signal. Please confirm it before use.

Notes of input level:

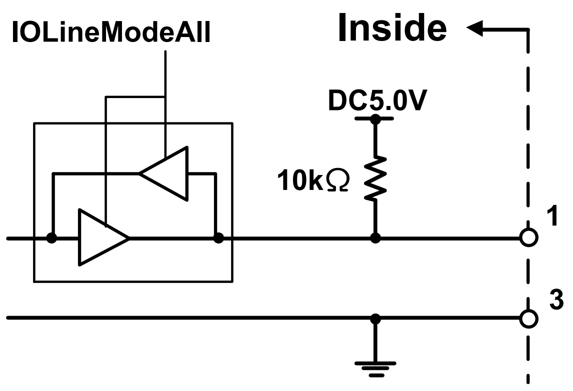
Line0 and Line2 have different input level. Please use input level within the voltage described in this specification.

- Output signal specification

Output Circuit : 5V CMOS
Maximum Current : +/-32mA
Polarity : High active / Low active (initial factory setting: Low active)
Signal Source (LineSource) : Off* / UserOutput / Timer0Active / AcquisitionActive
FrameTriggerWait / FrameActive / FrameTransferActive
ExposureActive

*Even if LineMode is set as Output, the input signal to Line2 is valid when LineSource is set as Off.

Input / Output circuit diagram



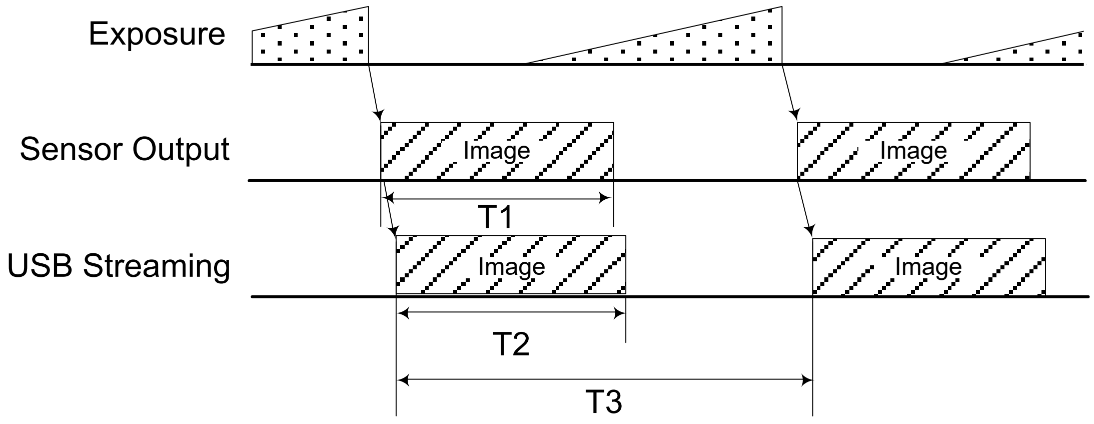
- Line1 (I/O connector : 2 pin)

Input / Output Specification	: Output only
Output Circuit	: 5V CMOS
Maximum Current	: +/-32mA
Polarity	: High active / Low active (initial factory setting: Low active)
Signal Source (LineSource)	: Off / UserOutput / Timer0Active / AcquisitionActive FrameTriggerWait / FrameActive / FrameTransferActive ExposureActive

Timing Specification

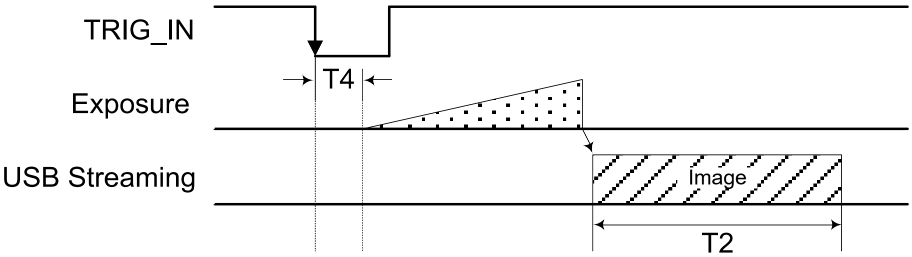
Image data outputs are transferred with USB bulk transfer. Timing numerical value below is described by absolute prerequisite that camera can use transmission band without restriction of other device. When there is other device on the same bus, the value described below is not guaranteed.

● In Manual Shutter mode

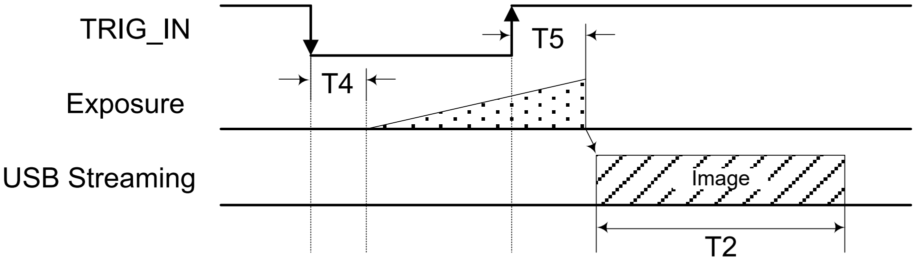


Model Name	format	T1 [ms]	T2 [ms]	T3 [s]
BU040M	Mono8	2.3	2.3	1/(Frame Rate setting)
	Mono10, Mono12	3.1	3.1	
BU040MCG	Bayer8, Mono8	2.3	2.3	
	YUV 4:1:1	3.1	3.1	
	YUV 4:2:2			
	Bayer10, Bayer12			
	RGB, BGR			
BU160M	Mono8	4.4	4.4	
	Mono10, Mono12	6.0	7.8	
BU160MCG	Bayer8, Mono8	4.4	4.4	
	YUV 4:1:1	6.0	6.1	
	YUV 4:2:2		8.2	
	Bayer10, Bayer12		8.2	
	RGB, BGR		12.3	

● In Random Trigger Shutter mode



Edge mode / Bulk mode (at all pixels readout)



Level mode (at all pixels readout)

Model Name	format	T4 [μs]	T5 [μs]
BU040M	Mono8	11.6	25.9
	Mono10, Mono12	15.9	30.1
BU040MCG	Bayer8, Mono8	11.6	25.9
	YUV 4:1:1	15.9	30.1
	YUV 4:2:2		
	Bayer10, Bayer12		
	RGB, BGR		
BU160M	Mono8	11.7	25.9
	Mono10, Mono12	15.9	30.2
BU160MCG	Bayer8, Mono8	11.6	25.9
	YUV 4:1:1	15.9	30.1
	YUV 4:2:2		
	Bayer10, Bayer12		
	RGB, BGR		

* The value of T2 is the same as the value of normal shutter mode.

* T4 and T5 are typical value.

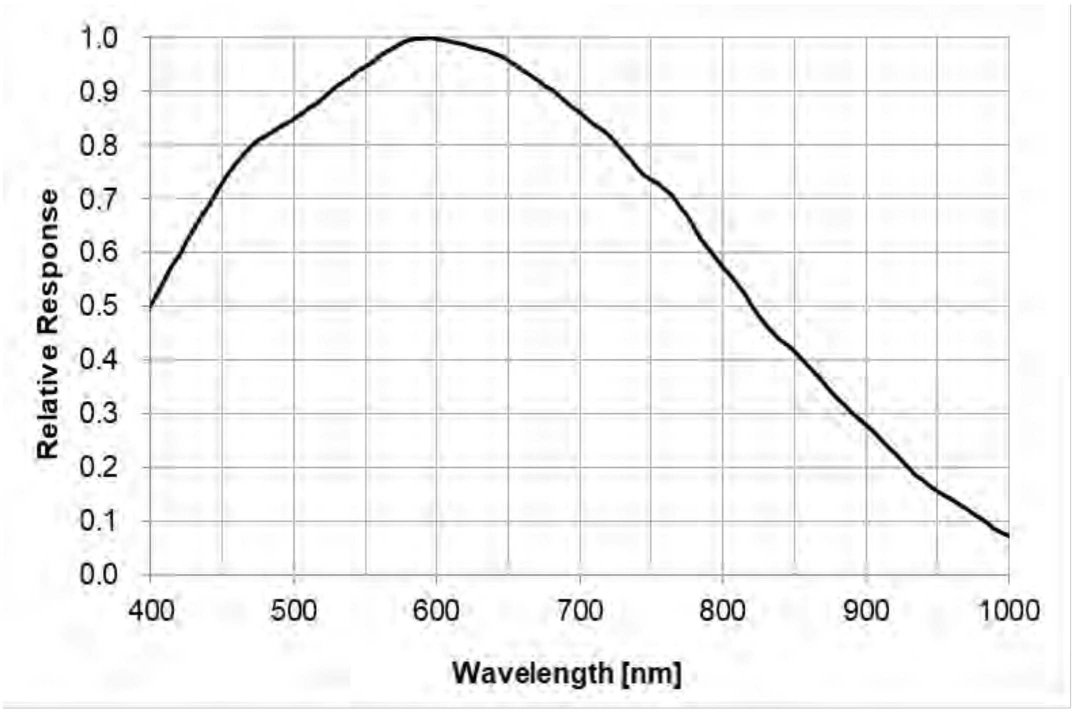
Notes of random trigger shutter mode:

- In the period when FRAME_TRIGGER_WAIT (GPIO signal) is inactive, user must not input external trigger signal to this camera.
- When the interval of the input trigger signal is extremely short, or when the trigger signal is noisy, there is a possibility of causing the malfunction. In this case, please input a proper trigger signal.

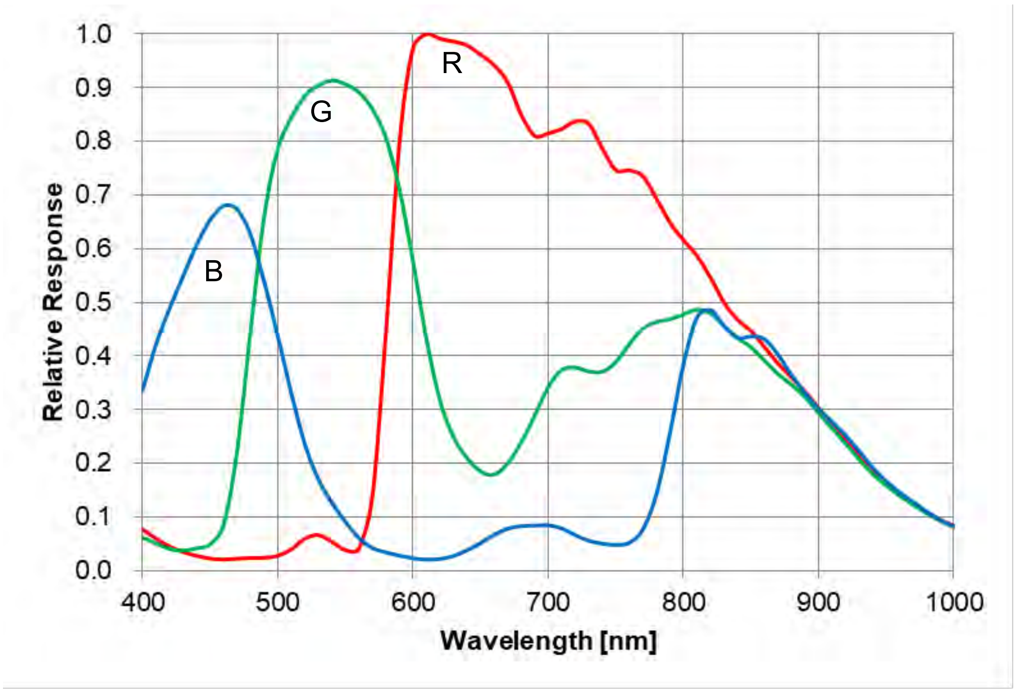
Typical Spectral Response

* The lens characteristics and light source characteristics is not reflected in table.

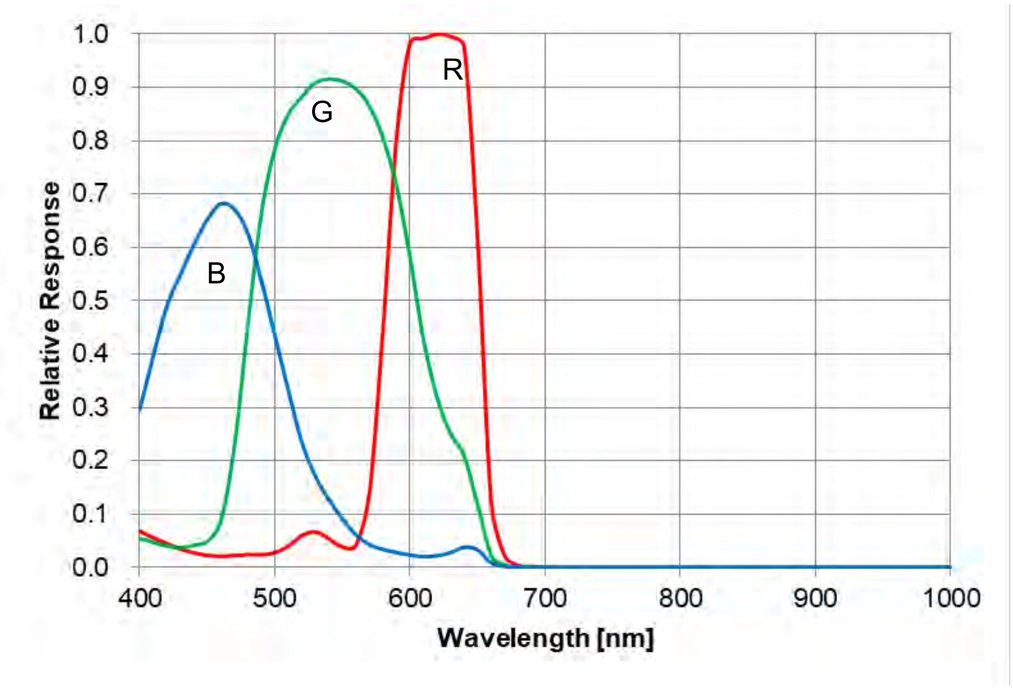
● BU040M/BU160M



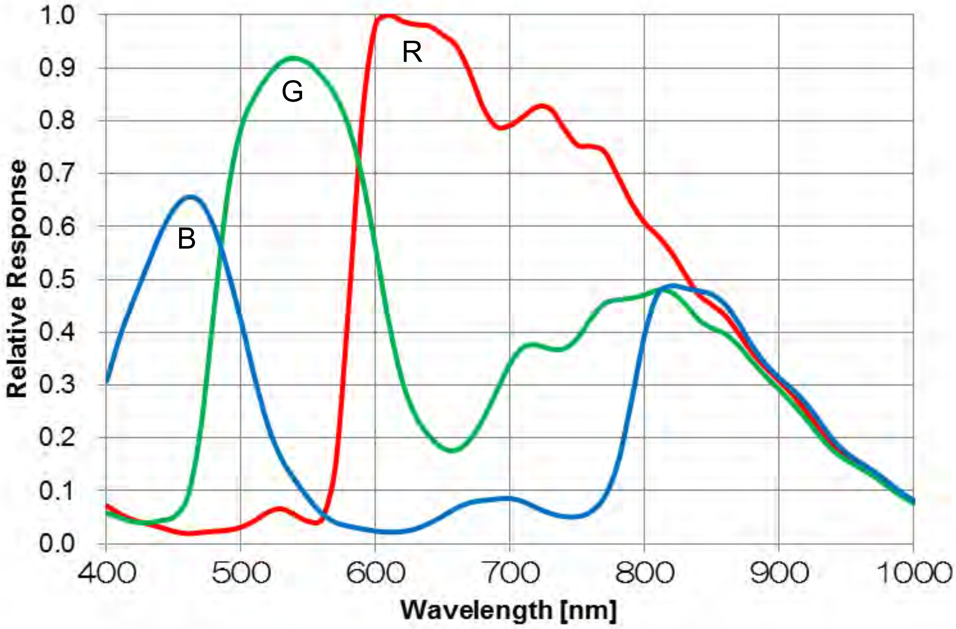
● BU040MCG



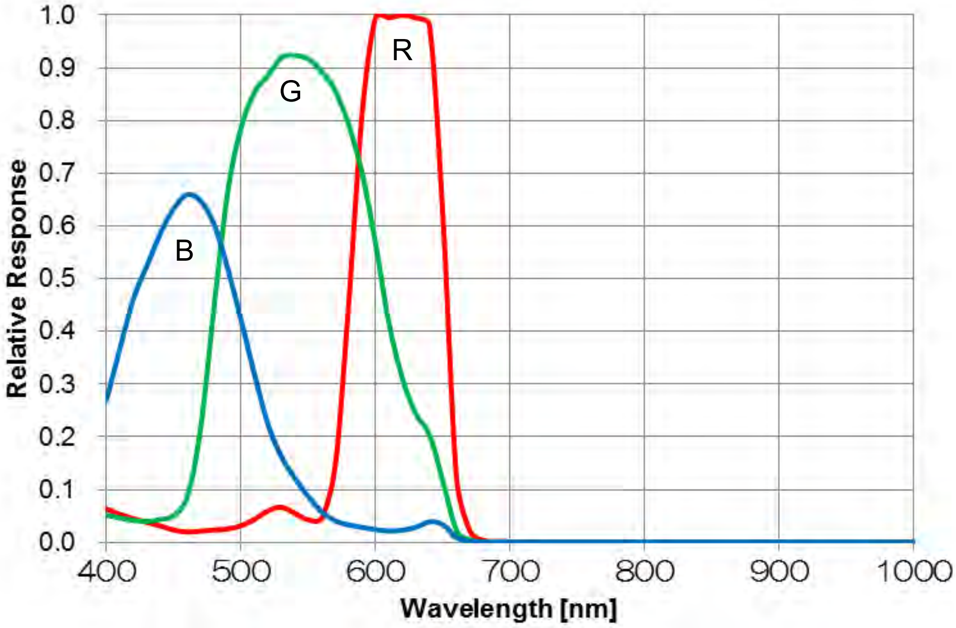
● BU040MCF



- BU160MCG



- BU160MCF



Operating Ambient Conditions

- Ambient conditions
  - Operating Assurance

Temperature:	0°C to +40°C, Camera housing temperature: less than 60 °C
Humidity:	10% to 90% (no condensation)

  - Storage Assurance

Temperature:	-20°C to +60°C
Humidity:	90% or less (no condensation)

Notes on Heat Radiation:

The temperature of camera housing must be kept less than 60 °C.

However, about the upper limit of top surface temperature of camera housing and the allowed ambient temperature of each model, please refer to the "Thermal design manual" on our HP.

- EMC Conditions

- EMI (Electro-Magnetic Interference):	EN61000-6-4
	FCC Part 15 Subpart B Class A
- EMS (Electro-Magnetic Susceptibility):	EN61000-6-2

Notes on Conformity of the EMC:

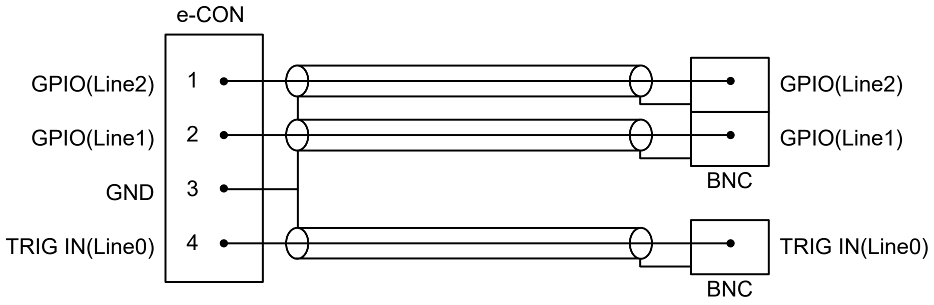
The adaptability of the safety standard of this camera is assured in the condition of combination with the following parts:

- USB Cable	USB3-KR1-A-MBS-030	(OKI Electric Cable Co., Ltd.)
- e-CON Cable	3.0m, Shield cable	(Fabricated parts)

Parts:

- e-CON connector	37104-3163-000 FL	(3M Japan Limited)
- Shielded wire	UL1533 (AWG28)	(Hitachi cable, Ltd.)

Connection:



Please confirm the EMC adaptability when it combines with parts other than them.

Functions

This section introduces standard functions. BU040M / BU160M series provides following functions.

Category	Function	
USB3 Vision	Bootstrap Registers	USB3 Vision standard registers
DeviceControl	DeviceControl	Device information
ImageFormatControl	ImageFormatSelector	Image format selection
	Scalable	Scalable control
	Binning	Binning control
	Decimation	Decimation control
	Reverse	Image flip
	PixelFormat	Pixel format selection
	TestPattern	Test pattern control
AcquisitionControl	AcquisitionControl	Image stream start / stop
	ImageBuffer	Image buffer control
	TriggerControl	Trigger control
	ExposureControl	Exposure time control
DigitalIOControl	DigitalIOControl	GPIO signal control
	AntiGlitch	AntiGlitch control
	AntiChattering	AntiChattering control
CounterAndTimerControl	TimerControl	Timer0Active signal control
AnalogControl	Gain	Gain control
	BlackLevel	Black level control
	Gamma	Gamma correction
	Sharpness	Sharpness control
	BalanceRatio	Color gain (R, B)
	BalanceWhiteAuto	Execute auto white balance once
	ColorCorrectionMatrix	Color matrix correction
LUTControl	LUTControl	LUT control
UserSetControl	UserSetControl	Load / Save user setting
EventControl	EventControl	Event packet control
VenderUniqueControl	FrameSynchronization	Frame synchronization control
	LEDIndicatorLuminance	LED luminance control
DPCControl	DPCControl	Defect pixel correction control
SequentialShutterControl	SequentialShutterControl	Sequential shutter control
ChunkDataControl	Chunk	ChunkDataControl

Features supported by each model are as follows.

Function	BU040M	BU160M	BU040MCG	BU160MCG
Bootstrap Registers	✓	✓	✓	✓
DeviceControl	✓	✓	✓	✓
ImageFormatSelector	✓	✓	✓	✓
Scalable	✓	✓	✓	✓
Binning	✓	✓	✓	✓
Decimation	✓	✓	✓	✓
Reverse	✓	✓	✓	✓
PixelFormat	✓	✓	✓	✓
TestPattern	✓	✓	✓	✓
AcquisitionControl	✓	✓	✓	✓
ImageBuffer	✓	✓	✓	✓
TriggerControl	✓	✓	✓	✓
ExposureControl	✓	✓	✓	✓
DigitalIOControl	✓	✓	✓	✓
AntiGlitch	✓	✓	✓	✓
AntiChattering	✓	✓	✓	✓
TimerControl	✓	✓	✓	✓
Gain	✓	✓	✓	✓
BlackLevel	✓	✓	✓	✓
Gamma	✓	✓	✓	✓
Sharpness	-	-	✓	✓
BalanceRatio	-	-	✓	✓
BalanceWhiteAuto	-	-	✓	✓
ColorCorrectionMatrix	-	-	✓	✓
LUTControl	✓	✓	✓	✓
UserSetControl	✓	✓	✓	✓
EventControl	✓	✓	✓	✓
FrameSynchronization	✓	✓	✓	✓
LEDIndicatorLuminance	✓	✓	✓	✓
DPCControl	✓	✓	✓	✓
SequentialShutterControl	✓	✓	✓	✓
Chunk	✓	✓	✓	✓

Details of each feature are described in following pages.

Bootstrap Registers

This camera is based on USB3 Vision.
Please refer to USB3 Vision specification for details about Bootstrap Registers defined in USB3 Vision.
AIA (Automated Imaging Association) USB3 Vision Homepage.
http://www.visiononline.org/vision-standards-details.cfm?type=11

Followings are commonly used registers.

- **UserDefinedName**
  UserDefinedName is used for assigning unique ID to the camera.
  You can store an arbitrary string into non-volatile memory.
- **StreamEnable**
  When **StreamEnable** bit is set. Camera starts to send image streaming.
  TeliU3vSDK / TeliCamSDK require some setups on your application in opening and closing the stream channel. Please refer to the library manual for details.
- **EventEnable**
  When **EventEnable** bit is set. Camera is enabled to send event packets.
  TeliU3vSDK / TeliCamSDK require some setups on your application in opening and closing the event channel. Please refer to the library manual for details.

● Registers

USB3 Vision ABRM					
Register	Address	GenICam Interface	Length Byte / [bit]	Access	Description
ManufactureName	0x00004	String	64	R	Manufacturer name
ModelName	0x00044	String	64	R	Device model name
FamilyName	0x00084	String	64	R	Device family name
DeviceVersion	0x000C4	String	64	R	Device version
ManufacturerInfo	0x00104	String	64	R	Additional manufacturer specific information
SerialNumber	0x00144	String	64	R	Serial number of the device
UserDefinedName	0x00184	String	64	R/W	User defined name of the device.
SBRM Address	0x001D8	Integer	8	R	Start Address of the SBRM

USB3 Vision SBRM

Register	Address	GenICam Interface	Length Byte / [bit]	Access	Description
SIRMAddress	0x10020	Integer	8	R	Start Address of the SIRM
EIRMAddress	0x1002C	Integer	8	R	Start Address of the EIRM
IIDC2Address	0x10038	Integer	8	R	Start Address of the IIDC2
CurrentSpeed	0x10040	Integer	[3..0]	R	[0]: Low-Speed connection (not supported) [1]: Full-Speed connection (not supported) [2]: High-Speed connection [3]: Super-Speed connection

USB3 Vision SIRM

Register	Address	GenICam Interface	Length Byte / [bit]	Access	Description
StreamEnable	0x20004	Integer	[0]	R/W	0: Disable data transfer of the streaming 1: Enable data transfer of the streaming
SIRequiredPayloadSize	0x20008	Integer	8	R	Minimum required payload size with current settings
SIRequiredLeaderSize	0x20010	Integer	4	R	Minimum required leader size
SIRequiredTrailerSize	0x20014	Integer	4	R	Minimum required trailer size
SIMaximumLeaderSize	0x20018	Integer	4	R	Maximum leader size
SIPayloadTransferSize	0x2001C	Integer	4	R	Expected Size of a single Payload Transfer
SIPayloadTransferCount	0x20020	Integer	4	R	Expected Number of Payload Transfers
SIPayloadFinalTransfer1Size	0x20024	Integer	4	R	Size of first final Payload transfer
SIPayloadFinalTransfer2Size	0x20028	Integer	4	R	Size of second final Payload transfer
SIMaximumTrailerSize	0x2002C	Integer	4	R	Maximum trailer size

USB3 Vision EIRM

Register	Address	GenICam Interface	Length Byte / [bit]	Access	Description
EventEnable	0x30000	Integer	[0]	R/W	0: Disable data transfer of the event 1: Enable data transfer of the event

● Note

BU series doesn’t support image transfer in High-Speed connection.

CurrentSpeed register could be used for showing the warning message on your application when camera is connected to USB2.0 port.

DeviceControl

Registers of this category provide various information of the camera.

● Registers

Register	Address	GenICam Interface	Length Byte / [bit]	Access	Description
DeviceReset	0x20003C	Command	[0]	W	[1] Resets the device.
DeviceVendorName	0x200070	String	16	R	Same as ManufactureName in Bootstrap Registers
DeviceModelName	0x200090	String	16	R	Same as ModelName in Bootstrap Registers
DeviceManufactureInfo	0x2000B0	String	16	R	Same as ManufacturerInfo in Bootstrap Registers
DeviceVersion	0x2000D0	String	16	R	Same as DeviceVersion in Bootstrap Registers
DeviceID	0x200110	String	16	R	Same as SerialNumber in Bootstrap Registers

● Note

– **DeviceReset**

Camera executes the USB Bus reset operation by DeviceReset command.

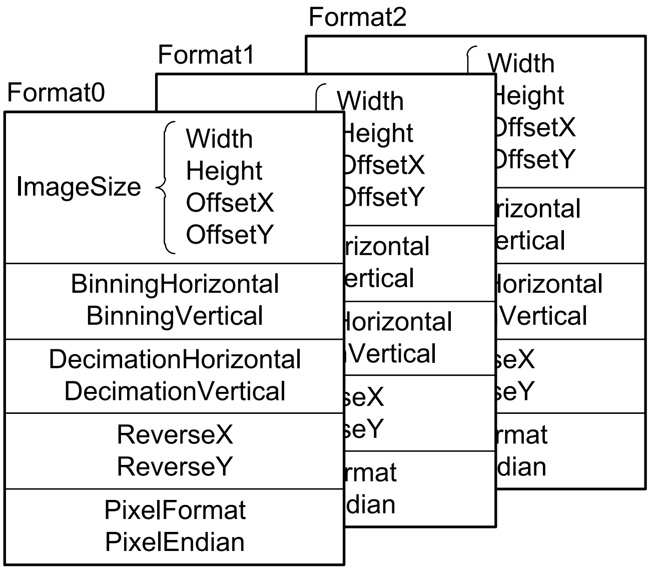
- Plug-and-play happens, camera handles used in the application become invalid.

- The application is required to close and re-open the camera.

ImageFormatControl

Registers of this category are related to image format control.

Camera has three different banks of image format. You can select image format by ImageFormatSelector.



● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ImageFormatSelector	IEnumeration	4	R/W	Selects an image format.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ImageFormatSelector	Implemented	0x202020	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20202C	16	R	[0] : Format0 [1] : Format1 [2] : Format2
	Value	0x20203C	4	R/W	Selects an image format.

List of registers to be applied by ImageFormatSelector

ImageFormat 0 - 2
Width
Height
OffsetX
OffsetY
BinningHorizontal
BinningVertical
DecimationHorizontal
DecimationVertical
ReverseX
ReverseY
PixelFormat
PixelEndian

● Control with TeliCamSDK

Camera feature API

Control ImageFormat using dedicated API.

API name	Description
GetCamImageFormatSelector	Get current ImageFormatSelector value
SetCamImageFormatSelector	Set new ImageFormatSelector value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control ImageFormat using GenICam API.

◆ImageFormat

Select an image format by ‘ImageFormatSelector’.

Integer value and string value of Enumeration are as follows.

Integer	String
0(*)	Format0
1	Format1
2	Format2

* initial factory setting

```
// GenICam node handle
CAM_NODE_HANDLE    hSelector = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "ImageFormatSelector", &hSelector);

// ImageFormat = Format2
Nd_SetEnumStrValue(s_hCam, hSelector, "Format2");
```

Please refer to [INode functions], [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Gain by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆ImageFormat

Write to ‘Value’ field of ‘ImageFormatSelector’ register.

```
// ImageFormat = Format2
uint32_t       uiSelector;
uiSelector = 2;
Cam_WriteReg(s_hCam, 0x20203C, 1, &uiSelector);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

Changing ‘ImageFormatSelector’ register value is invalid during image stream data output.

Scalable

Scalable function reads out the region of interest (ROI) of the sensor.

If height size is set small, it is possible to increase the frame rate.

Only single rectangle is selectable. Concave or convex shape is not selectable.

- Window size: {A + 4 × m (H)} × {B + 2 × n (V)}

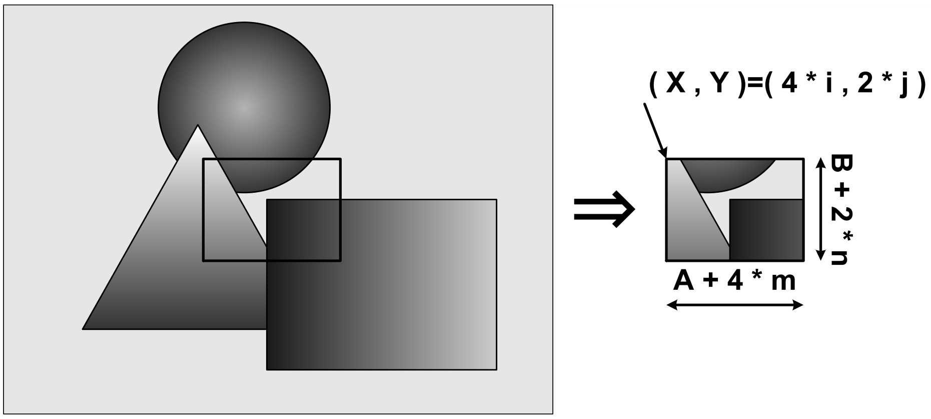
  A, B = minimum unit size

  m, n = integer

  The window size is equal or less than maximum image size.
- Start address: {4 x i (H)} x {2 x j (V)}

  i, j = integer

  The window size is equal or less than maximum image size.



Scalable

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
Width	IInteger	4	R/W	Sets width (in pixels) of the image data.
Height	IInteger	4	R/W	Sets Height (in pixels) of the image data.
OffsetX	IInteger	4	R/W	Sets horizontal offset (in pixels) from the origin to the region of interest.
OffsetY	IInteger	4	R/W	Sets vertical offset (in pixels) from the origin to the region of interest.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ImageSize	Implemented	0x202060	[31]	R	Returns the state whether the function is implemented.
	OffsetXMin	0x20206C	4	R	Returns the minimum starting position of the horizontal direction
	OffsetXInc	0x202070	4	R	Returns the unit starting position of the horizontal direction
	WidthMin	0x202074	4	R	Returns the minimum size of width
	WidthInc	0x202078	4	R	Returns the unit size of width
	SensorWidth	0x20207C	4	R	Returns effective width of the sensor in pixels
	OffsetYMin	0x202080	4	R	Returns the minimum starting position of the vertical direction
	OffsetYInc	0x202084	4	R	Returns the unit starting position of the vertical direction
	HeightMin	0x202088	4	R	Returns the minimum size of height
	HeightInc	0x20208C	4	R	Returns the unit size of height.
	SensorHeight	0x202090	4	R	Returns effective Height of the sensor in pixels
	OffsetX	0x202094	4	RW	Sets horizontal offset (in pixels) from the origin to the region of interest
	Width	0x202098	4	RW	Sets width (in pixels) of the image data.
	OffsetY	0x20209C	4	RW	Sets vertical offset (in pixels) from the origin to the region of interest
	Height	0x2020A0	4	RW	Sets Height (in pixels) of the image data.

● Control with TeliCamSDK

Camera feature API

Control Scalable using dedicated API.

API name	Description
GetCamSensorWidth	Get effective width of the sensor in pixels
GetCamSensorHeight	Get effective height of the sensor in pixels
GetCamRoi	Get ROI (Region of Interest) of the camera
SetCamRoi	Set ROI (Region of Interest) to the camera
GetCamWidthMinMax	Get minimum and maximum value
GetCamWidth	Get width of image
SetCamWidth	Set width of image
GetCamHeightMinMax	Get minimum and maximum value
GetCamHeight	Get height of image
SetCamHeight	Set height of image
GetCamOffsetXMinMax	Get minimum and maximum value
GetCamOffsetX	Get horizontal offset of image
SetCamOffsetX	Set horizontal offset of image
GetCamOffsetYMinMax	Get minimum and maximum value
GetCamOffsetY	Get vertical offset of image
SetCamOffsetY	Set vertical offset of image

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Scalable using GenICam API.

◆Scalable

```
// GenICam node handle
CAM_NODE_HANDLE    hWidth = NULL;
CAM_NODE_HANDLE    hHeight = NULL;
CAM_NODE_HANDLE    hOffsetX = NULL;
CAM_NODE_HANDLE    hOffsetY = NULL;

// ROI = {OffsetX, Width, OffsetY, Height};
uint64_t       ROI[] = {360, 720, 270, 540};

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “Width”, &hWidth);
Nd_GetNode(s_hCam, “Height”, &hHeight);
Nd_GetNode(s_hCam, “OffsetX”, &hOffsetX);
Nd_GetNode(s_hCam, “OffsetY”, &hOffsetY);

// Set ROI
Nd_SetIntValue(s_hCam, hWidth, ROI[1]);
Nd_SetIntValue(s_hCam, hOffsetX, ROI[0]);
Nd_SetIntValue(s_hCam, hHeight, ROI[3]);
Nd_SetIntValue(s_hCam, hOffsetY, ROI[2]);
```

If you’re going to reduce width size, set Width, first. Then set OffsetX.

If you’re going to increase width size, set OffsetX, first. Then set Width.

If you’re going to reduce height size, set Height, first. Then set OffsetY.

If you’re going to increase height size, set OffsetY, first. Then set Height.

Please refer to [INode functions] and [IInteger node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Scalable by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Scalable

Write to ‘Value’ field of ‘OffsetX’, ’Width’, ‘OffsetY’, ‘Height’ register.

```
// ROI = {OffsetX, Width, OffsetY, Height};
uint32_t       ROI[] = {360, 720, 270, 540};

// Set ROI (in one by one)
Cam_WriteReg(s_hCam, 0x202094, 1, &ROI[0]);
Cam_WriteReg(s_hCam, 0x202098, 1, &ROI[1]);
Cam_WriteReg(s_hCam, 0x20209C, 1, &ROI[2]);
Cam_WriteReg(s_hCam, 0x2020A0, 1, &ROI[3]);

// Set ROI (in block)
Cam_WriteReg(s_hCam, 0x202094, 4, &ROI[0]);
```

In IIDC2 registers access,

you can set OffsetX, Width, OffsetY, Height in any order. (in one by one access)

you can also set OffsetX, Width, OffsetY, Height with single access. (in block access)

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆**Minimum/Maximum Value**

Model	BU040M/BU040MCG	BU160M/BU160MCG
Width/OffsetX unit size	4	4
Height/OffsetY unit size	2	2
Minimum unit size	64 x 64	64 x 64
Maximum unit size(*)	720 x 540	1440 x 1080

* initial factory setting

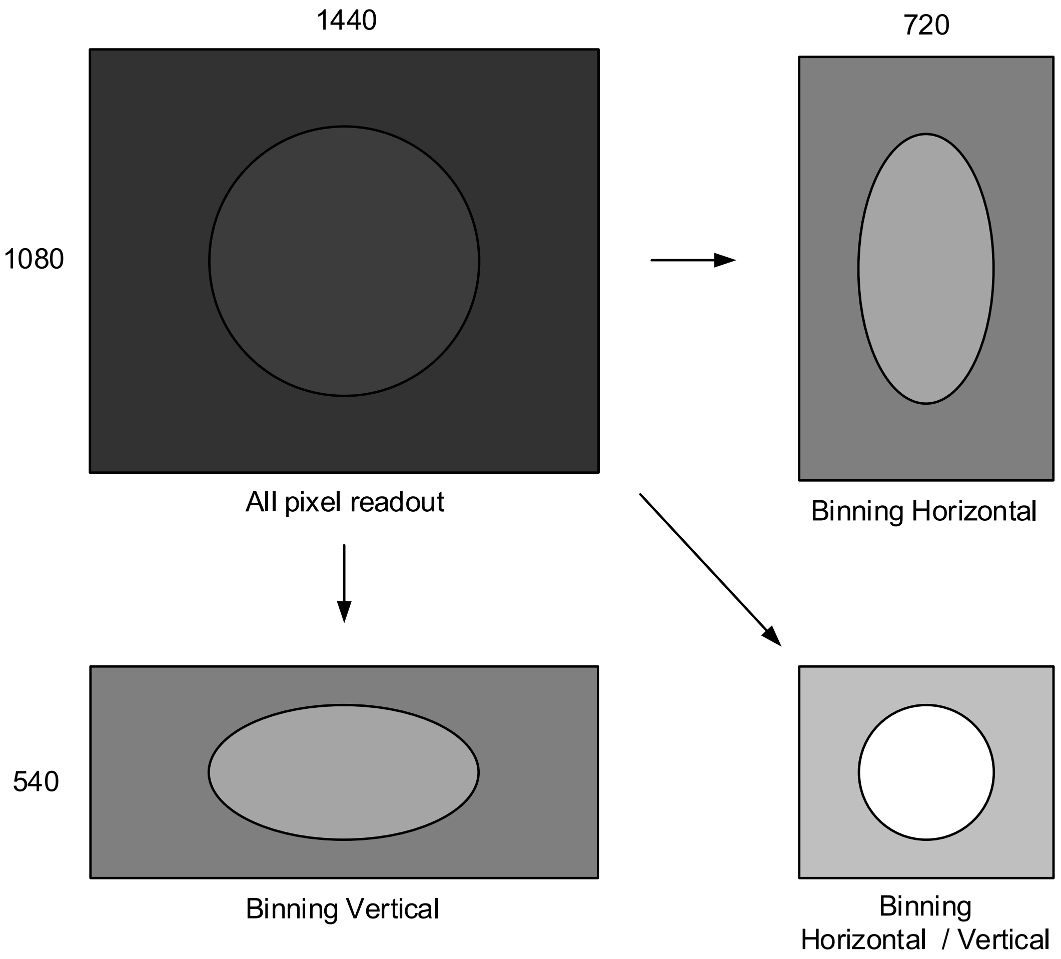
● Note

Changing “Width”, “Height”, “OffsetX”, “OffsetY” register value is invalid during image stream data output.

Binning

In the binning mode, a pixel is added with the neighboring pixel(s).

This increases the sensitivity of the image. It’s alike scalable, the frame rate can be faster and USB bandwidth occupation decrease.



Binning operation (e.g. BU160M)

- Framerate in each output format (fps)

- BU040M

Mono8		H	
		1	2
V	1	437	437
	2	437	437

Mono10/		H	
Mono12		1	2
V	1	320	320
	2	320	320

- BU160M

Mono8		H		
		1	2	4
V	1	226	226	226
	2	433	501	501
	4	433	501	501

Mono10/		水平		
Mono12		1	2	4
垂直	1	122	166	166
	2	244	317	317
	4	317	317	317

- BU040MCG

Bayer8/		H	
Mono8		1	2
V	1	436	436
	2	436	436

Bayer10/		H	
Bayer12		1	2
V	1	320	320
	2	320	320

YUV411		H	
		1	2
V	1	320	320
	2	320	320

YUV422		H	
		1	2
V	1	320	320
	2	320	320

RGB/BGR		H	
		1	2
V	1	320	320
	2	320	320

- BU160MCG

Bayer8/		H		
Mono8		1	2	4
V	1	227	227	227
	2	227	227	227
	4	227	227	227

Bayer10/		H		
Bayer12		1	2	4
V	1	122	166	166
	2	166	166	166
	4	166	166	166

YUV411		H		
		1	2	4
V	1	162	166	166
	2	166	166	166
	4	166	166	166

YUV422		H		
		1	2	4
V	1	122	166	166
	2	166	166	166
	4	166	166	166

RGB/BGR		H		
		1	2	4
V	1	81	162	166
	2	162	166	166
	4	166	166	166

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
BinningHorizontal	IInteger	4	R/W	Sets the Binning Horizontal.
BinningVertical	IInteger	4	R/W	Sets the Binning Vertical.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
BinningHorizontal	Implemented	0x202120	[31]	R	Returns the state whether the function is implemented.
	Min	0x202134	4	R	Returns the minimum value of BinningHorizontal Value register.
	Max	0x202138	4	R	Returns the maximum value of BinningHorizontal Value register.
	Value	0x20213C	4	R/W	Sets the Binning Horizontal.
BinningVertical	Implemented	0x202140	[31]	R	Returns the state whether the function is implemented.
	Min	0x202154	4	R	Returns the minimum value of BinningVertical Value register.
	Max	0x202158	4	R	Returns the maximum value of BinningVertical Value register.
	Value	0x20215C	4	R/W	Sets the Binning Vertical.

● Control with TeliCamSDK

Camera feature API

Control Binning using dedicated API.

API name	Description
GetCamBinningHorizontalMinMax	Get minimum and maximum value
GetCamBinningHorizontal	Get current BinningHorizontal value
SetCamBinningHorizontal	Set new BinningHorizontal value
GetCamBinningVerticalMinMax	Get minimum and maximum value
GetCamBinningVertical	Get current BinningVertical value
SetCamBinningVertical	Set new BinningVertical value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Binning using GenICam API.

◆Binning

```
// GenICam node handle
CAM_NODE_HANDLE    hBinning = NULL;

// Binning = 2x2
uint64_t       Binning = 2;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “BinningHorizontal”, &hBinning);
// Nd_GetNode(s_hCam, “BinningVertical”, &hBinning); // either will do

// Set Binning
Nd_SetIntValue(s_hCam, hBinning, Binning);
```

Please refer to [INode functions], [IInteger node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Binning by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Binning

Write to ‘Value’ field of ‘BinningHorizontal’ register to control BinningHorizontal.

Or write to ‘Value’ field of ‘BinningVertical’ register to control BinningVertical.

```
// Binning = 2x2
uint32_t       Binning = 2;

// Set Binning
Cam_WriteReg(s_hCam, 0x20213C, 1, &Binning);
// Cam_WriteReg(s_hCam, 0x20215C, 1, &Binning); // either will do
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

Binning is disabled when the camera is running in Decimation mode.

Scalable is enabled when the camera is running in Binning mode.

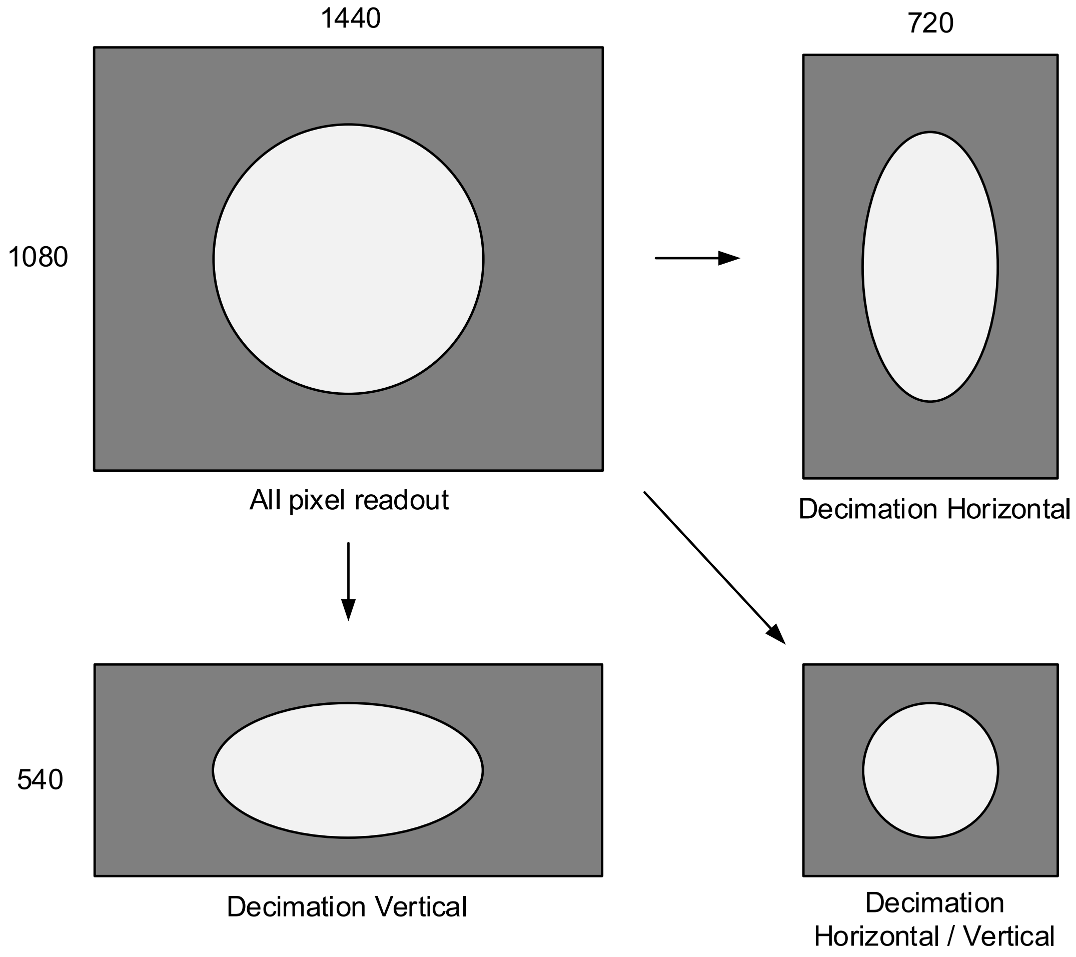
Changing “BinningHorizontal”, “BinningVertical” register value is invalid during image stream data output.

Binning Horizontal / Vertical = 3 is not available.

Decimation

Decimation feature reads out all effective areas at high speed by skipping pixels and lines.

Decimation feature can make frame rate faster, and decrease interface bandwidth occupation.



Decimation operation (e.g. BU160M)

- FrameRate in each output format (fps)

- BU040M

Mono8		H	
		1	2
V	1	437	437
	2	437	437

Mono10/		H	
Mono12		1	2
V	1	320	320
	2	320	320

- BU160M

Mono8		H		
		1	2	4
V	1	226	226	226
	2	433	501	501
	4	433	501	501

Mono10/		H		
Mono12		1	2	4
V	1	122	166	166
	2	244	317	317
	4	317	317	317

- BU040MCG

Bayer8/		H	
Mono8		1	2
V	1	436	436
	2	436	436

Bayer10/		H	
Bayer12		1	2
V	1	320	320
	2	320	320

YUV411		H	
		1	2
V	1	320	320
	2	320	320

YUV422		H	
		1	2
V	1	320	320
	2	320	320

RGB/BGR		H	
		1	2
V	1	320	320
	2	320	320

- BU160MCG

Bayer8/		H		
Mono8		1	2	4
V	1	227	227	227
	2	227	502	502
	4	227	502	502

Bayer10/		H		
Bayer12		1	2	4
V	1	122	166	166
	2	166	317	317
	4	166	317	317

YUV411		H		
		1	2	4
V	1	162	166	166
	2	166	317	317
	4	166	317	317

YUV422		H		
		1	2	4
V	1	122	166	166
	2	166	317	317
	4	166	317	317

RGB/BGR		H		
		1	2	4
V	1	81	162	166
	2	162	166	166
	4	166	166	166

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
DecimationHorizontal	IInteger	4	R/W	Sets the Decimation Horizontal.
DecimationVertical	IInteger	4	R/W	Sets the Decimation Vertical.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
DecimationHorizontal	Implemented	0x202160	[31]	R	Returns the state whether the function is implemented.
	Min	0x202174	4	R	Returns the minimum value of DecimationHorizontal Value register.
	Max	0x202178	4	R	Returns the maximum value of DecimationHorizontal Value register.
	Value	0x20217C	4	R/W	Sets the Decimation Horizontal.
DecimationVertical	Implemented	0x202180	[31]	R	Returns the state whether the function is implemented.
	Min	0x202184	4	R	Returns the minimum value of DecimationVertical Value register.
	Max	0x202188	4	R	Returns the maximum value of DecimationVertical Value register.
	Value	0x20218C	4	R/W	Sets the Decimation Vertical.

● Control with TeliCamSDK

Camera feature API

Control Decimation using dedicated API.

API name	Description
GetCamDecimationHorizontalMinMax	Get minimum and maximum value
GetCamDecimationHorizontal	Get current DecimationHorizontal value
SetCamDecimationHorizontal	Set new DecimationHorizontal value
GetCamDecimationVerticalMinMax	Get minimum and maximum value
GetCamDecimationVertical	Get current DecimationVertical value
SetCamDecimationVertical	Set new DecimationVertical value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Decimation using GenICam API.

◆Decimation

```
// GenICam node handle
CAM_NODE_HANDLE    hDecimation = NULL;

// Decimation = 2x2
uint64_t       Decimation = 2;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “DecimationHorizontal”, &hDecimation);
// Nd_GetNode(s_hCam, “DecimationVertical”, &hDecimation); // either will do

// Set Decimation
Nd_SetIntValue(s_hCam, hDecimation, Decimation);
```

Please refer to [INode functions], [IInteger node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Decimation by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Decimation

Write to ‘Value’ field of ‘DecimationHorizontal’ register to control DecimationHorizontal.

Or write to ‘Value’ field of ‘DecimationVertical’ register to control DecimationVertical.

```
// Decimation = 2x2
uint32_t       Decimation = 2;

// Set Decimation
Cam_WriteReg(s_hCam, 0x20213C, 1, &Decimation);
// Cam_WriteReg(s_hCam, 0x20215C, 1, &Decimation); // either will do
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

Decimation is disabled when the camera is running in Binning mode.

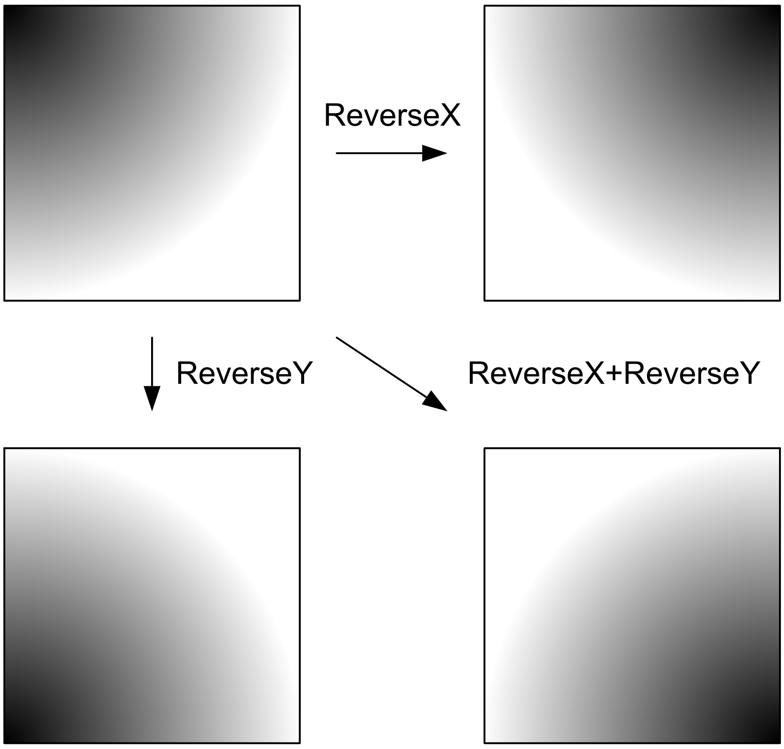
Scalable is enabled when the camera is running in Decimation mode.

Changing “DecimationHorizontal”, “DecimationVertical” register value is invalid during image stream data output.

Binning Horizontal / Vertical = 3 is not available.

Reverse

Reverse function flips image in horizontal and/or vertical direction..



● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ReverseX	IBoolean	4	R/W	Sets the Reverse Horizontal.
ReverseY	IBoolean	4	R/W	Sets the Reverse Vertical.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ReverseX	Implemented	0x2021A0	[31]	R	Returns the state whether the function is implemented.
	Value	0x2021B0	4	R/W	Sets the Reverse Horizontal. [0] : Off [1] : On
ReverseY	Implemented	0x2021C0	[31]	R	Returns the state whether the function is implemented.
	Value	0x2021D0	4	R/W	Sets the Reverse Vertical. [0] : Off [1] : On

● Control with TeliCamSDK

Camera feature API

Control Reverse using dedicated API.

API name	Description
GetCamReverseX	Get current ReverseX value
SetCamReverseX	Set new ReverseX value
GetCamReverseY	Get current ReverseY value
SetCamReverseY	Set new ReverseY value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Reverse using GenICam API.

◆Reverse

Sets the Reverse Horizontal by ‘ReverseX’.

Sets the Reverse Vertical by ‘ReverseY’.

```
// GenICam node handle
CAM_NODE_HANDLE    hReverseX = NULL;
CAM_NODE_HANDLE    hReverseY = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “ReverseX”, &hReverseX);
Nd_GetNode(s_hCam, “ReverseY”, &hReverseY);

// Set Reverse (flip horizontal and vertical direction)
Nd_SetBoolValue(s_hCam, hReverseX, true);
Nd_SetBoolValue(s_hCam, hReverseY, true);
```

Please refer to [INode functions], [IBoolean node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Reverse by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Reverse

Write to ‘Value’ field of ‘ReverseX’ register.

Write to ‘Value’ field of ‘ReverseY’ register.

```
// Set Reverse (flip horizontal and vertical direction)
uint32_t      dat = 1;
Cam_WriteReg(s_hCam, 0x2021B0, 1, &dat);
Cam_WriteReg(s_hCam, 0x2021D0, 1, &dat);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

- Changing ‘ReverseX’, ‘ReverseY’ register value is invalid during image stream data output.

PixelFormat

Select a pixel format of image stream data.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
PixelCoding	IEnumeration	4	R/W	Selects a pixel coding.
PixelSize	IEnumeration	4	R/W	Selects a bit size of image pixel.
PixelFormat	IEnumeration	4	R/W	Selects a pixel format. PixelFormat is conformed to AIA Pixel Format Naming Convention.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
PixelCoding	Implemented	0x2020C0	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x2020CC ~0x2020D8	16	R	Returns the presence of the pixel coding. [0] : Mono [32] : RGB [34] : RGBPacked [40] : BGR [42] : BGRPacked [66] : YUV411Packed [74] : YUV422Packed [96] : BayerGR [99] : BayerRG [102] : BayerGB [105] : BayerBG
	Value	0x2020DC	4	R/W	Selects a pixel cording.
PixelSize	Implemented	0x2020E0	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x2020EC ~0x2020F8	16	R	[8] : Bpp8 [10] : Bpp10 [12] : Bpp12 [16] : Bpp16 [24] : Bpp24
	Value	0x2020FC	4	R/W	Selects a bit size of image pixel.

Register	Field	Address	Length Byte / [bit]	Access	Description
PixelFormat	Implemented	0x202400	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20240C ~0x202418	16	R	Returns the presence of the pixel endian. [1] : Mono8 [3] : Mono10 [5] : Mono12 [8] : BayerGR8 [9] : BayerRG8 [10] : BayerGB8 [11] : BayerBG8 [12] : BayerGR10 [13] : BayerRG10 [14] : BayerGB10 [15] : BayerBG10 [16] : BayerGR12 [17] : BayerRG12 [18] : BayerGB12 [19] : BayerBG12 [20] : RGB [21] : BGR [30] : YUV411Packed [31] : YUV422Packed
	Value	0x20241C	4	R/W	Selects a pixel format

● Supported PixelFormat

◆B/W model

PixelSize / PixelCoding	Bpp8	Bpp10	Bpp12
Mono	Mono8 (*)	Mono10	Mono12
PixelFormat ID	0x01080001	0x01100003	0x01100005

* initial factory setting

◆Color model

PixelSize / PixelCoding	Bpp8	Bpp10	Bpp12	Bpp16	Bpp24
Mono	Mono8	-	-	-	-
PixelFormat ID	0x01080001				
BayerBG(*2)	BayerBG8	BayerBG10	BayerBG12	-	-
PixelFormat ID	0x0108000B	0x0110000F	0x01100013		
YUV411	-	-	YUV411Packed	-	-
PixelFormat ID			0x020C001E		
YUV422	-	-	-	YUV422Packed	-
PixelFormat ID				0x0210001F	
RGB	-	-	-	-	RGB8 (*1)
PixelFormat ID					0x02180014
BGR	-	-	-	-	BGR8
PixelFormat ID					0x02180015

*1 initial factory setting

*2 ReverseX and ReverseYsettings are FALSE

◆Bayer PixelFormat in Reverse function

ReverseX / ReverseY	FALSE		TRUE	
FALSE	BayerBG		BayerGB	
	Bpp8	0x0108000B	Bpp8	0x0108000A
	Bpp10	0x0110000F	Bpp10	0x0110000E
	Bpp12	0x01100013	Bpp12	0x01100012
TRUE	BayerGR		BayerRG	
	Bpp8	0x01080008	Bpp8	0x01080009
	Bpp10	0x0110000C	Bpp10	0x0110000D
	Bpp12	0x01100010	Bpp12	0x01100011

- **Control with TeliCamSDK**

Camera feature API

Control PixelFormat using dedicated API.

API name	Description
GetCamPixelFormat	Get current PixelFormat value
SetCamPixelFormat	Set new PixelFormat value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control PixelFormat using GenICam API.

◆PixelCoding/PixelSize

To determine PixelFormat combination of ‘PixelCoding‘ and ‘PixelSize‘ register.

1. Select a pixel coding. (‘PixelCoding’).

   Integer value and string value of Enumeration are as follows.

Integer	String
0	Mono
32	RGB
34	RGBPacked
40	BGR
42	BGRPacked
66	YUV411Packed
74	YUV422Packed
96	BayerGR
99	BayerRG
102	BayerGB
105	BayerBG

2. Select a bit size of image pixel (‘PixelSize’).

   Integer value and string value of Enumeration are as follows.

Integer	String
8	Bpp8
10	Bpp10
12	Bpp12
16	Bpp16
24	Bpp24

```
// GenICam node handle
CAM_NODE_HANDLE    hCoding = NULL;
CAM_NODE_HANDLE    hSize = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "PixelCoding", &hCoding);
Nd_GetNode(s_hCam, "PixelSize", &hSize);

// 1.Select a pixel coding.
Nd_SetEnumStrValue(s_hCam, hCoding, "Mono");
// 2.Select a pixel size.
Nd_SetEnumStrValue(s_hCam, hSize, "Bpp10");
```

◆PixelFormat

Select a pixel format. (‘PixelFormat’).

Integer value and string value of Enumeration are as follows.

Integer	String
1	Mono8
3	Mono10
5	Mono12
8	BayerGR8
9	BayerRG8
10	BayerGB8
11	BayerBG8
12	BayerGR10
13	BayerRG10
14	BayerGB10

Integer	String
15	BayerBG10
16	BayerGR12
17	BayerRG12
18	BayerGB12
19	BayerBG12
20	RGB8
21	BGR8
30	YUV411Packed
31	YUV422Packed

```
// GenICam node handle
CAM_NODE_HANDLE    hFormat = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "PixelFormat", &hFormat);

// 1.Select a pixel format.
Nd_SetEnumStrValue(s_hCam, hFormat, "Mono10");
```

Please refer to [INode functions], [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control PixelFormat by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆PixelCoding/PixelSize

To determine PixelFormat combination of ‘PixelCoding‘ and ‘PixelSize‘ register.

1. Write to ‘Value’ field of ‘PixelCoding’ register.
2. Write to ‘Value’ field of ‘PixelSize’ register.

```
uint32_t coding = 0; // Mono
uint32_t size = 10; // Bpp10

// 1.Select a pixel coding.
Cam_WriteReg(s_hCam, 0x2020DC, 1, &coding);
// 2.Select a pixel size.
Cam_WriteReg(s_hCam, 0x2020FC, 1, &size);
```

◆PixelFormat

Write to ‘Value’ field of ‘PixelFormat’ register.

```
uint32_t format = 3; // Mono10

// 1.Select a pixel format.
Cam_WriteReg(s_hCam, 0x20241C, 1, &format);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

Changing ‘PixelFormat’ register value is invalid during image stream data output.

BayerProcessingMode

In the color model, you can select the BayerProcessingMode of Bayer output.

Color process function be applied is dependent on the setting.

Selection of BayerProcessingMode is as following table.

BayerProcessingMode	Function
Full	Full processes are available
Partial	Partial processes are available
Raw	Only Gain feature is available

Function	Full (*)	Partial	Raw
Gain	✓	✓	✓
BlackLevel	✓	✓	-
Gamma	✓	✓	-
BalanceRatio	✓	✓	-
ColorCorrectionMatrix	✓	-	-
LUTControl	✓	✓	-
DPCControl	✓	✓	-

* initial factory setting

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
BayerProcessingMode	IEnumeration	4	R/W	Selects a Bayer Processing Mode.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
BayerProcessingMode	Implemented	0x21F420	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F42C ~0x21F438	16	R	Returns the presence of the pixel endian. [0] : Raw [8] : Partial [16] : Full
	Value	0x21F43C	4	R/W	Selects a Bayer Processing Mode.

● Control with TeliCamSDK

GenICam function API

Control BayerProcessingMode using dedicated API.

◆BayerProcessingMode

Select BayerProcessingMode by ‘BayerProcessingMode’.

Integer value and string value of Enumeration are as follows.

Integer	String
0	Raw
8	Partial
16	Full

```
// GenICam node handle
CAM_NODE_HANDLE    hMode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "BayerProcessingMode", &hMode);

// BayerProcessingMode = Raw
Nd_SetEnumStrValue(s_hCam, hMode, "Raw");
```

Please refer to [INode functions], [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

This API access IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆BayerProcessingMode

Write to ‘Value’ field of ‘BayerProcessingMode’ register.

```
// BayerProcessingMode = Raw
uint32_t       uiMode;
uiMode = 0;
Cam_WriteReg(s_hCam, 0x21F43C, 1, &uiMode);
```

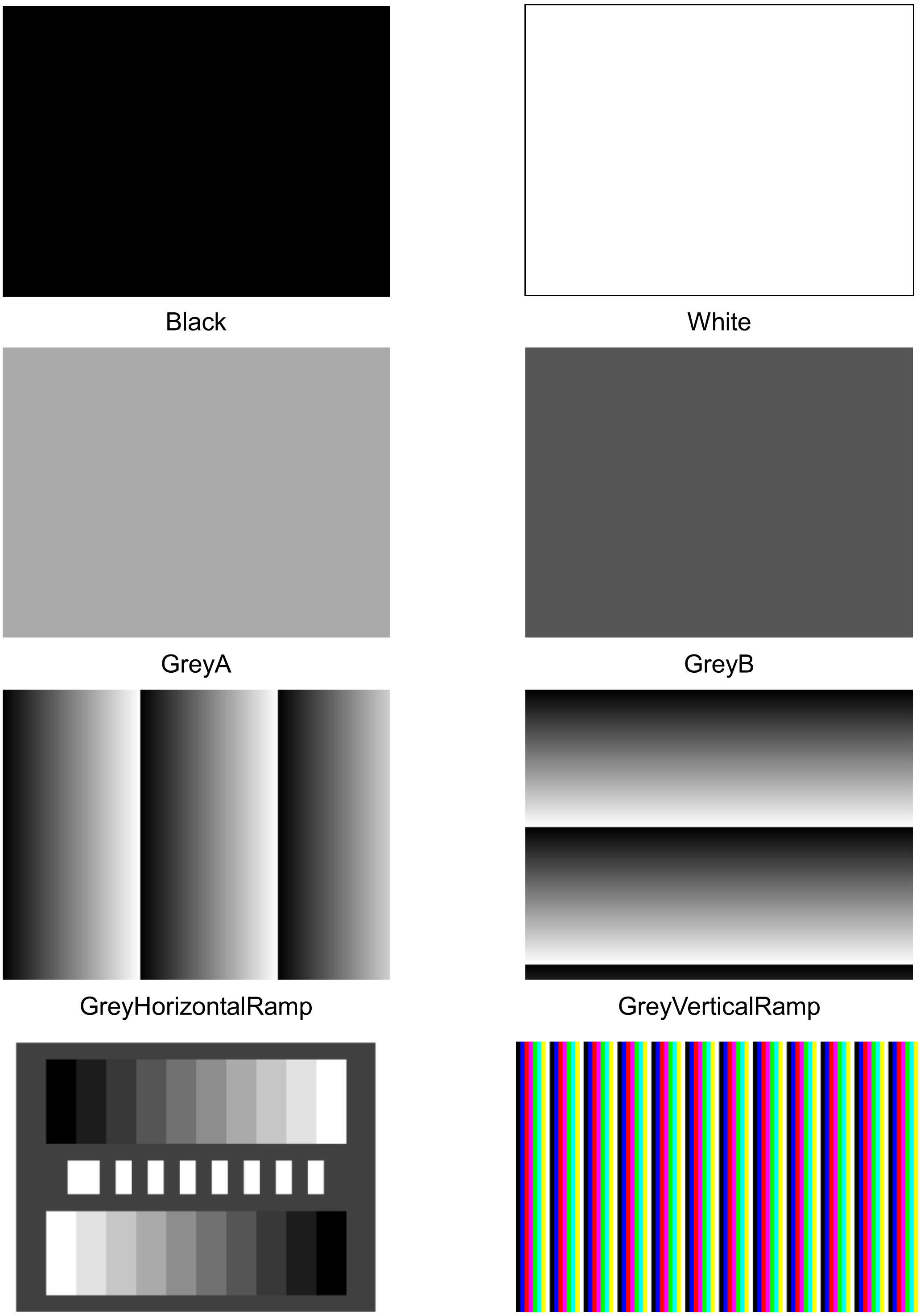
Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

Changing ‘BayerProcessingMode’ register value is invalid during image stream data output.

TestPattern

BU series supports test pattern data output. Camera provides following Test patterns;



Black

White

GreyA

GreyB

GreyHorizontalRamp

GreyVerticalRamp

GreyScale

(B/W model only)

ColorBar

(Color model only)

Test pattern (e.g. BU040M/BU040MCG)

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
TestPattern	IEnumeration	4	R/W	Selects a Test pattern.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
TestPattern	Implemented	0x21F120	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F12C ~0x21F138	16	R	Returns the presence of test pattern. [0] : Off [1] : Black [2] : White [3] : GrayA [4] : GrayB [5] : GreyHorizontalRamp [6] : GrayScale [7] : ColorBar [8] : GreyVerticalRamp
	Value	0x21F13C	4	R/W	Selects a Test pattern.

● Control with TeliCamSDK

Camera feature API

Control TestPattern using dedicated API.

API name	Description
GetCamTestPattern	Get current TestPattern value
SetCamTestPattern	Set new TestPattern value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control TestPattern using GenICam API.

◆TestPattern

Select a test pattern.

Integer value and string value of Enumeration are as follows.

Integer	String	Function
0 (*)	Off (*)	Test pattern disable(Normal data output)
1	Black	All pixel = 0 LSB
2	White	All pixel = 255 @Mono8
3	GreyA	All pixel = 170 @Mono8
4	GreyB	All pixel = 85 @Mono8
5	GreyHorizontalRamp	Horizontal Ramp
6	GreyScale	Grey scale (B/W model only)
7	ColorBar	Color bars (Color model only)
8	GreyVerticalRamp	Vertical Ramp

* initial factory setting

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "TestPattern", & hNode);

// 1.Select a test pattern.
Nd_SetEnumStrValue(s_hCam, hNode, "GreyHorizontalRamp");
```

Please refer to [INode functions], [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control TestPattern by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆TestPattern

Write to ‘Value’ field of ‘TestPattern’ register.

```
uint32_t dat = 5; // Horizontal Ramp

// 1.Select a test pattern.
Cam_WriteReg(s_hCam, 0x21F13C, 1, &dat);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

AcquisitionControl

AcquisitionControl features are related to image acquisition.

Camera starts image stream output by receiving AcquisitionStart command.

Camera stops image stream output by receiving AcquisitionStop command.

There are some registers that require camera to stop image stream output to change values.

Acquisition frame rate is variable. Maximum acquisition frame rate depends on camera operation mode. (Scalable, Pixel format, etc.)

Color models have HighFramerateMode. You can improve the frame rate by using HighFramerateMode.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
AcquisitionMode	IEnumeration	4	R/W	Selects an acquisition mode.
AcquisitionStart	ICommand	4	W	Executes the image stream output start.
AcquisitionStop	ICommand	4	W	Executes the image stream output stop.
AcquisitionAbort	ICommand	4	W	Executes the image stream output abort.
AcquisitionFrameCount	IInteger	4	R/W	Sets the number of frames to transfer in MultiFrame/ImageBuffer mode.
AcquisitionFrameRateEnable	IEnumeration	4	R/W	Selects an AcquisitionFrameRate setting priority.
AcquisitionFrameRate	IFloat	4	R/W	Sets frame rate of image stream.
AcquisitionFrameIntervalControl	IEnumeration	4	R/W	Selects an AcquisitionFrameInterval setting priority.
AcquisitionFrameInterval	IFloat	4	R/W	Sets frame interval of image stream.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
AcquisitionMode	Implemented	0x203020	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20302C ~0x203038	16	R	[0]: Abort [1]: Stop [8]: Continuous [9]: MultiFrame [10]: Image Buffer Read
	Value	0x20303C	4	R/W	Selects an Acquisition mode.
AcquisitionFrameCount	Implemented	0x203040	[31]	R	Returns the state whether the function is implemented.
	Min	0x203054	4	R	Returns the minimum number of frames in MultiFrame/ImageBuffer mode.
	Max	0x203058	4	R	Returns the maximum number of frames in MultiFrame/ImageBuffer mode.
	Value	0x20305C	4	R/W	Sets the number of frames in MultiFrame/ImageBuffer mode.
AcquisitionFrameRate	Implemented	0x2030A0	[31]	R	Returns the state whether the function is implemented.
	Control	0x2030A8	4	R/W	[0] : NoSpecify [1] : Manual
	Mult	0x2030AC	4	R	Absolute Value [fps] = Raw * (Mult / Div)
	Div	0x2030B0	4	R	
	Min	0x2030B4	4	R	Returns the minimum value of frame rate.
	Max	0x2030B8	4	R	Returns the maximum value of frame rate.
	Value	0x2030BC	4	R/W	Sets frame rate of image stream in raw value.
AcquisitionFrameInterval	Implemented	0x2030C0	[31]	R	Returns the state whether the function is implemented.
	Control	0x2030C8	4	R/W	[0] : NoSpecify [1] : Manual
	Mult	0x2030CC	4	R	AbsoluteValue[sec] = Raw * (Mult / Div)
	Div	0x2030D0	4	R	
	Min	0x2030D4	4	R	Returns the minimum value of frame interval.
	Max	0x2030D8	4	R	Returns the maximum value of frame interval.
	Value	0x2030DC	4	R/W	Sets frame interval of image stream in raw value.
HighFramerateMode	Implemented	0x21F660	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F66C	4	R	[0] : Off [1] : On
	Value	0x21F67C	4	R/W	Only Color models. Sets the activation of HighFramerateMode.

● Control with TeliCamSDK

There are three way to access this feature.

Camera feature API

This API is dedicated to AcquisitionControl.

API name	Description
GetCamAcquistionFrameCountMinMax	Get minimum and maximum value
GetCamAcquistionFrameCount	Get current value
SetCamAcquistionFrameCount	Set new value
GetCamAcquistionFrameRateControl	Get current value
SetCamAcquistionFrameRateControl	Set new value
GetCamAcquistionFrameRateMinMax	Get minimum and maximum value
GetCamAcquistionFrameRate	Get current value
SetCamAcquistionFrameRate	Set new value

1. Set the number of frames to transfer in MultiFrame/ImageBuffer mode. (optional)

   Set 'AcquisitionFrameCount' with 'SetCamAcquistionFrameCount' function.

   *If you want to capture image stream continuously, it's not necessary to set an AcquisitionFrameCount.

2. Select an 'AcquisitionFrameRate' setting priority. (optional)

   Set 'AcquisitionFrameRateControl' with 'SetCamAcquistionFrameRateControl' function.

AcquisitionFrameRateControl	Function
NoSpecify (*)	The frame rate is determined by giving priority to ExposureTime setting value.
Manual	The frame rate is determined by giving priority to AcquisitionFrameRate setting value.

* initial factory setting

3. Set frame rate of image stream. (optional)

   Set 'AcquisitionFrameRate' with 'SetCamAcquistionFrameRate' function.

   Or, set 'AcquisitionFrameInterval' with 'SetCamAcquistionFrameInterval' function.

   'AquisitionFrameInterval' is a reciprocal of 'AcquisitionFrameRate'.

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

4. Capture image stream.

   TeliCamAPI provides streaming functions including command handling and buffer handling to capture image stream simply.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.

GenICam function API

This API controls AcquisitionControl using GenICam.

1. Set the number of frames to transfer in MultiFrame/ImageBuffer mode. (optional)

   Get ‘IInteger’ interface handle of ‘AcquisitionFrameCount’ by ‘Nd_GetNode’.

   Set AcquisitionFrameCount with IInteger node functions.

   (e.g. ‘Nd_GetIntMin’, ‘Nd_GetIntMax’, ‘Nd_GetIntValue’, ‘Nd_SetIntValue’)

   *If you want to capture image stream continuously, it’s not necessary to set an AcquisitionFrameCount.

2. Select an ‘AcquisitionFrameRate’ setting priority. (optional)

   Get ‘IEnumeration’ interface handle of ‘AcquisitionFrameRateControl’ by ‘Nd_GetNode’.

   Set AcquisitionFrameRateControl with IEnumeration node functions.

   (e.g. ‘Nd_GetEnumIntValue’, ‘Nd_SetEnumIntValue’, ‘Nd_GetEnumStrValue’, ‘Nd_SetEnumStrValue’)

   Integer value and string value of Enumeration are as follows.

Integer	String
0(*)	NoSpecify
1	Manual

* initial factory setting

3. Set frame rate of image stream. (optional)

   Get ‘IFloat’ interface handle of ‘AcquisitionFrameRate’ or ‘AcquisitionFrameInterval’ by ‘Nd_GetNode’.

   Set AcquisitionFrameRate or AcquisitionFrameInterval with IFloat node functions.

   ‘AquisitionFrameInterval’ is a reciprocal of ‘AcquisitionFrameRate’.

   (e.g. ‘Nd_GetFloatMin’, ‘Nd_GetFloatMax’, ‘Nd_GetFloatValue’, ‘Nd_SetFloatValue’)

4. Capture image stream.

   TeliCamAPI provides streaming functions including command handling and buffer handling to capture image stream simply.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.

Register access API

This API access IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

1. Set the number of frames to transfer in MultiFrame/ImageBuffer mode. (optional)

   Write to ‘Value’ field of ‘AcquisitionFrameCount’ register to control AcquisitionFrameCount.

2. Select an ‘AcquisitionFrameRate’ setting priority. (optional)

   Write to ‘Control’ field of ‘AcquisitionFrameRate’ register to control AcquisitionFrameRate setting priority.

3. Set a ‘HighFrameRateMode’ setting. (optional)

   Write to ‘Value’ field of ‘HighFrameRateMode’ register to control HighFrameRateMode setting.

4. Set frame rate of image stream. (optional)

   Write to ‘Value’ field of ‘AcquisitionFrameRate’ register to control AcquisitionFrameRate.

5. Capture image stream.

   TeliCamAPI provides streaming functions including command handling and buffer handling to capture image stream simply.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

AcquisitionFrameRate		BU040M	BU160M	BU040MCG	BU160MCG
Minimum	Raw Value	1			
	Absolute (Float) Value	0.061 [fps]			
Maximum	Raw Value	Depend on register setting of "Height".			
	Absolute (Float) Value				
Initial Value	Raw Value	7167	3713	5253	1332
	Absolute (Float) Value	437.4 [fps]	226.6 [fps]	320.6 [fps]	81.2[fps]
Fomula		Absolute Value [fps] = Raw Value * 125 / 2048			

Notes on HighFramerateMode:

If you use HighFramerateMode, improves frame rate and sensitivity by about 4 times, but image quality may deteriorate. Also, the actual exposure time may vary depending on the individual differences and the operating environment (such as the operating temperature). When using the HighFramerateMode, I ask you to have final image quality checked with your emvironment.

● Note

Changing 'AcquisitionFrameRateControl', 'AcquisitionFrameRate', 'AcquisitionFrameIntervalControl', 'AcquisitionFrameInterval', 'HighFramerateMode' register value is invalid during image stream data output. When exposure time setting is longer than frame rate setting, camera operation gives priority to exposure time setting.

Notes on Frame Drops of Image:

Depends on your PC or interface card configurations, images may not be captured normally (e.g. frame drops may occur). In this case, change to frame rate setting lower.

ImageBuffer

In ImageBuffer mode, Camera stores images temporarily in image buffer, and read them out in arbitrary timing.

This function is typically used in Random Trigger Shutter mode.

Please refer to TriggerControl section as well.

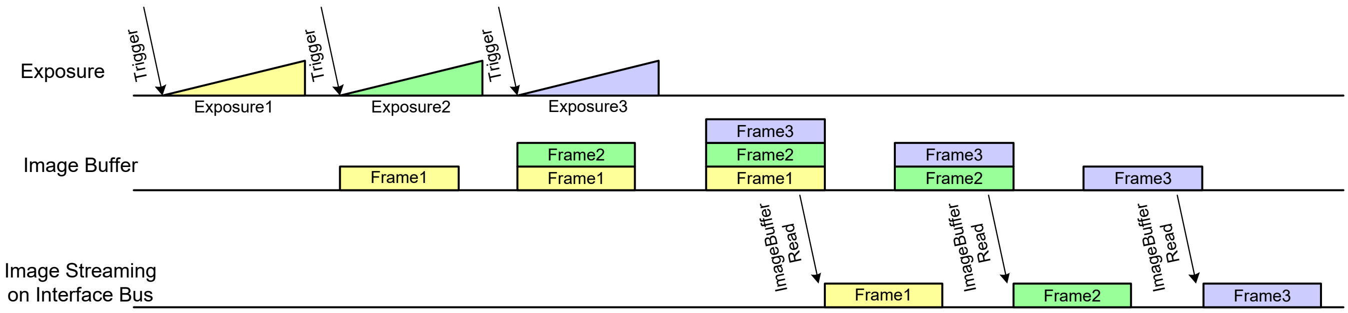


Image Buffer

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ImageBufferMode	IEnumeration	4	R/W	Selects an Image Buffer mode.
ImageBufferFrameCount	IInteger	4	R/W	Sets the number of frames to transfer in MultiFrame/ImageBuffer mode.
ImageBufferRead	ICommand	4	W	Read Image from Image Buffer

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ImageBufferMode	Implemented	0x203060	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20306C ~0x203078	16	R	[0] : Off [1] : On
	Value	0x20307C	4	R/W	Sets an Image Buffer mode.
ImageBufferFrameCount	Implemented	0x203080	[31]	R	Returns the state whether the function is implemented.
	Min	0x203094	4	R	Returns the minimum number of frames in Image Buffer mode.
	Max	0x203098	4	R	Returns the maximum number of frames in Image Buffer mode.
	Value	0x20309C	4	R	Returns the number of frames in Image Buffer mode.

● Control with TeliCamSDK

Camera feature API

Control ImageBuffer using dedicated API.

API name	Description
GetCamImageBufferMode	Get current ImageBuffer mode
SetCamImageBufferMode	Set new ImageBuffer mode
GetCamImageBufferFrameCount	Get current ImageBufferFrameCount value
ExecuteCamImageBufferRead	Read Image from Image Buffer

◆ImageBuffer

1. Select an Image Buffer mode.

   Set 'ImageBufferMode' with 'SetCamImageBufferMode' function.

2. Capture image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to 'AcquisitionControl' section as well.

3. Read the number of frames in Image Buffer.

   Read the number of frames in Image Buffer with 'GetCamImageBufferFrameCount' function.

4. Read Image from Image Buffer.

   Start to read Image from Image Buffer with 'ExecuteCamImageBufferRead' function.

5. Receive Image from Image Buffer.

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control ImageBuffer using GenICam API.

◆ImageBuffer

1. Select an Image Buffer mode by 'ImageBufferMode'.

   Integer value and string value of Enumeration are as follows.

Integer	String
0(*)	Off
1	On

* initial factory setting

2. Open and Start image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to AcquisitionControl section as well.
3. Read the number of frames in Image Buffer by ‘ImageBufferFrameCount’.
4. Read Image from Image Buffer by ‘ImageBufferRead’.
5. Receive Image from Image Buffer.
6. Stop and Close image stream.

```
// GenICam node handle
CAM_NODE_HANDLE    hMode = NULL;
CAM_NODE_HANDLE    hCount = NULL;
CAM_NODE_HANDLE    hRead = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "ImageBufferMode", &hMode);
Nd_GetNode(s_hCam, "ImageBufferFrameCount", &hCount);
Nd_GetNode(s_hCam, "ImageBufferRead", &hRead);

// 1.Select an Image Buffer mode
Nd_SetEnumStrValue(s_hCam, hMode, "On");

// 2.Open and Start image stream.
//    2.1.Set Trigger mode
        SetCamTriggerMode(s_hCam, true);
        SetCamTriggerSource(s_hCam, CAM_TRIGGER_SOFTWARE);
//    2.2.Open Stream
        s_hStrmEvt = CreateEvent(NULL, FALSE, FALSE, NULL);
        Strm_OpenSimple(s_hCam, &s_hStrm, &s_uiImgBufSize, s_hStrmEvt);
        s_pucImgBuf = (uint8_t *)VirtualAlloc(NULL, s_uiImgBufSize, MEM_RESERVE
| MEM_COMMIT, PAGE_EXECUTE_READWRITE);
//    2.3.Stream Start
        Strm_Start(s_hStrm);
//    2.4.Execute Software Trigger
        ExecuteCamSoftwareTrigger(s_hCam);

// 3.Read the number of frames in Image Buffer by 'ImageBufferFrameCount'.
int64_t count = 0;
while(count==0)
{
        Nd_GetIntValue(s_hCam, hCount, &count);
}

// 4.Read Image from Image Buffer by 'ImageBufferRead'.
Nd_CmdExecute(s_hCam, hRead);

// 5.Receive Image from Image Buffer
        uint32_t uiSize    = s_uiImgBufSize;
        WaitForSingleObject(s_hStrmEvt, 1000);
        Strm_ReadCurrentImage(s_hStrm, s_pucImgBuf, &uiSize, NULL);

// 6.Stop and Close image stream.
//    6.1.Stream Stop
        Strm_Stop(s_hStrm);
//    6.2.Close Stream
        Strm_Close(s_hStrm);
        CloseHandle(s_hStrmEvt);
        VirtualFree(s_pucImgBuf, 0, MEM_RELEASE);
```

Register access API

Control Image Buffer by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆ImageBuffer

1. Select an Image Buffer mode.

   Write to ‘Value’ field of ‘ImageBufferMode’ register to control ImageBufferMode.

2. Open and Start image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to AcquisitionControl section as well.

3. Read the number of frames in Image Buffer.

   Read ‘Value’ field of ‘ImageBufferFrameCount’ register.

4. Read Image from Image Buffer.

   Write [10] to ‘Value’ field of ‘AcquisitionCommand’ register to execute ImageBufferRead.

   Please refer to AcquisitionCommand in AcquisitionControl section as well.

5. Receive Image from Image Buffer.

6. Stop and Close image stream.

```
// 1.Select an Image Buffer mode
uint32_t      dat = 1;
Cam_WriteReg(s_hCam, 0x20307C, 1, &dat);

// 2.Open and Start image stream.
//    2.1.Set Trigger mode
        SetCamTriggerMode(s_hCam, true);
        SetCamTriggerSource(s_hCam, CAM_TRIGGER_SOFTWARE);
//    2.2.Open Stream
        s_hStrmEvt = CreateEvent(NULL, FALSE, FALSE, NULL);
        Strm_OpenSimple(s_hCam, &s_hStrm, &s_uiImgBufSize, s_hStrmEvt);
        s_pucImgBuf = (uint8_t *)VirtualAlloc(NULL, s_uiImgBufSize, MEM_RESERVE
| MEM_COMMIT, PAGE_EXECUTE_READWRITE);
//    2.3.Stream Start
        Strm_Start(s_hStrm);
//    2.4.Execute Software Trigger
        ExecuteCamSoftwareTrigger(s_hCam);

// 3.Read the number of frames in Image Buffer.
uint32_t count = 0;
while(count==0)
{
        Cam_ReadReg(s_hCam, 0x20309C, 1, &count);
}

// 4.Read Image from Image Buffer. AcquisitionCommand = 10 : Image Buffer Read
dat = 10;
Cam_WriteReg(s_hCam, 0x20303C, 1, &dat);

// 5.Receive Image from Image Buffer
        uint32_t uiSize    = s_uiImgBufSize;
        WaitForSingleObject(s_hStrmEvt, 1000);
        Strm_ReadCurrentImage(s_hStrm, s_pucImgBuf, &uiSize, NULL);

// 6.Stop and Close image stream.
//    6.1.Stream Stop
        Strm_Stop(s_hStrm);
//    6.2.Close Stream
        Strm_Close(s_hStrm);
        CloseHandle(s_hStrmEvt);
        VirtualFree(s_pucImgBuf, 0, MEM_RELEASE);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

The maximum number of frames storable depends on the image size. (maximum 256MByte.)

The number of frames to be transferred by ‘ImageBufferRead’ command is determined by ‘AcquisitionFrameCount’.

Changing ‘ImageBufferMode’ register value is invalid during image stream data output.

TriggerControl

TriggerControl features are related to image acquisition using trigger.

This camera series provides two kinds of exposure synchronization.

1. Normal Shutter mode : Free run operation (internal synchronization)
2. Random Trigger Shutter mode : Synchronized with external trigger input

In Random Trigger Shutter mode, two kinds of trigger input are available.

1. Trigger signal via the I/O connector (HardwareTrigger)
2. Trigger command via software command (SoftwareTrigger)

The following table shows the combination of operation mode of this camera series.

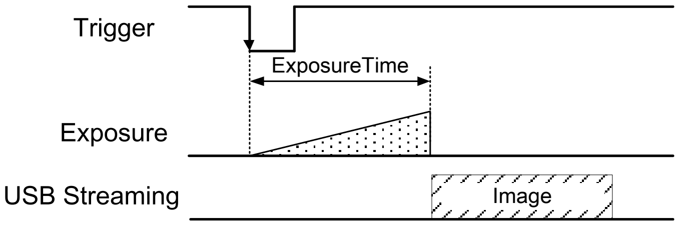
Operation Mode

Trigger Mode	Synchronization	Exposure Control
Normal Shutter mode	Free run	“ExposureTime” register control
Random Trigger Shutter mode	HardwareTrigger	-Edge mode:TriggerSequence0 -Bulk mode:TriggerSequence6 “ExposureTime” register control
		-Level mode:TriggerSequence1 Trigger pulse width control
	SoftwareTrigger	-Edge mode:TriggerSequence0 -Bulk mode:TriggerSequence6 “ExposureTime” register control

* The camera operation not mentioned above is not supported.

- Edge mode : TriggerSequence0

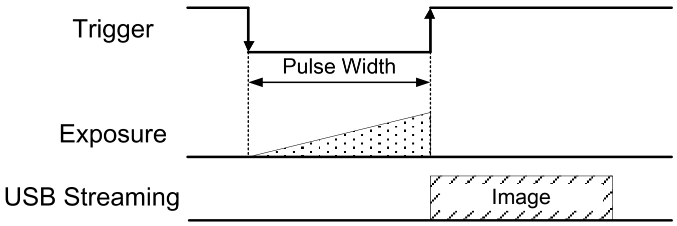
The exposure time is determined by Exposure Time setting.



- Level mode : TriggerSequence1

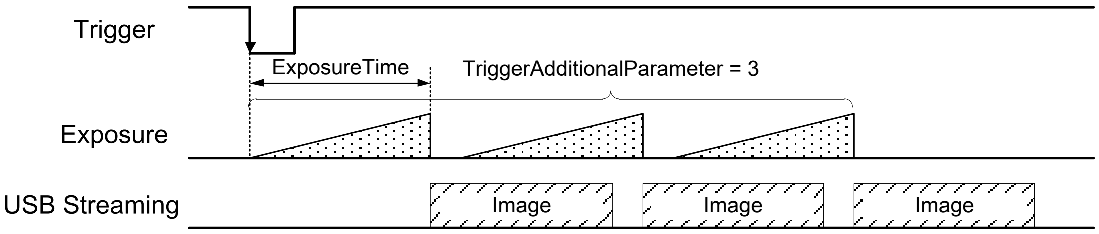
The exposure time is determined by the pulse width of the trigger signal.

It is not supported when the ShortExposureMode is enable (ON).

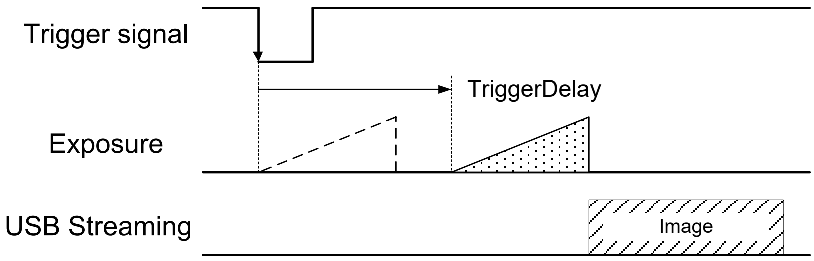


- Bulk mode (FrameBurstTrigger) : TriggerSequence6

Camera exposes and transfers multiple frames by a single trigger.



Operation point of HardwareTrigger is at the edge of trigger signal, and active edge polarity is able to change by register setting. And you can add delay time from trigger edge to exposure start by register setting.



Trigger Delay

For details of Random Trigger Shutter operation, please refer to ‘Timing’ section in ‘Specification’.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
TriggerMode	IEnumeration	4	R/W	Selects a trigger mode.
TriggerSoftware	ICommand	4	W	Executes software trigger.
TriggerSource	IEnumeration	4	R/W	Selects a trigger source of random trigger shutter.
TriggerActivation	IEnumeration	4	R/W	Selects a trigger activation of hardware trigger.
TriggerDelay	IFloat	4	R/W	Sets a trigger delay.
TriggerSequence	IEnumeration	4	R/W	Selects a trigger sequence of random trigger shutter.
TriggerAdditionalParameter	IInteger	4	R/W	Sets the number of frames to exposure in Bulk mode.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
TriggerMode	Implemented	0x207020	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20702C ~0x207038	16	R	[0] : OFF (Normal shutter) [1] : ON (Random trigger shutter)
	Value	0x20703C	4	R/W	Selects a trigger mode.
TriggerSequence	Implemented	0x207040	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20704C ~0x207058	16	R	[0] : TriggerSequence0 (Edge mode) [1] : TriggerSequence1 (Level mode) [6] : TriggerSequence6 (Bulk mode)
	Value	0x20705C	4	R/W	Selects an exposure mode.
TriggerSource	Implemented	0x207060	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20706C ~0x207078	16	R	[0] : Line0 (Hardware Trigger) [2] : Line2 (Hardware Trigger) [64] : Software (SoftwareTrigger)
	Value	0x20707C	4	R/W	Selects a trigger source.
TriggerAdditionalParameter	Implemented	0x207080	[31]	R	Returns the state whether the function is implemented.
	Min	0x207094	4	R	Returns the minimum value of the number of images in Bulk mode.
	Max	0x207098	4	R	Returns the maximum value of the number of images in Bulk mode.
	Value	0x20709C	4	R/W	Sets the number of frames to exposure in Bulk mode.

Register	Field	Address	Length Byte / [bit]	Access	Description
TriggerDelay	Implemented	0x2070A0	[31]	R	Returns the state whether the function is implemented.
	Mult	0x2070AC	4	R	AbsoluteValue[sec] = Raw * (Mult / Div)
	Div	0x2070B0	4	R	
	Min	0x2070B4	4	R	Returns the minimum value of trigger delay.
	Max	0x2070B8	4	R	Returns the maximum value of trigger delay.
	Value	0x2070BC	4	R/W	Sets the trigger delay.(Raw value)
SoftwareTrigger	Implemented	0x207040	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20702C ~0x207038	16	R	[0] : Inactive [8] : Impulse
	Value	0x20705C	4	R/W	Executes software trigger.

● Control with TeliCamSDK

Camera feature API

Control Trigger features using dedicated API.

API name	Description
GetCamTriggerMode	Get current TriggerMode
SetCamTriggerMode	Set new TriggerMode
GetCamTriggerSequence	Get current TriggerSequence
SetCamTriggerSequence	Set new TriggerSequence
GetCamTriggerSource	Get current TriggerSource
SetCamTriggerSource	Set new TriggerSource
GetCamTriggerAdditionalParameterMinMax	Get minimum and maximum value
GetCamTriggerAdditionalParameter	Get current TriggerAdditionalParameter value
SetCamTriggerAdditionalParameter	Set new TriggerAdditionalParameter value
GetCamTriggerDelayMinMax	Get minimum and maximum value
GetCamTriggerDelay	Get current TriggerDelay value
SetCamTriggerDelay	Set new TriggerDelay value
ExecuteCamSoftwareTrigger	Execute Software Trigger

1. Select a trigger mode.

   Set 'TriggerMode' with 'SetCamTriggerMode' function.
2. Select a trigger sequence of random trigger shutter.

   Set 'TriggerSequence' with 'SetCamTriggerSequence' function.
3. Select a trigger source of random trigger shutter.

   Set 'TriggerSource' with 'SetCamTriggerSource' function.
4. Select a trigger activation of hardware trigger.
5. Set the number of frames to exposure in Bulk mode.

   Set 'TriggerAdditionalParameter' with 'SetCamTriggerAdditionalParameter' function.
6. Set a trigger delay.

   Set 'TriggerDelay' with 'SetCamTriggerDelay' function.
7. Capture image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.
8. Execute software trigger.

   Execute Software Trigger in SoftwareTrigger mode with 'ExecuteCamSoftwareTrigger' function.

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Trigger features using GenICam API.

1. Select a trigger mode by ‘TriggerMode’.

   Integer value and string value of Enumeration are as follows.

Integer	String
0(*)	Off
1	On

* initial factory setting

2. Select a trigger sequence of random trigger shutter by ‘TriggerSequence’.

   Integer value and string value of Enumeration are as follows.

Integer	String
0(*)	TriggerSequence0
1	TriggerSequence1
6	TriggerSequence6

* initial factory setting

3. Select a trigger source of random trigger shutter by ‘TriggerSource’.

   Integer value and string value of Enumeration are as follows.

Integer	String
0(*)	Line0
2	Line2
64	Software

* initial factory setting

4. Select a trigger activation of hardware trigger by ‘Trigger Activation.
5. Set the number of frames to exposure in Bulk mode by ‘TriggerAdditionalParameter’.
6. Set a trigger delay by ‘TriggerDelay’.
7. Capture image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.
8. Execute software trigger in SoftwareTrigger mode by ‘TriggerSoftware’.
9. Receive Image
10. Stop and Close image stream.

```
// GenICam node handle
CAM_NODE_HANDLE   hMode = NULL;
CAM_NODE_HANDLE   hSequence = NULL;
CAM_NODE_HANDLE   hSource = NULL;
CAM_NODE_HANDLE   hAdditionalParameter = NULL;
CAM_NODE_HANDLE   hDelay = NULL;
CAM_NODE_HANDLE   hTriggerSoftware = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "TriggerMode", &hMode);
Nd_GetNode(s_hCam, "TriggerSequence", &hSequence);
Nd_GetNode(s_hCam, "TriggerSource", &hSource);
Nd_GetNode(s_hCam, "TriggerAdditionalParameter", &hAdditionalParameter);
Nd_GetNode(s_hCam, "TriggerDelay", &hDelay);
Nd_GetNode(s_hCam, "TriggerSoftware", &hTriggerSoftware);

// 1. Select a trigger mode by 'TriggerMode'.
Nd_SetEnumStrValue(s_hCam, hMode, "On");

// 2. Select a trigger sequence of random trigger shutter by 'TriggerSequence'.
Nd_SetEnumStrValue(s_hCam, hSequence, "TriggerSequence6");   // Bulk mode

// 3. Select a trigger source of random trigger shutter by 'TriggerSource'.
Nd_SetEnumStrValue(s_hCam, hSource, "Software");             // Software

// 4.Select a trigger activation of hardware trigger by 'LineInverterAll'.
//    -- not applicable for Software Trigger

// 5.Set the number of frames to exposure in Bulk mode by
'TriggerAdditionalParameter'.
Nd_SetIntValue(s_hCam, hAdditionalParameter, 3);         // 3 frames

// 6.Set a trigger delay by 'TriggerDelay'.
Nd_SetFloatValue(s_hCam, hDelay, 1000.0);                // 1ms

// 7.Open and Start image stream.
//    7.1.Open Stream
        s_hStrmEvt = CreateEvent(NULL, FALSE, FALSE, NULL);
        Strm_OpenSimple(s_hCam, &s_hStrm, &s_uiImgBufSize, s_hStrmEvt);
        s_pucImgBuf = (uint8_t *)VirtualAlloc(NULL, s_uiImgBufSize, MEM_RESERVE
| MEM_COMMIT, PAGE_EXECUTE_READWRITE);
//    7.2.Stream Start
        Strm_Start(s_hStrm);

// 8.Execute software trigger in SoftwareTrigger mode by 'TriggerSoftware'.
Nd_CmdExecute(s_hCam,hTriggerSoftware);

// 9.Receive Image
uint32_t uiSize    = s_uiImgBufSize;
CAM_IMAGE_INFO sImageInfo;
for(int i=0; i<3; i++)
{
        WaitForSingleObject(s_hStrmEvt, 1000);
        Strm_ReadCurrentImage(s_hStrm, s_pucImgBuf, &uiSize, &sImageInfo);
}

// 10.Stop and Close image stream.
//    10.1.Stream Stop
        Strm_Stop(s_hStrm);
//    10.2.Close Stream
        Strm_Close(s_hStrm);
        CloseHandle(s_hStrmEvt);
        VirtualFree(s_pucImgBuf, 0, MEM_RELEASE);
```

Please refer to [INode functions], [IEnumeration node functions][ICommand node functions][IInteger node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Trigger features by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

1. Select a trigger mode.

   Write to 'Value' field of 'TriggerMode' register to control TriggerMode.
2. Select a trigger sequence of random trigger shutter.

   Write to 'Value' field of 'TriggerSequence' register to control TriggerSequence.
3. Select a trigger source of random trigger shutter.

   Write to 'Value' field of 'TriggerSource' register to control TriggerSource.
4. Select a trigger activation of hardware trigger.
5. Set the number of frames to exposure in Bulk mode.

   Write to 'Value' field of 'TriggerAdditionalParameter' register to control TriggerAdditionalParameter.
6. Set a trigger delay.

   Write to 'Value' field of 'TriggerDelay' register to control TriggerDelay.
7. Open and Start image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.
8. Execute software trigger.

   Write [8] to 'Value' field of 'SoftwareTrigger' register to execute Software trigger..
9. Receive Image
10. Stop and Close image stream.

```
uint32_t       dat;

// 1. Select a trigger mode by 'TriggerMode'.
dat = 1;
Cam_WriteReg(s_hCam, 0x20703C, 1, &dat); // TriggerMode = On

// 2. Select a trigger sequence of random trigger shutter by 'TriggerSequence'.
dat = 6;
Cam_WriteReg(s_hCam, 0x20705C, 1, &dat); // TriggerSequence6 (Bulk mode)

// 3. Select a trigger source of random trigger shutter by 'TriggerSource'.
dat = 64;
Cam_WriteReg(s_hCam, 0x20707C, 1, &dat); // Software

// 4.Select a trigger activation of hardware trigger by 'LineInverterAll'.
//    -- not applicable for Software Trigger

// 5.Set the number of frames to exposure in Bulk mode by
'TriggerAdditionalParameter'.
dat = 3;
Cam_WriteReg(s_hCam, 0x20709C, 1, &dat); // 3 frames

// 6.Set a trigger delay by 'TriggerDelay'.
// TriggerDelay = 1000.0us (Raw value = 60000)
dat = 60000;
Cam_WriteReg(s_hCam, 0x2070BC, 1, &dat); // 1ms

// 7.Open and Start image stream.
//    7.1.Open Stream
        s_hStrmEvt = CreateEvent(NULL, FALSE, FALSE, NULL);
        Strm_OpenSimple(s_hCam, &s_hStrm, &s_uiImgBufSize, s_hStrmEvt);
        s_pucImgBuf = (uint8_t *)VirtualAlloc(NULL, s_uiImgBufSize, MEM_RESERVE
| MEM_COMMIT, PAGE_EXECUTE_READWRITE);
//    7.2.Stream Start
        Strm_Start(s_hStrm);

// 8.Execute software trigger in SoftwareTrigger mode by 'SoftwareTrigger'.
dat = 8;
Cam_WriteReg(s_hCam, 0x2070DC, 1, &dat); // SoftwareTrigger

// 9.Receive Image
uint32_t uiSize    = s_uiImgBufSize;
CAM_IMAGE_INFO sImageInfo;
for(int i=0; i<3; i++)
{
        WaitForSingleObject(s_hStrmEvt, 1000);
        Strm_ReadCurrentImage(s_hStrm, s_pucImgBuf, &uiSize, &sImageInfo);
}

// 10.Stop and Close image stream.
//    10.1.Stream Stop
        Strm_Stop(s_hStrm);
//    10.2.Close Stream
        Strm_Close(s_hStrm);
        CloseHandle(s_hStrmEvt);
        VirtualFree(s_pucImgBuf, 0, MEM_RELEASE);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆**Minimum/Maximum Value**

TriggerAdditional Parameter	Raw Value = Absolute (Float) Value
Minimum	0
Maximum	255
Initial Value	0

TriggerDelay	Raw Value	Absolute (Float) Value
Minimum	0	0.0 [us]
Maximum	120000000	2000000.0 [us]
Initial Value	0	0.0 [us]
Formula	Absolute Value [us] = Raw Value / 60	

● Note

In SoftwareTrigger operation, the delay time from ‘TriggerSoftware’ to exposure is not guaranteed.

BU series have different hardware trigger source.

Please refer to ‘I/O Specification’ section as well.

TriggerSource	Description
Line0 (*)	I/O connector : 4 pin, High level 2.0 to 24.0V
Line2	I/O connector : 1 pin, High level 4.0 to 5.0V

* initial factory setting

TriggerDelay is applied to both of HardwareTrigger and SoftwareTrigger.

TriggerAdditionalParameter is only effective in Bulk mode.

ExposureTime

ExposureTime controls the duration where the image sensor is exposed to light.

This camera series provides three kinds of exposure time control mode.

Color models have ShortExposureMode which can set high-speed exposure time at the time of MANUAL setting. AutoExposure is not supported when the ShortExposureMode is enable (ON).

- NoSpecify : The exposure time is determined by 'AcquisitionFrameRate' register setting value
- Manual : The exposure time is determined by 'ExposureTime' register setting value.
- Auto : The exposure time is adjusted automatically.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ExposureTime	IFloat	4	R/W	Sets an exposure time.
ExposureAuto	IEnumeration	4	R/W	Selects exposure time control mode.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ExposureTime	Implemented	0x204020	[31]	R	Returns the state whether the function is implemented.
	Control	0x204028	4	R/W	[0]: NoSpecify [1]: Manual [2]: Auto
	Mult	0x20402C	4	R	Absolute Value [sec] = Raw * (Mult / Div)
	Div	0x204030	4	R	
	Min	0x204034	4	R	Returns the minimum value of ExposureTime Value register.
	Max	0x204038	4	R	Returns the maximum value of ExposureTime Value register.
	Value	0x20403C	4	R/W	Sets Time value. (Raw value)
ShortExposureMode	Implemented	0x204380	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20438C	4	R	[0] : Off [1] : On
	Value	0x20439C	4	R/W	Only Color models. Sets the activation of ShortExposureMode.

● Control with TeliCamSDK

Camera feature API

Control ExposureTime using dedicated API.

API name	Description
GetCamExposureTimeMinMax	Get minimum and maximum value
GetCamExposureTime	Get current ExposureTime value
SetCamExposureTime	Set new ExposureTime value
GetCamExposureTimeControl	Get current ExposureTime mode
SetCamExposureTimeControl	Set new ExposureTime mode

ExposureTimeControl parameter	Description
CAM_EXPOSURE_TIME_CONTROL_NO_SPECIFY	NoSpecify
CAM_EXPOSURE_TIME_CONTROL_MANUAL	Manual
CAM_EXPOSURE_TIME_CONTROL_AUT	Auto

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control ExposureTime using GenICam API.

◆ExposureTime

Sets an exposure time by ‘ExposureTime’

```
// GenICam node handle
CAM_NODE_HANDLE    hMode = NULL;
CAM_NODE_HANDLE    hSelector = NULL;
CAM_NODE_HANDLE    hSource = NULL;

// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// ExposureTime = 1000us
float64_t       dExposureTime = 1000.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "ExposureTime", &hNode);

// Set ExposureTime Value
Nd_SetFloatValue(s_hCam, hNode, dExposureTime);
```

◆ExposureAuto

Selects an exposure time mode by ‘ExposureAuto’

Integer value and string value of Enumeration are as follows.

Integer	String	Description
0	DeviceSpecific	Operates NoSpecify mode.
1 (*)	Off	Operates Manual mode.
2	Continuous	Operates Auto mode.

* initial factory setting

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "ExposureAuto", &hNode);

// Set ExposureAuto = "Continuous "
Nd_SetEnumStrValue(s_hCam, hNode, "Continuous ");
```

Please refer to [INode functions], [IFloat node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Gain by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆ExposureTime

Write to ‘Value’ field of ‘ExposureTime’ register.

```
// ExposureTime = 1000us (Raw value = 60000)
uint32_t      uiExposureTimeRaw = 60000;

// Set ExposureTime Value
Cam_WriteReg(s_hCam, 0x20403C, 1, &uiExposureTimeRaw);
```

◆ExposureAuto

Write to ‘Control’ field of ‘ExposureTime’ register to control ExposureAuto.

```
// ExposureAuto = "Continuous "
uint32_t      uiExposureAuto = 2;

// Set ExposureAuto = "Continuous "
Cam_WriteReg(s_hCam, 0x204028, 1, &uiExposureAuto);
```

◆ShortExposureMode

Write to ‘Value’ field of ‘ShortExposureMode’ register to control ShortExposureMode.

```
// Set ShortExposureMode ON
uint32_t      dat = 1;
Cam_WriteReg(s_hCam, 0x20439C, 1, &dat);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

ShortExposureMode=OFF

ExposureTime			BU040M	BU160M	BU040MCG	BU160MCG
Minimum		Raw Value	1800		1200	
		Absolute Value	30 [us]		20 [us]	
Maximum	Manual	Raw Value	960000000		960000000	
		Absolute Value	16000000 [us]		16000000 [us]	
	Auto	Raw Value	60000000		60000000	
		Absolute Value	1000000 [us]		1000000 [us]	
Initial Value		Raw Value	120000	240000	120000	240000
		Absolute Value	2000.00 [us]	4000.00 [us]	2000.00 [us]	4000.00 [us]
Formula			Absolute Value [us] = Raw Value / 60			

ShortExposureMode=ON

ExposureTime			BU040M	BU160M	BU040MCG	BU160MCG
Minimum		Raw Value	Incompatible		20	
		Absolute Value			1.08 [us]	
Maximum	Manual	Raw Value			247	
		Absolute Value			13.31 [us]	
	Auto	Raw Value			Incompatible	
		Absolute Value				
Initial Value		Raw Value			247	
		Absolute Value			13.31 [us]	
Formula					Absolute Value [s] = Raw Value * 53 / 983850009	

Notes on ShortExposureMode setting:

- If you use ShortExposureMode, the image quality may deteriorate. Also, the actual exposure time may vary depending on the individual differences and the operating environment (such as the operating temperature). When using the ShortExposureMode, I ask you to have final image quality checked with your environment.
- The brightness of the upper part of the screen may be different from that of the lower part. Note that this is a characteristic of a CMOS image sensor and is not a fault.

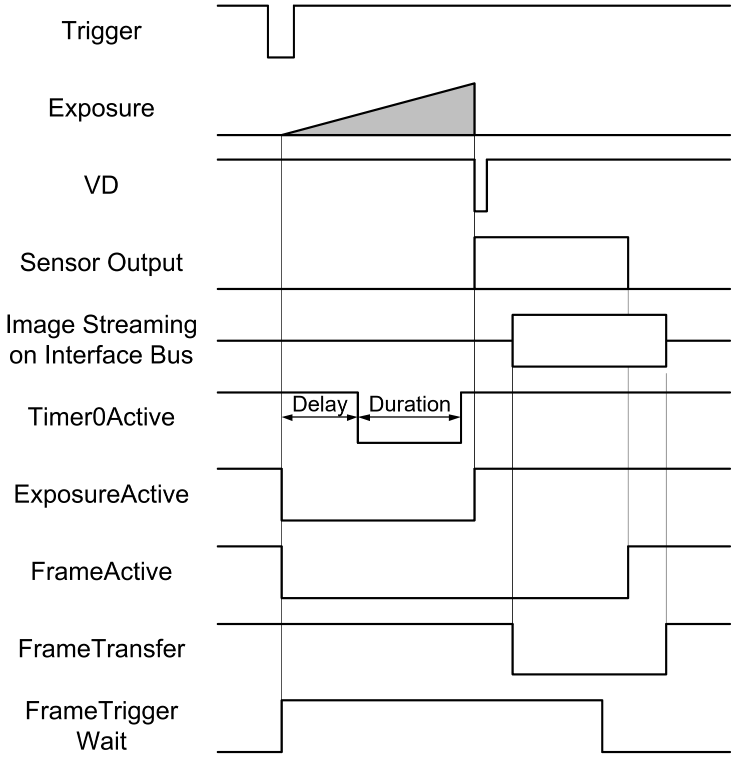
● **Note**

- In ExposureAuto mode, current exposure time will be reflected to ExposureTime value register.
- Changing ‘ShortExposureMode’ register value is invalid during image stream data output.

DigitalIOControl

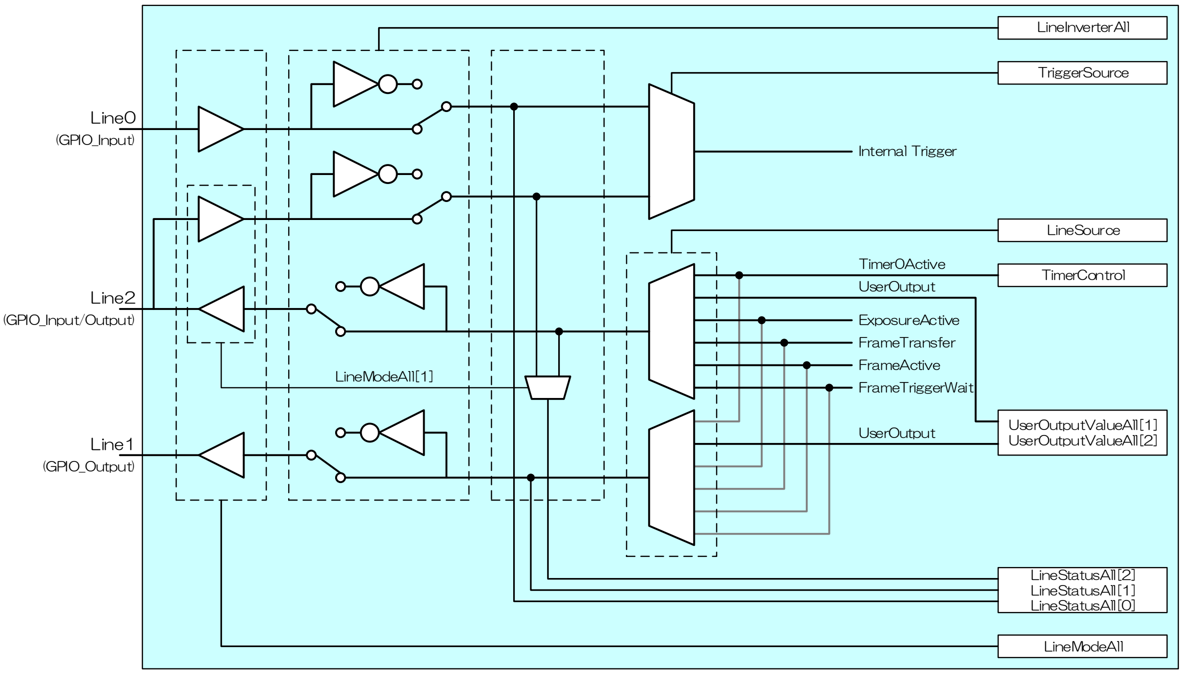
This section describes DigitalIOControl category for the BU series.

This camera provides GPIO output selected by the register setting. And the polarity of the signal is able to switch by the register setting. The following chart shows the specifications of the selectable signals.

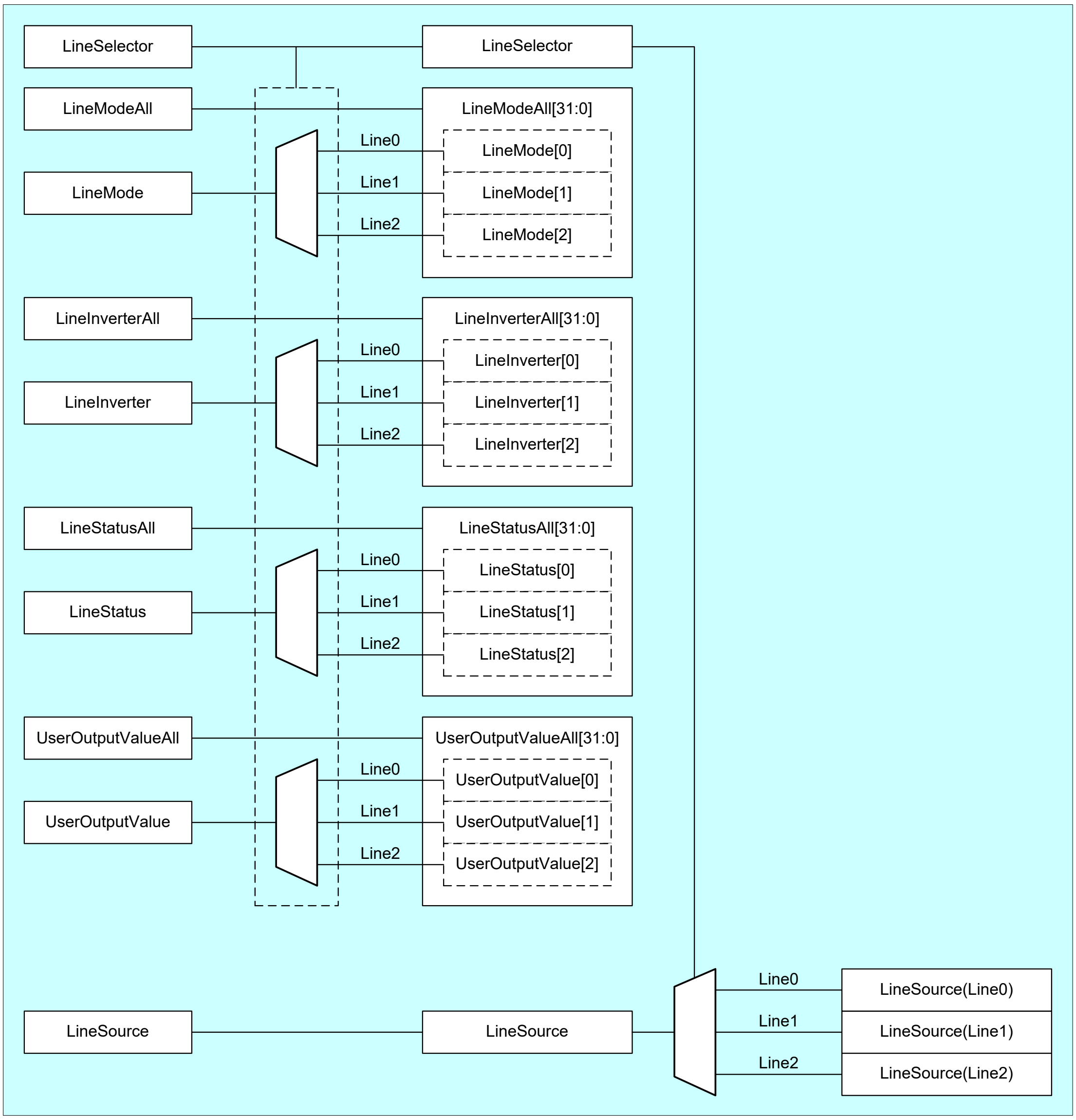


*AcquisitionActive is active during AcquisitionStart state.
* All signals in this figure are Active Low.

Selectable signals



GPIO internal circuit diagram



Details of each signal

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
LineMode	IEnumeration	4	R/W	Selects the Input / Output of each Line selected by LineSelector.
LineModeAll	IInteger	4	R/W	Selects the Input / Output of each Line.
LineInverter	IBoolean	4	R/W	Selects the polarity of each Line signal selected by LineSelector.
LineInverterAll	IInteger	4	R/W	Selects the polarity of each Line signal.
LineStatus	IBoolean	4	R	Returns the status of each Line signal selected by LineSelector.
LineStatusAll	IInteger	4	R	Returns the status of each Line signal.
UserOutputValue	IBoolean	4	R/W	Sets the user output value of each line selected by LineSelector.
UserOutputValueAll	IInteger	4	R/W	Sets the user output value of each Line.
LineSelector	IEnumeration	4	R/W	Selects the Line of I/O connector.
LineSource	IEnumeration	4	R/W	Selects the source of the output signal.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
LineModeAll	Implemented	0x209020	[31]	R	Returns the state whether the function is implemented.
	BitWritable	0x20902C	4	R	[0]: Line0 [1]: Line1 [2]: Line2
	Value	0x209030	4	R/W	Selects the direction of each line signal.
LineInverterAll	Implemented	0x209040	[31]	R	Returns the state whether the function is implemented.
	Value	0x209050	4	R/W	Selects the inversion of I/O lines. Reflected both input and output buffers.
LineStatusAll	Implemented	0x209060	[31]	R	Returns the state whether the function is implemented.
	Value	0x209070	4	R	Returns the current status of all I/O lines.
UserOutputValueAll	Implemented	0x209080	[31]	R	Returns the state whether the function is implemented.
	Value	0x209090	4	R/W	Selects the output value.
LineSelector	Implemented	0x2090A0	[31]	R	Returns the state whether the function is implemented.
	Value	0x2090BC	4	R/W	Selects the I/O line to output LineSource.
LineSource	Implemented	0x2090C0	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x2090CC	4	R	[0] : Off [32] : UserOutput [64] : Timer0Active [99] : AcquisitionActive [106] : FrameTriggerWait [107] : FrameActive [115] : FrameTransferActive [123] : ExposureActive
	Value	0x2090DC	4	R/W	Selects the source signal.

- **Control with TeliCamSDK**

Camera feature API

Control DigitalIOControl using dedicated API.

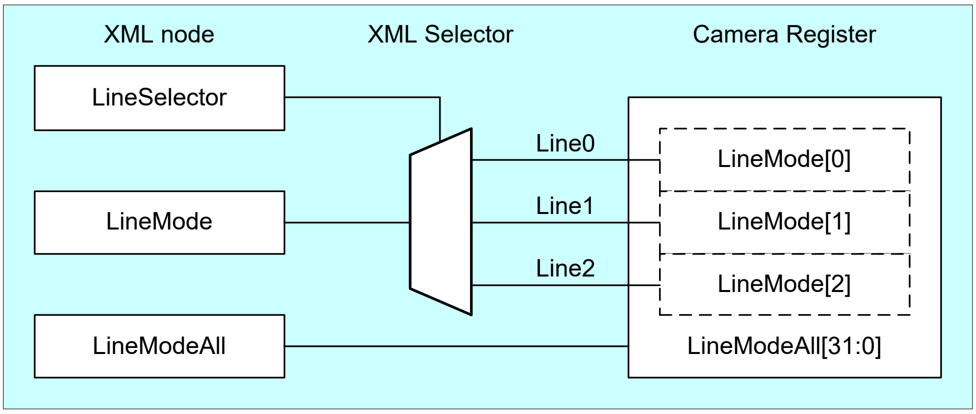
API name	Description
GetCamLineModeAll	Get current LineMode
GetCamLineInverterAll	Get current LineInverter
SetCamLineInverterAll	Set new LineInverter
GetCamLineStatusAll	Get current LineStatus
GetCamUserOutputValueAll	Get current UserOutput
SetCamUserOutputValueAll	Set new UserOutput
GetCamLineSource	Get current LineSource
SetCamLineSource	Set new LineSource

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control DigitalIOControl using GenICam API.

◆LineModeAll



Selects the Input / Output of each Line by ‘LineModeAll’.

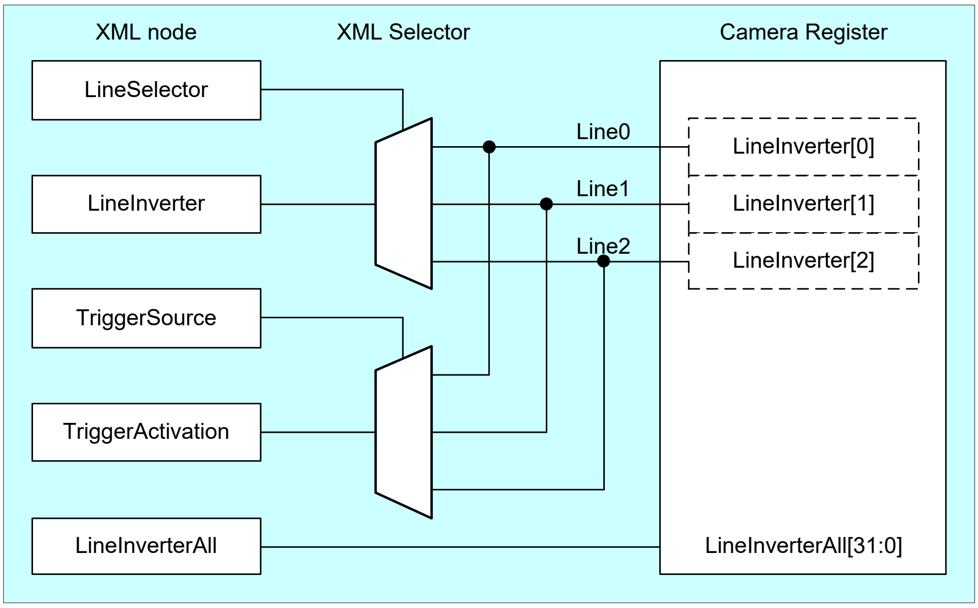
Each bit of integer value corresponds to each Line (bit0=Line0(not changeable), bit1=Line1(not changeable), bit2=Line2).

bit value	I/O Direction
0	Input
1	Output

LineModeAll Value	bit value [Line2] [Line1] [Line0]	I/O Direction [Line2] [Line1] [Line0]
0	not used	
1		
2(*)	[0] [1] [0]	[in] [out] [in]
3	not used	
4		
5		
6	[1] [1] [0]	[out] [out] [in]
7	not used	

* initial factory setting

◆LineInverterAll



Selects the polarity of each Line signal by 'LineInverterAll'.

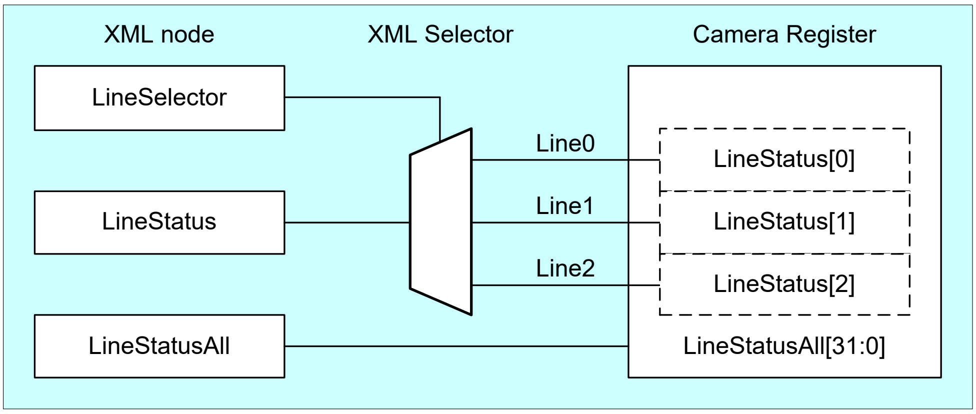
Each bit of integer value corresponds to each Line (bit0=Line0, bit1=Line1, bit2=Line2).

bit value	Inverter
0	False (not inverted)
1	True (inverted)

LineInverterAll Value	bit value [Line2] [Line1] [Line0]	Inverter [Line2] [Line1] [Line0]
0(*)	[0] [0] [0]	[off] [off] [off]
1	[0] [0] [1]	[off] [off] [on]
2	[0] [1] [0]	[off] [on] [off]
3	[0] [1] [1]	[off] [on] [on]
4	[1] [0] [0]	[on] [off] [off]
5	[1] [0] [1]	[out [off] [on]
6	[1] [1] [0]	[on] [on] [off]
7	[1] [1] [1]	[on] [on] [on]

* initial factory setting

◆LineStatusAll



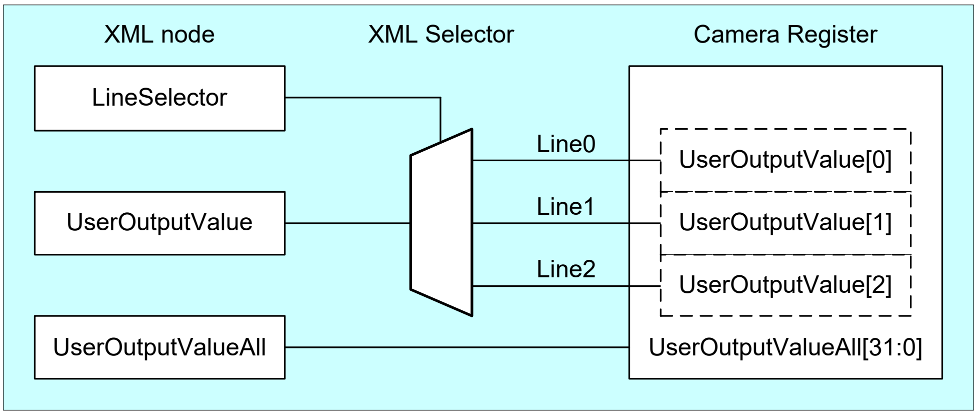
Gets the status of each Line signal by 'LineStatusAll'

Each bit of integer value corresponds to each Line (bit0=Line0, bit1=Line1, bit2=Line2).

bit value	Line Status
0	False(Low)
1	True(High)

LineStatusAll	bit value	Line Status
Value	[Line2] [Line1] [Line0]	[Line2] [Line1] [Line0]
0	[0] [0] [0]	[low] [low] [low]
1	[0] [0] [1]	[low] [low] [high]
2	[0] [1] [0]	[low] [high] [low]
3	[0] [1] [1]	[low] [high] [high]
4	[1] [0] [0]	[high] [low] [low]
5	[1] [0] [1]	[high] [low] [high]
6	[1] [1] [0]	[high] [high] [low]
7	[1] [1] [1]	[high] [high] [high]

◆**UserOutputValueAll**



Sets the user output value by ‘UserOutputValueAll’.

Each bit of integer value corresponds to each Line (bit0=Line0(not used), bit1=Line1, bit2=Line2).

bit value	Output
0	False(Low)
1	True(High)

UserOutputValueAll	bit value [Line2] [Line1]	Output [Line2] [Line1]
0 (*)	[0] [0]	[low] [low]
1		
2	[0] [1]	[low] [high]
3		
4	[1] [0]	[high] [low]
5		
6	[1] [1]	[high] [high]
7		

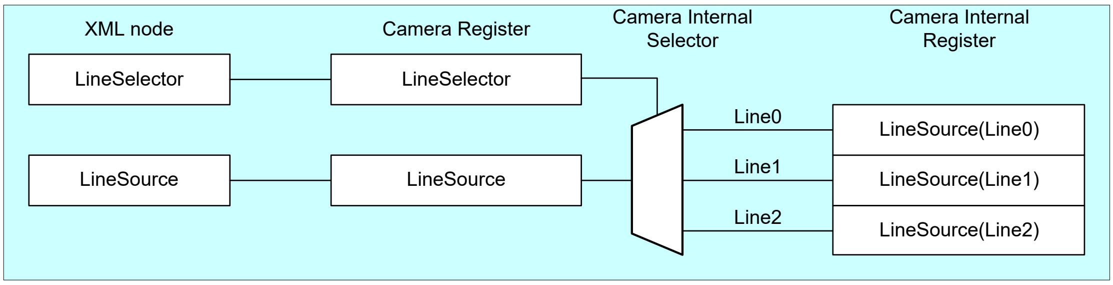
◆**LineSelector**

Selects the I/O line to output by ‘LineSelector’.

Integer value and string value of Enumeration are as follows.

Integer	String
1	Line1
2	Line2

◆LineSource



Selects the source of the output signal.

Integer value and string value of Enumeration are as follows.

Integer	String	Description
0	Off	Off
32	UserOutput	Outputs the value set in ‘UserOutputValueAll’.
64	Timer0Active	This signal can be used as strobe control signal. The delay time and pulse width of this signal are configurable.
99	AcquisitionActive	Indicates AcquisitionStart state of camera.
106	FrameTriggerWait	Indicates that camera is ready to accept trigger signal. (both hardware and software)
107	FrameActive	Period from exposure start to sensor read-out completion.
115	FrameTransferActive	Period of transferring image streaming data on interface bus.
123	ExposureActive	Period from exposure start to exposure end.

```
// GenICam node handle
CAM_NODE_HANDLE    hMode = NULL;
CAM_NODE_HANDLE    hSelector = NULL;
CAM_NODE_HANDLE    hSource = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "LineModeAll", &hMode);
Nd_GetNode(s_hCam, "LineSelector", &hSelector);
Nd_GetNode(s_hCam, "LineSource", &hSource);

// Line2/Line1 = output, Line0 = input
int64_t       Mode = 6;
Nd_SetIntValue(s_hCam, hMode, Mode);

// Line1 = ExposureActive
Nd_SetEnumStrValue(s_hCam, hSelector, "Line1");
Nd_SetEnumStrValue(s_hCam, hSource, "ExposureActive");

// Line2 = FrameTransferActive
Nd_SetEnumStrValue(s_hCam, hSelector, "Line2");
Nd_SetEnumStrValue(s_hCam, hSource, "FrameTransferActive");
```

Please refer to [INode functions], [IInteger node functions], [IBoolean node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control DigitalIOControl by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆LineModeAll

Write to ‘Value’ field of ‘LineModeAll’ register.

◆LineInverterAll

Write to ‘Value’ field of ‘LineInverterAll’ register.

◆LineStatusAll

Read ‘Value’ field of ‘LineStatusAll’ register.

◆UserOutputValueAll

Write to ‘Value’ field of ‘UserOutputAll’ register.

◆LineSelector

Write to ‘Value’ field of ‘LineSelector’ register.

◆LineSource

Write to ‘Value’ field of ‘LineSource’ register.

```
// Line2/Line1 = output, Line0 = input
uint32_t      uiMode = 6;
Cam_WriteReg(s_hCam, 0x209030, 1, &uiMode);

// Set Value
uint32_t      uiSelector;
uint32_t      uiSource;
uiSelector = 1;  // Line1
uiSource = 123;       // ExposureActive
Cam_WriteReg(s_hCam, 0x2090BC, 1, &uiSelector);
Cam_WriteReg(s_hCam, 0x2090DC, 1, &uiSource);

uiSelector = 2;  // Line2
uiSource = 115;// FrameTransferActive
Cam_WriteReg(s_hCam, 0x2090BC, 1, &uiSelector);
Cam_WriteReg(s_hCam, 0x2090DC, 1, &uiSource);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

● Note

- Line0 is dedicated input.
- Line1 is dedicated output.
- Line2 is bi-directional. The default direction is input.

For [Timer0Active] signal, please refer to [TimerControl] Section for more detail.

AntiGlitch / AntiChattering

AntiGlitch and AntiChattering functions filter noise and unstable state of the digital input (trigger signal).

AntiGlitch circuit performs the digital integration of the trigger signal.

It is effective to remove impulsive noise.

AntiChattering circuit sets the edge insensible time to avoid trigger malfunction.

It is effective to remove unstable logic state and switch-chattering.

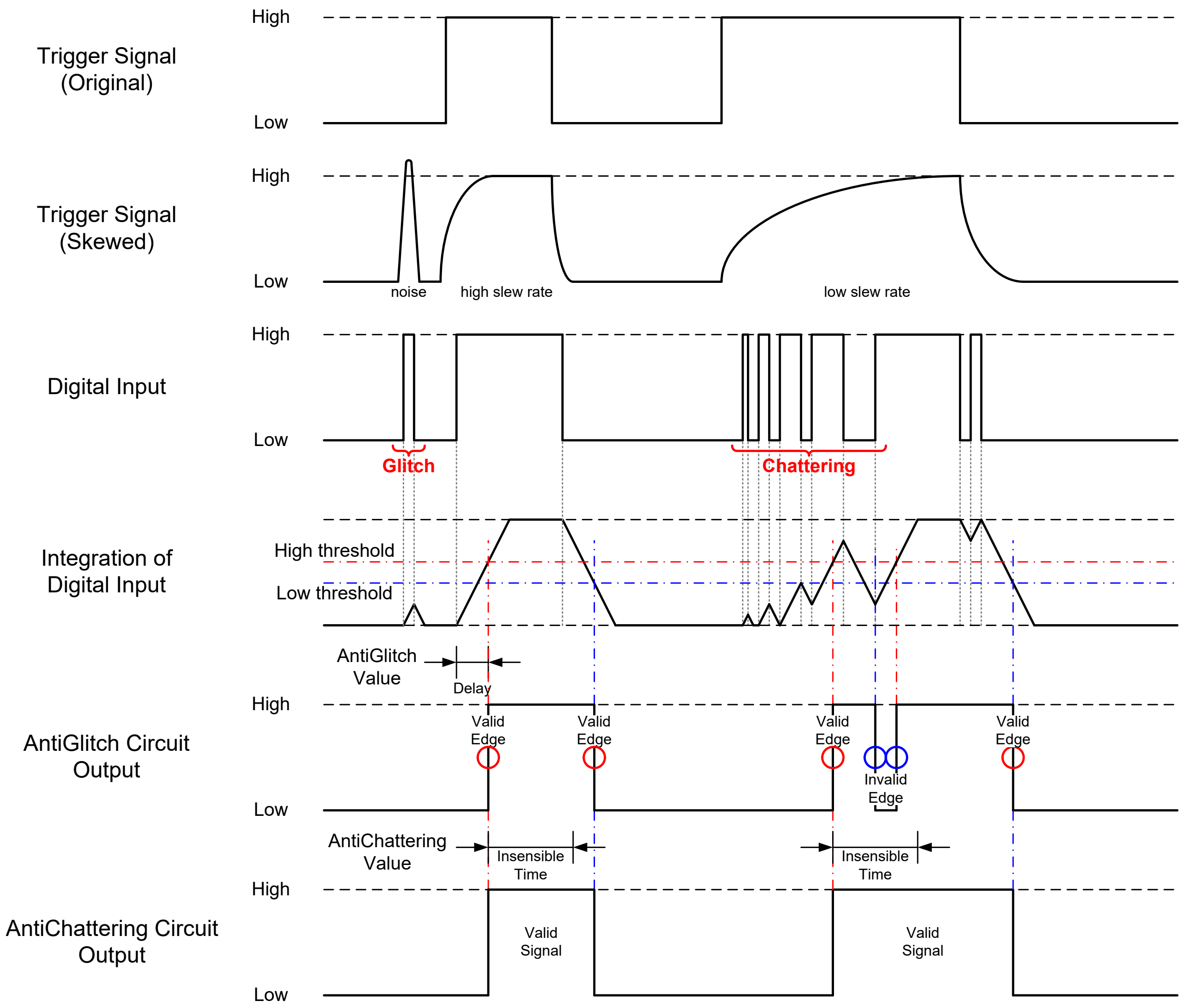


Fig. AntiGlitch and AntiChattering

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
AntiGlitch	IFloat	4	R/W	Integration time of digital input signal. [sec]
AntiChattering	IFloat	4	R/W	Insensible time of digital input signal. [sec]

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
AntiGlitch	Implemented	0x21F3C0	[31]	R	Returns the state whether the function is implemented.
	Mult	0x21F3CC	4	R	Absolute Value [sec] = Raw * (Mult / Div)
	Div	0x21F3D0	4	R	
	Min	0x21F3D4	4	R	Returns the minimum value of AntiGlitch period.
	Max	0x21F3D8	4	R	Returns the maximum value of AntiGlitch period.
	Value	0x21F3DC	4	R/W	Sets the AntiGlitch period.(Raw value)
AntiChattering	Implemented	0x21F3E0	[31]	R	Returns the state whether the function is implemented.
	Mult	0x21F3EC	4	R	Absolute Value [sec] = Raw * (Mult / Div)
	Div	0x21F3F0	4	R	
	Min	0x21F3F4	4	R	Returns the minimum value of AntiGlitch period.
	Max	0x21F3F8	4	R	Returns the maximum value of AntiGlitch period.
	Value	0x21F3FC	4	R/W	Sets the AntiGlitch period.(Raw value)

● Control with TeliCamSDK

GenICam function API

Control AntiGlitch/AntiChattering using GenICam API.

◆AntiGlitch/AntiChattering

1. Set Integration time of digital input signal in [sec] to ‘AntiGlitch'.
2. Set Insensible time of digital input signal in [sec] to ‘AntiChattering'.

```
// GenICam node handle
CAM_NODE_HANDLE    hGlitch = NULL;
CAM_NODE_HANDLE    hChattering = NULL;

// AntiGlitch = 1.0[us], AntiChattering = 10.0[us]
float64_t       dGlitch = 0.000001;
float64_t       dChattering = 0.000010;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “AntiGlitch”, &hGlitch);
Nd_GetNode(s_hCam, “AntiChattering”, &hChattering);

// Set Value
Nd_SetFloatValue(s_hCam, hGlitch, dGlitch);
Nd_SetFloatValue(s_hCam, hChattering, dChattering);
```

Please refer to [INode functions], [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control AntiGlitch/AntiChattering by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆AntiGlitch/AntiChattering

1. Write to ‘Value’ field of ‘AntiGlitch’ register.
2. Write to ‘Value’ field of ‘AntiChattering’ register.

```
// AntiGlitch = 1.0[us] (Raw value = 100)
// AntiChattering = 10.0[us] (Raw value = 1250)
uint32_t      uiAntiGlitchRaw = 100;
uint32_t      uiAntiChatteringRaw = 1250;

// Set Value
Cam_WriteReg(s_hCam, 0x21F3DC, 1, &uiAntiGlitchRaw);
Cam_WriteReg(s_hCam, 0x21F3FC, 1, &uiAntiChatteringRaw);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

AntiGlitch	Raw Value	Absolute (Float) Value
Minimum	9	0.00000009 [s]
Maximum	200000	0.002 [s]
Initial Value	9	0.00000009 [s]
Formula	Absolute Value [us] = Raw Value / 100000000	

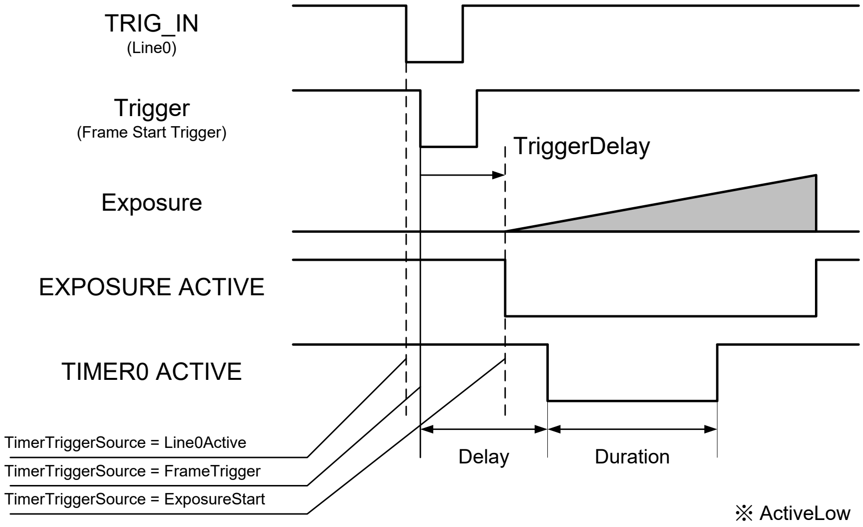
AntiChattering	Raw Value	Absolute (Float) Value
Minimum	249	0.000001992 [s]
Maximum	249999	0.001999992 [s]
Initial Value	249	0.000001992 [s]
Formula	Absolute Value [us] = Raw Value / 125000000	

TimerControl

This section describes TimerControl features.

This camera series is able to generate Timer0Active signal beginning from trigger or exposure start signal.

This signal can be used as strobe control signal.



Timer0Active

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
TimerSelector	IEnumeration	4	R	Returns the selected timer name.
TimerDelay	IFloat	4	R/W	Sets the delay of Timer0Active signal.
TimerDuration	IFloat	4	R/W	Sets the duration of Timer0Active signal.
TimerTriggerSource	IEnumeration	4	R/W	Selects the source of Timer0Active pulse to start.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
TimerSelector	Implemented	0x20A020	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20A02C	4	R	[0] : Timer0
	Value	0x20A03C	4	R/W	Returns the selected timer.
TimerDelay	Implemented	0x20A040	[31]	R	Returns the state whether the function is implemented.
	Mult	0x20A04C	4	R	AbsoluteValue[sec] = Raw * (Mult / Div)
	Div	0x20A050	4	R	
	Min	0x20A054	4	R	Returns the minimum value of TimerDelay Value register.
	Max	0x20A058	4	R	Returns the maximum value of TimerDelay Value register.
	Value	0x20A05C	4	R/W	Sets Timer0Active delay time.(Raw value)
TimerDuration	Implemented	0x20A060	[31]	R	Returns the state whether the function is implemented.
	Mult	0x20A06C	4	R	AbsoluteValue[sec] = Raw * (Mult / Div)
	Div	0x20A070	4	R	
	Min	0x20A074	4	R	Returns the minimum value of TimerDuration Value register.
	Max	0x20A078	4	R	Returns the maximum value of TimerDuration Value register.
	Value	0x20A07C	4	R/W	Sets Timer0Active duration time.(Raw value)
TimerTriggerSource	Implemented	0x20A080	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20A08C	4	R/W	[0] : Off [32] : Line0 [104] : FrameTrigger [124] : ExposureStart
	Value	0x20A09C	4	R/W	Selects the source of Timer0Active pulse to start.

- **Control with TeliCamSDK**

Camera feature API

Control TimerControl using dedicated API.

API name	Description
GetCamTimerDelayMinMax	Get minimum and maximum value
GetCamTimerDelay	Get current TimerDelay value
SetCamTimerDelay	Set new TimerDelay value
GetCamTimerDurationMinMax	Get minimum and maximum value
GetCamTimerDuration	Get current TimerDuration value
SetCamTimerDuration	Set new TimerDuration value
GetCamTimerTriggerSource	Get current TimerTriggerSource
SetCamTimerTriggerSource	Set new TimerTriggerSource

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control TimerControl using GenICam API.

◆TimerDelay/TimerDuration/TimerTriggerSource

1. Sets the delay of Timer0Active signal by ‘TimerDelay’.
2. Sets the duration of Timer0Active signal by ‘TimerDuration’.
3. Selects the source of Timer0Active pulse to start by ‘TimerTriggerSource’.

   Integer value and string value of Enumeration are as follows.

Integer	String	Description
0	Off	DisablesTimer0Active signal.
32	Line0	Starts when Line0 is active.
104	FrameTrigger	Starts with the reception of the Frame Start Trigger.
124	ExposureStart	Starts with the reception of the Exposure Start

```
// GenICam node handle
CAM_NODE_HANDLE    hDelay = NULL;
CAM_NODE_HANDLE    hDuration = NULL;
CAM_NODE_HANDLE    hSource = NULL;

// TimerDelay = 1000.0[us], TimerDuration = 2000.0[us]
float64_t       dDelay = 1000.0;
float64_t       dDuration = 2000.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "TimerDelay", &hDelay);
Nd_GetNode(s_hCam, "TimerDuration", &hDuration);
Nd_GetNode(s_hCam, "TimerTriggerSource", &hSource);

// 1.Sets the delay of Timer0Active signal.
Nd_SetFloatValue(s_hCam, hDelay, dDelay);

// 2.Sets the dulation of Timer0Active signal.
Nd_SetFloatValue(s_hCam, hDuration, dDuration);

// 3.Selects the source of Timer0Active pulse to start.
Nd_SetEnumStrValue(s_hCam, hSource, "ExposureStart");
```

Please refer to [INode functions], [IFloat node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control TimerControl by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆TimerDelay/TimerDuration/TimerTriggerSource

1. Write to ‘Value’ field of ‘TimerDelay’ register.
2. Write to ‘Value’ field of ‘TimerDuration’ register.
3. Write to ‘Value’ field of ‘TimerTriggerSource’ register.

```
// TimerDelay = 1000.0[us] (Raw value = 125000)
// TimerDuration = 2000.0[us] (Raw value = 250000)
// TimerTriggersource = "ExposureStart"
uint32_t     uiTimerDelayRaw = 125000;
uint32_t     uiTimerDurationRaw = 250000;
uint32_t     uiTimerTriggerSource = 124;

// Set Value
Cam_WriteReg(s_hCam, 0x20A05C, 1, &uiTimerDelayRaw);
Cam_WriteReg(s_hCam, 0x20A07C, 1, &uiTimerDurationRaw);
Cam_WriteReg(s_hCam, 0x21F27C, 1, &uiTimerTriggerSource);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

TimerDelay TimerDuration	Raw Value	Absolute (Float) Value
Minimum	0	0 [us]
Maximum	250000000	2000000 [us]
Initial Value	0	0.00 [us]
Formula	Absolute Value [us] = Raw Value / 125	

● Note

‘TimerTriggerSource’ operation is as follows,

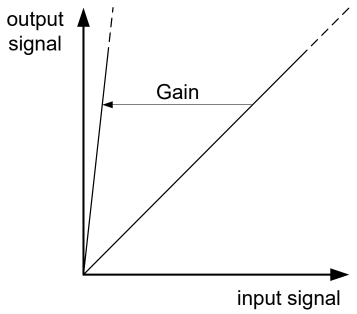
Line0Active is available in Hardware Trigger mode, and to be responded to FrameTriggerError.

FrameTrigger is available in both Hardware Trigger and Software Trigger mode, and not to be responded to FrameTriggerError.

ExposureStart is available in both Hardware Trigger and Software Trigger mode, and not to be responded to FrameTriggerError. Timer0Active pulse delays TiggerDelay+TimerDelay[us].

Gain

This section describes Gain feature. This control adjusts an amplification factor applied to the output signal. Gain feature adjusts manual gain. GainAuto feature adjusts gain automatically.



Gain

The formula of Gain value is as follows;

Output signal = Input signal x 10 ^ (Gain / 20)

Gain is in [dB] unit.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
Gain	IFloat	4	R/W	Sets a manual Gain.
GainAuto	IEnumeration	4	R/W	Sets a Gain operational mode.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
Gain	Implemented	0x204060	[31]	R	Returns the state whether the function is implemented.
	Control	0x204068	4	R/W	[1]: Manual [2] : Auto
	Mult	0x20406C	4	R	Absolute Value [dB] = Raw * (Mult / Div)
	Div	0x204070	4	R	
	Min	0x204074	4	R	Returns the minimum value of Gain Value register.
	Max	0x204078	4	R	Returns the maximum value of Gain Value register.
	Value	0x20407C	4	R/W	Gain value. (Raw value)

● Control with TeliCamSDK

Camera feature API

Control Gain using dedicated API.

API name	Description
GetCamGainMinMax	Get minimum and maximum value
GetCamGain	Get current Gain value
SetCamGain	Set new Gain value
GetCamGainAuto	Get current GainAuto mode
SetCamGainAuto	Set new GainAuto mode

GainAuto parameter	Description
CAM_GAIN_AUTO_OFF	Off
CAM_GAIN_AUTO_AUTO	Adjust continuously

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Gain using GenICam API.

◆Gain

Use ‘IFloat’ interface to control Gain.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Gain = 6.0dB
float64_t       dGain = 6.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “Gain”, &hNode);

// Set Gain Value
Nd_SetFloatValue(s_hCam, hNode, dGain);
```

◆GainAuto

Use ‘IEnumertion’ interface to control GainAuto.

Integer value and string value of Enumeration are as follows.

Integer	String	Description
1	Off	Operates Manual mode.
2	Continuous	Operates Auto mode.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “GainAuto”, &hNode);

// Set GainAuto = “Continuous”
Nd_SetEnumStrValue(s_hCam, hNode, “Continuous”);
```

Please refer to [INode functions], [IFloat node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Gain by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Gain

Write to ‘Value’ field of ‘Gain’ register.

```
// Gain = 6.0dB (Raw value = 60)
uint32_t      uiGainRaw = 60;

// Set Gain Value
Cam_WriteReg(s_hCam, 0x20407C, 1, &uiGainRaw);
```

◆GainAuto

Write to ‘Control’ field of ‘Gain’ register.

```
// GainAuto = “Continuous”
uint32_t      uiGainAuto = 2;

// Set GainAuto = “Continuous”
Cam_WriteReg(s_hCam, 0x204068, 1, &uiGainAuto);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

Gain		BU040M/BU160M	BU040MCG/BU160MCG
Minimum	Raw Value	0	0
	Absolute (Float) Value	0.00 [dB]	0.00 [dB]
Maximum	Raw Value	240	360
	Absolute (Float) Value	24.00 [dB]	36.00 [dB]
Initial Value	Raw Value	0	0
	Absolute (Float) Value	0.00 [dB]	0.00 [dB]
Formula		Absolute Value [dB] = Raw Value / 10	

● Note

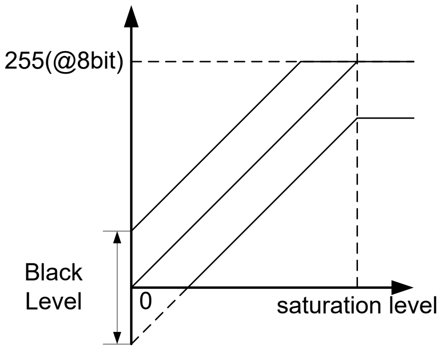
- In GainAuto mode, current Gain will be reflected to Gain value register.

BlackLevel

This section describes BlackLevel feature. This control adjusts the black level applied to the output signal.

It is adjustable from -25% to +25% as white saturation level is 100%.

If BlackLevel is set lower than 0[%], the image level may not be saturated.



Black Level

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
BlackLevel	IFloat	4	R/W	Black Level value.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
BlackLevel	Implemented	0x204040	[31]	R	Returns the state whether the function is implemented.
	Mult	0x20404C	4	R	Absolute Value [%] = Raw * (Mult / Div)
	Div	0x204050	4	R	
	Min	0x204054	4	R	Returns the minimum value of Black Level Value register.
	Max	0x204058	4	R	Returns the maximum value of Black Level Value register.
	Value	0x20405C	4	R/W	Black Level value. (Raw value)

● Control with TeliCamSDK

Camera feature API

Control BlackLevel using dedicated API.

API name	Description
GetCamBlackLevelMinMax	Get minimum and maximum value
GetCamBlackLevel	Get current BlackLevel value
SetCamBlackLevel	Set new BlackLevel value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control BlackLevel using GenICam API.

◆BlackLevel

Use ‘IFloat’ interface to control BlackLevel.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// BlackLevel = 25%
float64_t       dBlackLevel = 25.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “BlackLevel”, &hNode);

// Set BlackLevel Value
Nd_SetFloatValue(s_hCam, hNode, dBlackLevel);
```

Please refer to [INode functions] and [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control BlackLevel by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆BlackLevel

Write to ‘Value’ field of ‘BlackLevel’ register.

```
// BlackLevel = 25% (Raw value = 256)
int32_t       iBlackLevelRaw = 256; // signed

// Set BlackLevel Value
Cam_WriteReg(s_hCam, 0x20405C, 1, &iBlackLevelRaw);
```

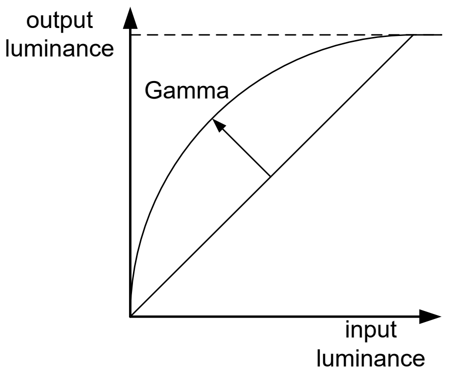
Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

Gain	Raw Value	Absolute (Float) Value
Minimum	-256	-25.00[%]
Maximum	256	+25.00[%]
Initial Value	0	0.00[%]
Formula	Absolute Value [%] = Raw Value * 100 / 1024	

Gamma

This section describes Gamma feature. This control adjusts the gamma correction of pixel intensity.



Gamma correction

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
Gamma	IFloat	4	R/W	Gamma value.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
Gamma	Implemented	0x204080	[31]	R	Returns the state whether the function is implemented.
	Mult	0x20408C	4	R	Absolute Value = Raw * (Mult / Div)
	Div	0x204090	4	R	
	Min	0x204094	4	R	Returns the minimum value of Gamma Value register.
	Max	0x204098	4	R	Returns the maximum value of Gamma Value register.
	Value	0x20409C	4	R/W	Gamma value. (Raw value)

● Control with TeliCamSDK

Camera feature API

Control Gamma using dedicated API.

API name	Description
GetCamGammaMinMax	Get minimum and maximum value
GetCamGamma	Get current Gamma value
SetCamGamma	Set new Gamma value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Gamma using GenICam API.

◆Gamma

Use ‘IFloat’ interface to control Gamma.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Gamma = 0.45
float64_t       dGamma = 0.45;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “Gamma”, &hNode);

// Set Gamma Value
Nd_SetFloatValue(s_hCam, hNode, dGamma);
```

Please refer to [INode functions] and [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Gamma by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Gamma

Write to ‘Value’ field of ‘Gamma’ register.

```
// Gamma = 0.45 (Raw value = 45)
uint32_t       uiGammaRaw = 45;

// Set Gamma Value
Cam_WriteReg(s_hCam, 0x20409C, 1, &uiGammaRaw);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

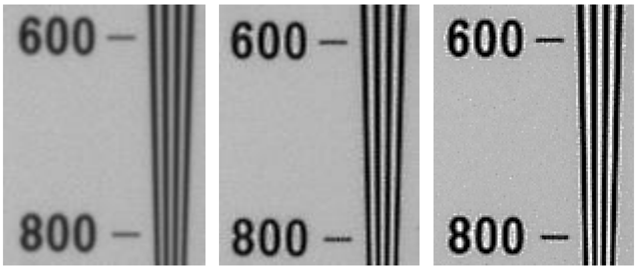
Gain	Raw Value	Absolute (Float) Value
Minimum	45	0.45
Maximum	100	1.00
Initial Value	100	1.00
Formula	Absolute Value = Raw Value / 100	

Sharpness

This section describes Sharpness feature. This control enhances the edges of the image.

The edge enhancement will become stronger when the setting value is larger.

This function is available only in Color models.



Sharpness=0 Sharpness=4 Sharpness=7

Sharpness

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
Sharpness	IInteger	4	R/W	Sets sharpness value.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
Sharpness	Implemented	0x2040A0	[31]	R	Returns the state whether the function is implemented.
	Min	0x2040B4	4	R	Returns the minimum value of sharpness Value register.
	Max	0x2040B8	4	R	Returns the maximum value of sharpness Value register.
	Value	0x2040BC	4	R/W	Sets sharpness value. (Raw value)

● Control with TeliCamSDK

Camera feature API

Control Sharpness using dedicated API.

API name	Description
GetCamSharpnessMinMax	Get minimum and maximum value
GetCamSharpness	Get current Sharpness value
SetCamSharpness	Set new Sharpness value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Sharpness using GenICam API.

◆Sharpness

Use ‘IFloat’ interface to control Sharpness.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Sharpness = 4
int64_t       nSharpness = 4;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "Sharpness", &hNode);

// Set Sharpness Value
Nd_SetFloatValue(s_hCam, hNode, nSharpness);
```

Please refer to [INode functions] and [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Sharpness by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Sharpness

Write to ‘Value’ field of ‘Sharpness’ register.

```
// Sharpness = 4 (Raw value = 4)
uint32_t       uiSharpnessRaw = 4;

// Set Sharpness Value
Cam_WriteReg(s_hCam, 0x2040BC, 1, &uiSharpnessRaw);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

Gain	Raw Value	Absolute (Float) Value
Minimum	0	0
Maximum	7	7
Initial Value	0	0
Formula	Absolute Value = Raw Value / 100	

BalanceRatio

This section describes BalanceRatio feature.

BalanceRatio control adjusts the white balance gain of the image.

BalanceWhiteAuto control adjusts white balance gain automatically.

This function is available only in Color models.

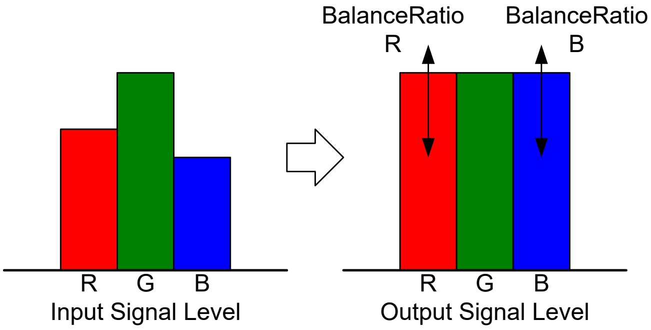


Fig. BalanceRatio

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
BalanceRatioSelector	IEnumeration	4	R/W	Selects a color component of white balance gain to control.
BalanceRatio	IFloat	4	R/W	Sets a white balance gain.
BalanceWhiteAuto	IEnumeration	4	R/W	Adjusts white balance gain automatically.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
WhiteBalanceR	Implemented	0x205060	[31]	R	Returns the state whether the function is implemented.
	Control	0x205068	4	R/W	[1]: Off [3]: Once
	Mult	0x20506C	4	R	Absolute Value [times] = Raw * (Mult / Div)
	Div	0x205070	4	R	
	Min	0x205074	4	R	Returns the minimum value of WhiteBalanceR Value register.
	Max	0x205078	4	R	Returns the maximum value of WhiteBalanceR Value register.
	Value	0x20507C	4	R/W	White balance Red gain (Raw value)
WhiteBalanceB	Implemented	0x205080	[31]	R	Returns the state whether the function is implemented.
	Control	0x205088	4	R/W	[1]: Off [3]: Once
	Mult	0x20508C	4	R	Absolute Value [times] = Raw * (Mult / Div)
	Div	0x205090	4	R	
	Min	0x205094	4	R	Returns the minimum value of WhiteBalanceB Value register.
	Max	0x205098	4	R	Returns the maximum value of WhiteBalanceB Value register.
	Value	0x20509C	4	R/W	White balance Blue gain (Raw value)

● Control with TeliCamSDK

Camera feature API

Control BalanceRatio using dedicated API.

API name	Description
GetCamBalanceRatioMinMax	Get minimum and maximum value
GetCamBalanceRatio	Get current BalanceRatio value
SetCamBalanceRatio	Set new BalanceRatio value
GetCamBalanceWhiteAuto	Get current BalanceWhiteAuto mode
SetCamBalanceWhiteAuto	Set new BalanceWhiteAuto mode

BalanceWhiteAuto parameter	Description
CAM_BALANCE_WHITE_AUTO_OFF	Off
CAM_BALANCE_WHITE_AUTO_ONCE	Adjust once

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control BalanceRatio and BalanceWhiteAuto using GenICam API.

◆BalanceRatio

1. Select a color component of white balance gain to control by ‘BalanceRatioSelector’.

   Integer value and string value of Enumeration are as follows.

Integer	String
1	Red
2	Blue

2. Set a white balance gain by ‘BalanceRatio’.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// 1.Select a color component
// Retrieve GenICam node.
Nd_GetNode(s_hCam, “BalanceRatioSelector”, &hNode);

// Select “Red”
Nd_SetEnumStrValue(s_hCam, hNode, “Red”);

// 2.Set a white balance gain
// white balance R gain = x 1.0
float64_t        dBalanceRatioR = 1.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “BalanceRatio”, &hNode);

// Set Gain Value
Nd_SetFloatValue(s_hCam, hNode, dBalanceRatioR);
```

◆BalanceWhiteAuto

Select ‘BalanceWhiteAuto’ mode by IEnumeration interface.

Integer value and string value of Enumeration are as follows.

Integer	String
1	Off
3	Once

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “BalanceWhiteAuto”, &hNode);

// Set BalanceWhiteAuto = “Once”
Nd_SetEnumStrValue(s_hCam, hNode, “Once”);
```

Please refer to [INode functions], [IEnumeration node functions] and [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control BalanceRatio and BalanceWhiteAuto by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆BalanceRatio

Write to ‘Value’ field of ‘WhiteBalanceR’ or ‘WhiteBalanceB’ register.

```
// WhiteBalance R Gain = x 1 (Raw value = 65536)
// WhiteBalance B Gain = x 1 (Raw value = 65536)
uint32_t      uiWhiteBalanceRRaw = 65536;
uint32_t      uiWhiteBalanceBRaw = 65536;

// Set WhiteBalance Gain Value
Cam_WriteReg(s_hCam, 0x20507C, 1, &uiWhiteBalanceRRaw);
Cam_WriteReg(s_hCam, 0x20509C, 1, &uiWhiteBalanceBRaw);
```

◆BalanceRatioAuto

Write to ‘Control’ field of ‘WhiteBalanceR’ or ‘WhiteBalanceB’ register.

```
// BalanceWhiteAuto = “Once”
uint32_t      uiBalanceWhiteAuto = 3;

// Set BalanceWhiteAuto = “Once”
Cam_WriteReg(s_hCam, 0x205068, 1, &uiBalanceWhiteAuto);
// Cam_WriteReg(s_hCam, 0x205088, 1, &uiBalanceWhiteAuto); // either will do
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

BalanceRatio	Raw Value	Absolute (Float) Value
Minimum	65536	x 1 [times]
Maximum	524287	x 8 [times]
Initial Value	65536	x 1 [times]
Formula	Absolute Value [times] = Raw Value / 65536	

● Note

After executing BalanceWhiteAuto, the result will be reflected to BalanceRatio Red and Blue value.

Notes on white balance gain setting:

Too high white balance gain increases noises. After adjusting the white balance of the image, I ask you to have final image quality checked with your environment.

ColorCorrectionMatrix

This section describes ColorCorrectionMatrix feature.

This control corrects RGB data by using color correction matrix.

This function is available only in Color models.

The relationship between original data (R, G, and B) and corrected data (R', G', and B') are represented in the following formula.

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & -mask\_rg & -mask\_rb \\ -mask\_gr & 1 & -mask\_gb \\ -mask\_br & -mask\_bg & 1 \end{bmatrix} \begin{bmatrix} R & (G-R) & (B-R) \\ (R-G) & G & (B-G) \\ (R-B) & (G-B) & B \end{bmatrix}$$

$$R' = (1 - mask\_rg - mask\_rb) \cdot R + mask\_rg \cdot G + mask\_rb \cdot B$$
$$G' = mask\_gr \cdot R + (1 - mask\_gr - mask\_gb) \cdot G + mask\_gb \cdot B$$
$$B' = mask\_br \cdot R + mask\_bg \cdot G + (1 - mask\_br - mask\_bg) \cdot B$$

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ColorCorrectionMatrixSelectorI	IEnumeration	4	R/W	Selects a row element of color correction matrix.
ColorCorrectionMatrixSelectorJ	IEnumeration	4	R/W	Selects a column element of color correction matrix.
ColorCorrectionMatrix	IFloat	4	R/W	Sets a coefficient of color correction matrix.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
MaskingRG	Implemented	0x21F040	[31]	R	Returns the state whether the function is implemented.
	Mult	0x21F04C	4	R	Absolute Value = Raw * (Mult / Div)
	Div	0x21F050	4	R	
	Min	0x21F054	4	R	Returns the minimum value of MaskingRG Value register.
	Max	0x21F058	4	R	Returns the maximum value of MaskingRG Value register.
	Value	0x21F05C	4	R/W	mask_rg value. (Raw value)
MaskingRB	Same structure as MaskingRG				
	Value	0x21F07C	4	R/W	mask_rb value (Raw value)
MaskingGR	Same structure as MaskingRG				
	Value	0x21F09C	4	R/W	mask_gr value. (Raw value)
MaskingGB	Same structure as MaskingRG				
	Value	0x21F0BC	4	R/W	mask_gb value (Raw value)
MaskingBR	Same structure as MaskingRG				
	Value	0x21F0DC	4	R/W	mask_br value (Raw value)
MaskingBG	Same structure as MaskingRG				
	Value	0x21F0FC	4	R/W	mask_bg value (Raw value)

● Control with TeliCamSDK

Camera feature API

Control ColorCorrectionMatrix using dedicated API.

API name	Description
GetCamColorCorrectionMatrixMinMax	Get minimum and maximum value
GetCamColorCorrectionMatrix	Get current ColorCorrectionMatrix value
SetCamColorCorrectionMatrix	Set new ColorCorrectionMatrix value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control ColorCorrectionMatrix using GenICam API.

◆ColorCorrectionMatrix

1. Select a row element of color correction matrix by ‘ColorCorrectionMatrixSelectorI’.

   Select a column element of color correction matrix by ‘ColorCorrectionMatrixSelectorJ’.

   Integer value and string value of Enumeration are as follows.

Integer	String
0	R
1	G
2	B

   The correspondence of ‘ColorCorrectionMatrixSelectorI’ and ‘ColorCorrectionMatrixSelectorJ’ to color correction matrix element is as follows.

	SelectorJ=R	SelectorJ=G	SelectorJ=B
SelectorI=R		mask_rg	mask_rb
SelectorI=G	mask_gr		mask_gb
SelectorI=B	mask_br	mask_bg	

Corresponding elements of color correction matrix

2. Set a coefficient of color correction matrix by ‘ColorCorrectionMatrix’.

```
// GenICam node handle
CAM_NODE_HANDLE    hSelectorI = NULL;
CAM_NODE_HANDLE    hSelectorJ = NULL;
CAM_NODE_HANDLE    hNode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “ColorCorrectionMatrixSelectorI”, &hSelectorI);
Nd_GetNode(s_hCam, “ColorCorrectionMatrixSelectorJ”, &hSelectorJ);
Nd_GetNode(s_hCam, “ColorCorrectionMatrix”, &hNode);

float64_t      dCoefficient[] = { 0.0,-0.2, 0.2,
                                  -0.3,0.0,-0.4,
                                  -0.1,-0.5,0.0 };    // 3x3 matrix

for(int64_t i=0; i<3; i++)
{
 for(int64_t j=0; j<3; j++)
 {
  if(i != j)
  {
   // 1.Select a color correction matrix element
   Nd_SetEnumIntValue(s_hCam, hSelectorI, i);
   Nd_SetEnumIntValue(s_hCam, hSelectorJ, j);

   // 2.Set a coefficient of color correction matrix.
   Nd_SetFloatValue(s_hCam, hNode, dCoefficient);
  }
 }
}
```

Please refer to [INode functions], [IEnumeration node functions] and [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control ColorCorrectionMatrix by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆ColorCorrectionMatrix

Write to ‘Value’ field of ‘MaskingRG’, ‘MaskingRB’, ‘MaskingGR’, ‘MaskingGB’, ‘MaskingBR’, ‘MaskingBG’ register.

```
// Masking value (Raw value)
int32_t      uiMasking[] = {-13108, 13107,-19661,-26215, -6554,-32768};

// Set Masking Value
uint64_t      addr = 0x21F05C;
for(int i=0; i<6; i++, addr+= 0x20)
{
  Cam_WriteReg(s_hCam, addr, 1, &uiMasking[i]);
}
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

BalanceRatio	Raw Value	Absolute (Float) Value
Minimum	-65536	-1.00
Maximum	65535	0.99
Formula	Absolute Value = Raw Value / 65536	

Initial Value

		SelectorJ		
		R	G	B
SelectorI	R		0.00	0.00
	G	0.00		0.00
	B	0.00	0.00	

● Note

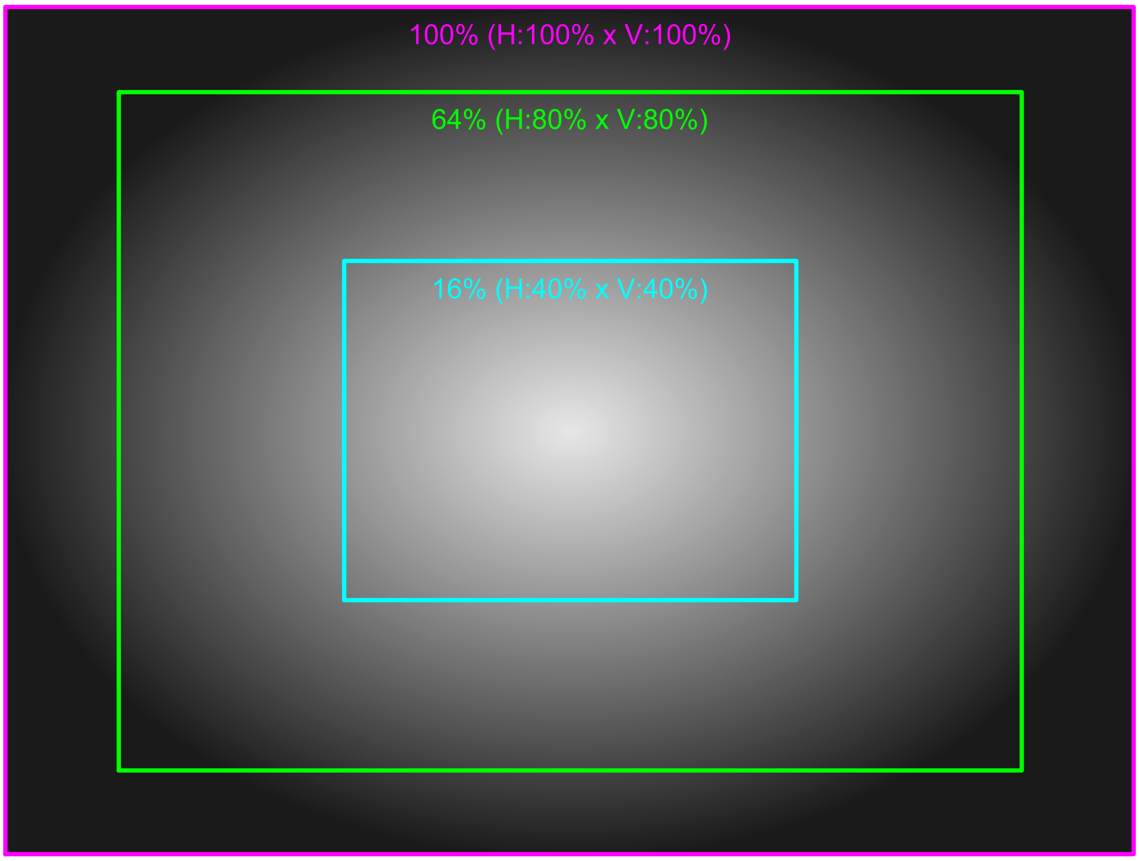
ColorCorrectionMatrix control is available in following PixelFormat.

Bayer8/10/12(BayerProcessingMode = Full), RGB8, BGR8, YUV411, YUV422

ALCControl

This section describes ALCControl feature.

－ ALCPhotometricAreaSize defines photometric area size for measuring luminance.



ALCPhotometricAreaSize (e.g. 100%, 64%, 16%)

－ ALCExposureValue defines a correction value for a convergence value.

Final convergence value is determined by the following formula.

Final convergence value = Reference Luminance x $2^{\text{ALCExposureValue}}$

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ALCPhotometricAreaSize	IFloat	4	R/W	Sets ALC photometric area size.
ALCExposureValue	IFloat	4	R/W	Sets ALC Exposure Value.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ALCPhotometricAreaSize	Implemented	0x21F360	[31]	R	Returns the state whether the function is implemented.
	Mult	0x21F36C	4	R	Absolute Value = Raw * (Mult / Div)
	Div	0x21F370	4	R	
	Min	0x21F374	4	R	Returns the minimum value of ALC photometric area size.
	Max	0x21F378	4	R	Returns the maximum value of ALC photometric area size.
	Value	0x21F37C	4	R/W	Sets ALC photometric area size. (Raw value)
ALCExposureValue	Implemented	0x2040C0	[31]	R	Returns the state whether the function is implemented.
	Mult	0x2040CC	4	R	Absolute Value = Raw * (Mult / Div)
	Div	0x2040D0	4	R	
	Min	0x2040D4	4	R	Returns the minimum value of ALC Exposure Value.
	Max	0x2040D8	4	R	Returns the maximum value of ALC Exposure Value.
	Value	0x2040DC	4	R/W	Sets ALC Exposure Value. (Raw value)

● Control with TeliCamSDK

GenICam function API

Control ALC parameters using GenICam API.

◆ALCPhotometricAreaSize/ALCExposureValue

Use ‘IFloat’ interface to control ALCPhotometricAreaSize and ALCExposureValue.

```
// GenICam node handle
CAM_NODE_HANDLE   hSize = NULL;
CAM_NODE_HANDLE   hEV = NULL;

// ALCPhotometricAreaSize = 25%
float64_t      dSize = 25.0;
// ALCExposureValue = +1.0EV
float64_t      dEV = 1.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “ALCPhotometricAreaSize”, &hSize);
Nd_GetNode(s_hCam, “ALCExposureValue”, &hEV);

// Set ALCPhotometricAreaSize
Nd_SetFloatValue(s_hCam, hSize, dSize);
// Set ALCExposureValue
Nd_SetFloatValue(s_hCam, hEV, dEV);
```

Please refer to [INode functions] and [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control ALC parameters by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆ALCPhotometricAreaSize/ALCExposureValue

Write to ‘Value’ field of ‘ALCPhotometricAreaSize’ register.

Write to ‘Value’ field of ‘ALCExposureValue’ register.

```
// ALCPhotometricAreaSize = 25% (Raw value = 25)
uint32_t      uiSizeRaw = 25;
// ALCExposureValue = +1.0EV (Raw value = 10)
int32_t      iEVRaw = 10; // signed

// Set ALCPhotometricAreaSize
Cam_WriteReg(s_hCam, 0x21F37C, 1, &uiSizeRaw);
// Set ALCExposureValue
Cam_WriteReg(s_hCam, 0x2040DC, 1, &iEVRaw);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

ALCPhotometricAreaSize	Raw Value	Absolute (Float) Value
Minimum	1	1.00[%]
Maximum	100	100.00[%]
Initial Value	100	100.00[%]
Formula	Absolute Value [%] = Raw Value	

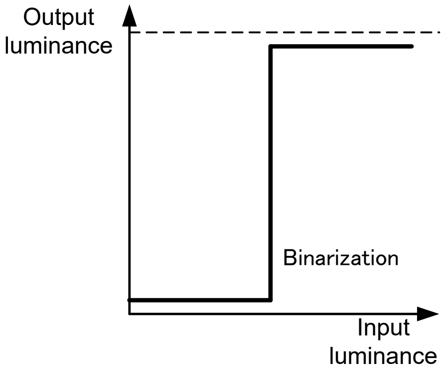
ALCExposureValue	Raw Value	Absolute (Float) Value
Minimum	-20	-2.0[EV]
Maximum	15	+1.5[EV]
Initial Value	100	0.0[EV]
Formula	Absolute Value [EV] = Raw Value / 10	

● Note

*The camera operation is not supported when the Random Trigger Shutter mode.

LUTControl

This function allows you to apply the arbitrary LUT (input: 12it, output: 12bit) to the output images.



example of LUT setting

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
LUTEnable	IBoolean	4	R/W	Sets the activation of LUT function.
LUTIndex	IInteger	4	R/W	Sets the input level of LUT.
LUTValue	IInteger	4	R/W	Sets the output level of LUT.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
LUTEnable	Implemented	0x206020	[31]	R	Returns the state whether the function is implemented.
	Value	0x20603C	4	R/W	Sets the activation of LUT function. [0] : Off [1] : On
LUTValueAll	Implemented	0x2FFFDC	[31]	R	Returns the state whether the function is implemented.
	Value[0]	0x300000	4	R/W	Sets the output level of LUT.
	Value[1]	0x300004	4	R/W	Sets the output level of LUT.
	Value[2]	0x300008	4	R/W	Sets the output level of LUT.
	…	…	…	…	…
	Value[4095]	0x303FFC	4	R/W	Sets the output level of LUT.

● Control with TeliCamSDK

Camera feature API

Control LUT using dedicated API.

API name	Description
GetCamLUTEnable	Get activation status
SetCamLUTEnable	Set activation status
GetCamLUTValue	Get LUT value
SetCamLUTValue	Set LUT value

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control LUT using GenICam API.

◆LUT

1. Set the input level of LUT to ‘LUTIndex’.
2. Set the output level of LUT to ‘LUTValue’
3. Set the activation of LUT function by ‘LUTEnable’

```
// GenICam node handle
CAM_NODE_HANDLE    hIndex = NULL;
CAM_NODE_HANDLE    hValue = NULL;
CAM_NODE_HANDLE    hEnable = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “LUTIndex”, &hIndex);
Nd_GetNode(s_hCam, “LUTValue”, &hValue);
Nd_GetNode(s_hCam, “LUTEnable”, &hEnable);

for(int64_t i=0; i<4096; i++)
{
    // 1.Set the input level of LUT to 'LUTIndex'.
    Nd_SetIntValue(s_hCam, hIndex, i);
    // 2.Set the output level of LUT to 'LUTValue'
    Nd_SetIntValue(s_hCam, hValue, 4095 - i); // invert
}

// 3.Set the activation of LUT function by 'LUTEnable'
Nd_SetBoolValue(s_hCam, hEnable, true);
```

Please refer to [INode functions], [IInteger node functions] and [IBoolean node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control LUT by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆LUT

1. Set the output level of LUT.

   Write to ‘Value[0]’ to ‘Value[4095]’ of ‘LUTValueAll’ register to set the output level of LUT.

   The index of ‘Value’ register means the input level of LUT.

2. Set the activation of LUT function.

   Write to ‘Value’ field of ‘LUTEnable’ register to activate LUT function.

```
// 1.Set the output level of LUT.
uint64_t    addr = 0x300000;
uint32_t    dat;
for(int i=0; i<4096; i++, addr+= 0x4)
{
        dat = 4095 - i; // invert
        Cam_WriteReg(s_hCam, addr, 1, &dat);
}

// 2.Set the activation of LUT function.
dat = 1;
Cam_WriteReg(s_hCam, 0x20603C, 1, &dat);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

LUTIndex/LUTValue	Value
Minimum	0
Maximum	4095

UserSetControl

You are able to save a user setting to the non-volatile or volatile memory of the camera. There are 15 user memory channels for user setting.

By using user memory, you are able to restore frequent used settings at the time of next start-up.

The following table is the list of registers applied to “UserSetLoad”/“UserSetSave”,“UserSetQuickSave”.

Table: List of registers to be applied to UserSet

Category	Register
ImageFormatControl	ImageFormatSelector
	Width
	Height
	OffsetX
	OffsetY
	Binning
	Decimation
	Reverse
	PixelFormat
	TestPattern
	BayerProcessingMode
AcquisitionControl	AcquisitionFrameRateEnable
	AcquisitionFrameRate
	AcquisitionFrameIntervalControl
	AcquisitionFrameInterval
	HighFramerateMode
TriggerControl	TriggerMode
	TriggerSequence
	TriggerSource
	TriggerAdditionalParameter
	TriggerDelay
ExposureControl	ExposureTime
	ExposureControl
	ShortExposureMode
DigitalIOControl	LineModeAll
	LineInverterAll
	UserOutputValueAll
DigitalIOControl	LineSelector
	LineSource
	AntiGlitch
	AntiChattering
TimerControl	TimerTriggerSource
	TimerDuration
	TimerDelay
AnalogControl	Gain
	GainControl
	BlackLevel
	Gamma
	Sharpness
	BalanceRatio
	ColorCorrectionMatrix
ALCControl	ALCExposureValue
	ALCPhotometricAreaSize
LUTControl	LUTEnable
EventControl	EventNotification
DPCControl	DPCEnable(*)
	DPCNumber(*)
	DPCEntryX(*)
	DPCEntryY(*)
SequentialShutterControl	SequentialShutterEnable(*)
	SequentialShutterTerminateAt(*)
	SequentialShutterEntry(*)

(*) DPC and SequentialShutter entries are stored to a single channel. Entries are shared with all channels.

This table lists all features of B/W and Color models. Please refer to [Table : supported features list] as well.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
UserSetSelector	IEnumeration	4	R/W	Selects a channel of user setting.
UserSetLoad	ICommand	4	W	Loads a user setting.
UserSetSave	ICommand	4	W	Saves a user setting to non-volatile memory.
UserSetQuickSave	ICommand	4	W	Saves a user setting to volatile memory.
UserSetDefault	IEnumeration	4	R/W	Selects a channel of user setting when camera powers up.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
UserSetSelector	Implemented	0x208060	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20806C	4	R	[0] : Default [1] : UserSet1 … [15] : UserSet15
	Value	0x20807C	4	R/W	Selects the UserSet.
UserSetCommand	Implemented	0x208080	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x20808C	4	R	[0] : Done [8] : Load [9] : Save [120] : QuickSave
	Value	0x20809C	4	R/W	Selects the UserSet command.
UserSetDefault	Implemented	0x2080A0	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x2080AC	4	R	[0] : Default [1] : UserSet1 … [15] : UserSet15
	Value	0x2080BC	4	R/W	Selects the number of UserSet when the camera starts up.

● **Control with TeliCamSDK**

Camera feature API

Control UserSetControl using dedicated API.

API name	Description
ExecuteCamUserSetLoad	Execute UserSetLoad
ExecuteCamUserSetSave	Execute UserSetSave
ExecuteCamUserSetSaveAndSetDefault	Execute UserSetDefault

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control UserSetControl using GenICam API.

◆**UserSetLoad/UserSetSave/UserSetQuickSave**

1. Select a channel of user setting by ‘UserSetSelector’.

   Integer value and string value of Enumeration are as follows.

Integer	String	Description	Save	Load
0	Default	Initial factory setting.	-	✓
1~15	UserSet1~15	Memory channel 1 to 15 for user setting.	✓	✓

2. Execute UserSetLoad, UserSetSave or UserSetQuickSave

   When execute ‘UserSetLoad’, the camera loads user setting from the channel selected in ‘UserSetSelector’ register and applies them.

   When execute ‘UserSetSave’ or ‘UserSetQuickSave’, the camera saves user setting to the channel selected in ‘UserSetSelector’ register.

   After execution of 'UserSetSave', wait until 'UserSetSave' completion by Nd_GetCmdIsDone.

```
// GenICam node handle
CAM_NODE_HANDLE    hSelector = NULL;
CAM_NODE_HANDLE    hSave = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “UserSetSelector”, &hSelector);
Nd_GetNode(s_hCam, “UserSetSave”, &hSave);

// 1.Select a channel of user setting by ‘UserSetSelector’.
uint64_t dat = 1; // UserSet1
Nd_SetEnumIntValue(s_hCam, hSelector, dat);


// 2. Execute UserSetSave
Nd_CmdExecute(s_hCam, hSave);

bool8_t bDone;
while(1) {
  Nd_GetCmdIsDone(s_hCam, hSave, &bDone);
  if(bDone == true)
    break;
  Sleep(0);
}
```

◆UserSetDefault

Select a channel of user setting when camera powers up by ‘UserSetDefault’.

Integer value and string value of Enumeration are as follows.

Integer	String	Description
0	Default	Initial factory setting.
1~15	UserSet1~15	Memory channel 1 to 15 for user setting.

```
// GenICam node handle
CAM_NODE_HANDLE    hSelector = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “UserSetDefault”, &hSelector);

// 1.Select a channel of user setting when camera powers up by ‘UserSetDefault’.
uint64_t dat = 1; // UserSet1
Nd_SetEnumIntValue(s_hCam, hSelector, dat);
```

Please refer to [INode functions], [IEnumeration node functions] and [ICommand node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control UserSetControl by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆UserSetLoad/UserSetSave/UserSetQuickSave

1. Select a channel of user setting by ‘UserSetSelector’.

   Write to ‘Value’ field of ‘UserSetSelector’ register.

2. Execute UserSetLoad, UserSetSave or UserSetQuickSave

   Write [8] to ‘Value’ field of ‘UserSetCommand’ register to execute UserSetLoad.

   Write [9] to ‘Value’ field of ‘UserSetCommand’ register to execute UserSetSave.

   Write [120] to ‘Value’ field of ‘UserSetCommand’ register to execute UserSetQuickSave.

   After execution of 'UserSetSave', wait until 'UserSetCommand’ is ‘Done’ (= 0).

```
uint32_t dat;

// 1.Select a channel of user setting by ‘UserSetSelector’.
dat = 1; // UserSet1
Cam_WriteReg(s_hCam, 0x20807C, 1, &dat);

// 2. Execute UserSetSave
dat = 9; // UserSetSave
Cam_WriteReg(s_hCam, 0x20809C, 1, &dat);

while(1) {
 Cam_ReadReg(s_hCam, 0x20809C, 1, &dat);
 if(dat == 0)
  break;
 Sleep(0);
}
```

◆UserSetDefault

1. Select a channel of user setting when camera powers up by ‘UserSetDefault’.

```
// 1.Select a channel of user setting when camera powers up
uint32_t dat = 1; // UserSet1
Cam_WriteReg(s_hCam, 0x2080BC, 1, &dat);
```

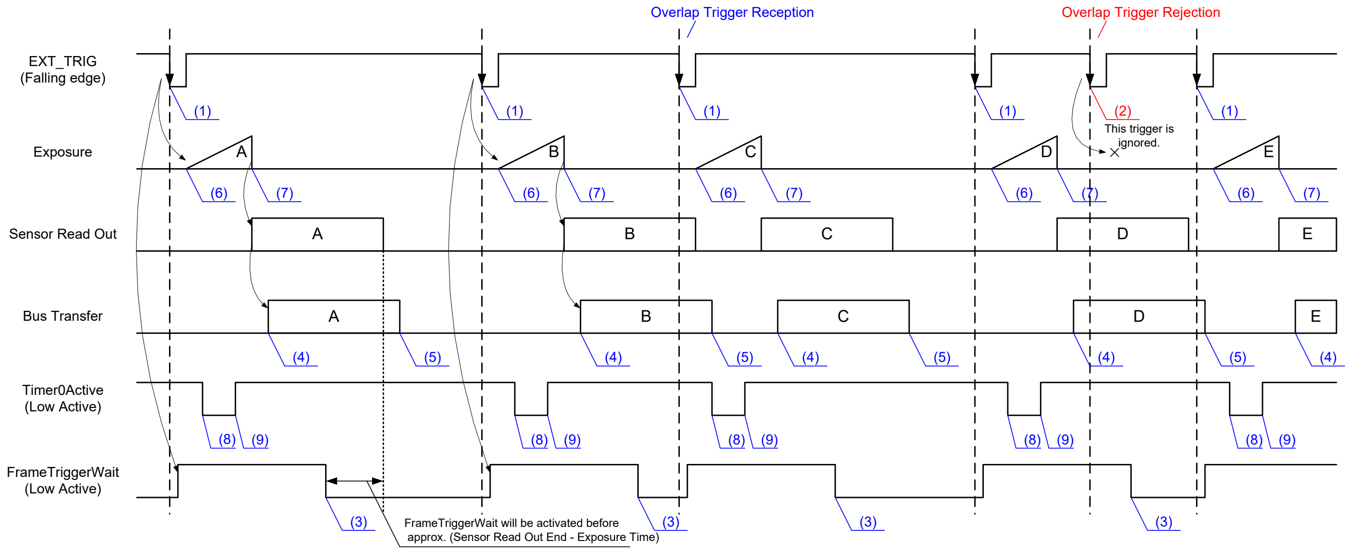
Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

EventControl

Camera notifies FrameTrigger status and other information by USB3 Vision Event Packet.

- FrameTrigger : Reception of Frame Start Trigger
- FrameTriggerError : Rejection of Frame Start Trigger
- FrameTriggerWait : Start of waiting for Frame Start Trigger
- FrameTransferStart : Start of transferring streaming data
- FrameTransferEnd : End of transferring streaming data
- ExposureStart : Start of Exposure
- ExposureEnd : End of Exposure
- Timer0Start : Start of Timer0
- Timer0End : End of Timer0

Events timing are as following chart.



Event Name

(1) FrameTrigger : Reception of Frame Start Trigger.
(2) FrameTriggerError : Rejection of Frame Start Trigger.
(3) FrameTriggerWait : Start of waiting for Frame Start Trigger.
(4) FrameTransferStart : Start of transferring streaming data.
(5) FrameTransferEnd : End of transferring streaming data.
(6) ExposureStart : Start of Exposure.
(7) ExposureEnd : End of Exposure.
(8) Timer0Start : Start of Timer0.
(9) Timer0End : End of Timer0.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
EventSelector	IEnumeration	4	R/W	Selects which Event to signal to the host application.
EventNotification	IEnumeration	4	R/W	Sets the activation of event notification

Event Name	Event ID	Event Data	Length Byte / [bit]	Description
EventFrameTrigger	0x8020	EventFrameTriggerTimestamp	8	Returns the timestamp at the time of Event.
EventFrameTriggeError	0x8021	EventFrameTriggerErrorTimestamp	8	Returns the timestamp at the time of Event.
EventFrameTriggeWait	0x8022	EventFrameTriggerWaitTimestamp	8	Returns the timestamp at the time of Event.
EventFrameTransferStart	0x8030	EventFrameTransferStartTimestamp	8	Returns the timestamp at the time of Event.
EventFrameTransferEnd	0x8031	EventFrameTransferEndTimestamp	8	Returns the timestamp at the time of Event.
EventExposureStart	0x8040	EventExposureStartTimestamp	8	Returns the timestamp at the time of Event.
EventExposureEnd	0x8041	EventExposureEndTimestamp	8	Returns the timestamp at the time of Event.
EventTimer0Start	0x9000	EventTimer0StartTimestamp	8	Returns the timestamp at the time of Event.
EventTimer0End	0x9001	EventTimer0EndTimestamp	8	Returns the timestamp at the time of Event.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
EventNotificationOfFrame	Implemented	0x21F220	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F22C	4	R	[0] : FrameTrigger [1] : FrameTriggerError [2] : FrameTriggerWait [3] : FrameStart [4] : FrameEnd [16] : FrameTransferStart [17] : FrameTransferEnd
	Value	0x21F230	4	R/W	Sets the activation of event notification of frame.
EventNotificationOfExposure	Implemented	0x21F240	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F24C	4	R	[0] : ExposureStart [1] : ExposureEnd
	Value	0x21F250	4	R/W	Selects the UserSet command.
EventNotificationOfTimerStart	Implemented	0x21F380	[31]	R	Sets the activation of event notification of exposure.
	ListOfElements	0x21F38C	4	R	[0] : Timer0Start
	Value	0x21F390	4	R/W	Sets the activation of event notification of timer start.
EventNotificationOfTimerEnd	Implemented	0x21F3A0	[31]	R	Sets the activation of event notification of exposure.
	ListOfElements	0x21F3AC	4	R	[0] : Timer0End
	Value	0x21F3B0	4	R/W	Sets the activation of event notification of timer end.

● Control with TeliCamSDK

Camera feature API

Control Event features using dedicated API.

API name		Description
High-level API functions	Evt_OpenSimple	Opens event interface
	Evt_Activate	Activates specified camera event.
	Evt_Deactivate	Deactivates specified camera event.
Low-level API functions	Evt_Open	Opens event interface
	Evt_CreateRequest	Creates EventRequest structure
	Evt_ReleaseRequest	Releases EventRequest structure
	Evt_EnqueueRequest	Puts specified EventRequest into the EventWaitQueue
	Evt_DequeueRequest	Retrieves an EventRequest structure from the EventCompleteQueue
	Evt_FlushWaitQueue	Stops all CameraEvent receiving operations, and moves all EventRequest structures in the EventWaitQueue to the EventCompleteQueue
Common functions	Evt_Close	Closes event interface

Please refer to [Camera event notification functions] in [TeliCamAPI Library manual] and [GrabEvent] sample code in your TeliCamSDK install folder.

GenICam function API

Please use dedicated API to control Event features.

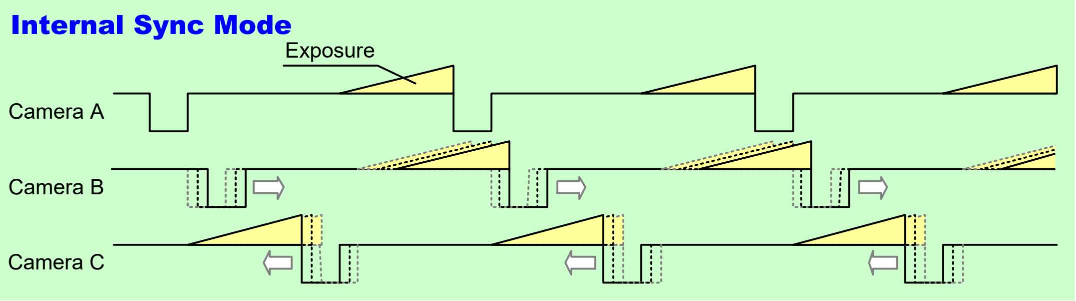
Register access API

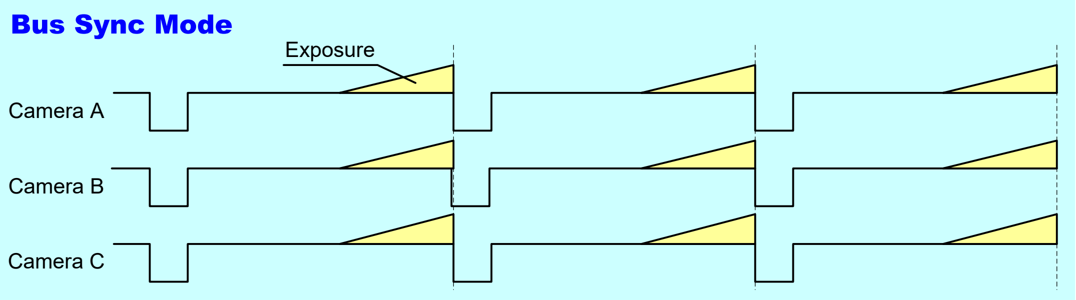
Please use dedicated API to control Event features.

FrameSynchronization

Selects the camera frame synchronization method.

1. Off : Free run operation (internal sync mode)
2. Bus : Synchronized with USB bus clock (Bus Sync mode)





● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
FrameSynchronization	IEnumeration	4	R/W	Selects the camera frame synchronization method.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
FrameSynchronization	Implemented	0x21F020	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F02C	16	R	[0] : Off [1] : Bus
	Value	0x21F03C	4	R/W	Selects the camera frame synchronization method.

● Control with TeliCamSDK

GenICam function API

Control FrameSynchronization using GenICam API.

◆FrameSynchronization

Select a frame synchronization method.

Integer value and string value of Enumeration are as follows.

Integer	String	Description
0	Off	Internal synchronization
1	Bus	Bus synchronization

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, “FrameSynchronization”, &hNode);

// Select a frame synchronization method
Nd_SetEnumStrValue(s_hCam, hNode, "Bus");
```

Please refer to [INode functions], [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control FrameSynchronization by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆FrameSynchronization

Write to ‘Value’ field of ‘FrameSynchronization register.

```
// Select a frame synchronization method
uint32_t dat = 1;
Cam_WriteReg(s_hCam, 0x21F03C, 1, &dat); // Bus Sync mode
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

LEDIndicatorLuminance

Adjusts LED indicator luminance.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
LEDIndicatorLuminance	IFloat	4	R/W	Sets the LED indicator luminance.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
LEDIndicatorLuminance	Implemented	0x21F100	[31]	R	Returns the state whether the function is implemented.
	Mult	0x21F10C	4	R	Absolute Value [%] = Raw * (Mult / Div)
	Div	0x21F110	4	R	
	Min	0x21F114	4	R	Returns the minimum value of LED indicator luminance register.
	Max	0x21F118	4	R	Returns the maximum value of LED indicator luminance register.
	Value	0x21F11C	4	R/W	Sets the LED indicator luminance

● Control with TeliCamSDK

GenICam function API

Control LED indicator luminance using GenICam API.

◆LEDIndicatorLuminance

Use ‘IFloat’ interface to control LED indicator luminance.

```
// GenICam node handle
CAM_NODE_HANDLE    hNode = NULL;

// luminance = 50[%]
// actual value (4/7)*100 = 57.14[%]
float64_t       dLuminance = 50.0;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "LEDIndicatorLuminance", &hNode);

// Set LED indicator luminance Value
Nd_SetFloatValue(s_hCam, hNode, dLuminance);
```

Please refer to [INode functions], [IFloat node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control LED indicator luminance by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆LEDIndicatorLuminance

Write to ‘Value’ field of ‘LEDIndicatorLuminance’ register.

```
// luminance = 50[%] (Raw value = 4)
// actual value (4/7)*100 = 57.14[%]
uint32_t       uiLuminanceRaw = 4;

// Set LED indicator luminance Value
Cam_WriteReg(s_hCam, 0x21F11C, 1, &uiLuminanceRaw);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

LEDIndicatorLuminance	Raw Value	Absolute (Float) Value
Minimum	0	0.00 [%]
Maximum	7	100.00 [%]
Initial Value	7	100.00 [%]
Formula	Absolute Value [%] = Raw Value / 7	

DPCControl

This DPC (Defective Pixel Correction) function corrects defective pixels from the image sensor. Specifying X and Y coordinates of the defective pixels, the defective pixels are corrected by calculation from the neighboring pixels.

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
DPCEnable	IEnumeration	4	R/W	Sets the activation of DPC function.
DPCNumber	IInteger	4	R/W	Sets the number of pixels to correct.
DPCIndex	IInteger	4	R/W	Sets the index number for entry.
DPCEntryX	IInteger	4	R/W	Sets the X coordinate of defective pixel.
DPCEntryY	IInteger	4	R/W	Sets the Y coordinate of defective pixel.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
DPCEnable	Implemented	0x21F280	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F28C	4	R	[0] : Off [1] : On
	Value	0x21F29C	4	R/W	Sets the activation of DPC function.
DPCNumber	Implemented	0x21F2A0	[31]	R	Returns the state whether the function is implemented.
	Value	0x21F2BC	4	R/W	Sets the number of Pixels to correct.
DPCCoord	Value[0][0]	0x400000	4	R/W	Sets the first X coordinate of defective pixel.
	Value[0][1]	0x400004	4	R/W	Sets the first Y coordinate of defective pixel.
	Value[1][0]	0x400008	4	R/W	Sets the second X coordinate of defective pixel.
	Value[1][1]	0x40000C	4	R/W	Sets the second Y coordinate of defective pixel.
	…	…	…	…	…
	Value[255][0]	0x4007F8	4	R/W	Sets the 255th X coordinate of defective pixel.
	Value[255][1]	0x4007FC	4	R/W	Sets the 255th Y coordinate of defective pixel.

● Control with TeliCamSDK

GenICam function API

Control DPC using GenICam API.

◆DPC

1. Set the coordinates of defective pixels by ‘DPCIndex’, ‘DPCEntryX’ and ‘DPCEntryY’.
2. Sets the number of pixels to correct to ‘DPCNumber’.
3. Sets the activation of DPC function by ‘DPCEnable’.

Integer value and string value of Enumeration are as follows.

Integer	String
0	Off
1	On

```
// GenICam node handle
CAM_NODE_HANDLE   hIndex = NULL;
CAM_NODE_HANDLE   hEntryX = NULL;
CAM_NODE_HANDLE   hEntryY = NULL;
CAM_NODE_HANDLE   hNumber = NULL;
CAM_NODE_HANDLE   hEnable = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "DPCIndex", &hIndex);
Nd_GetNode(s_hCam, "DPCEntryX", &hEntryX);
Nd_GetNode(s_hCam, "DPCEntryY", &hEntryY);
Nd_GetNode(s_hCam, "DPCNumber", &hNumber);
Nd_GetNode(s_hCam, "DPCEnable", &hEnable);

int64_t entry[2][2] = {{100,200},{150,300}};// {x.y}
int64_t number;

for(number=0; number<2; number++)
{
  // 1.Set the coordinates of defective pixels by 'DPCIndex', 'DPCEntryX' and
'DPCEntryY'.
  Nd_SetIntValue(s_hCam, hIndex, number); // 0 origin
  Nd_SetIntValue(s_hCam, hEntryX, entry[number][0]);
  Nd_SetIntValue(s_hCam, hEntryY, entry[number][1]);
}

// 2.Sets the number of pixels to correct to 'DPCNumber'.
Nd_SetIntValue(s_hCam, hNumber, number); // 2 pixels are to be corrected.

// 3.Sets the activation of DPC function by 'DPCEnable'.
Nd_SetEnumStrValue(s_hCam, hEnable, "On");
```

Please refer to [INode functions], [IInteger node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control DPC by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆DPC

1. Set the coordinates of defective pixels.

   Write to 'Value[index][0]' and 'Value[index][1]' field of 'DPCCoord' register.

   The first index of 'Value' register indicates defective pixel number.

   The second index of [0] and [1] correspond to X and Y coordinates.

2. Sets the number of pixels to correct.

   Write to 'Value' field of 'DPCNumber' register.

3. Sets the activation of DPC function.

   Write [1] to 'Value' field of 'DPCEnable' register.

```
// 1.Set the coordinates of defective pixels.
uint32_t    entry[2][2] = {{100,200},{150,300}};    // {x.y}
uint32_t    number;
uint64_t    addr = 0x400000;
for(number=0; number<2; number++, addr+= 0x8)
{
        Cam_WriteReg(s_hCam, addr       , 1, &entry[number][0]);
        Cam_WriteReg(s_hCam, addr+0x4, 1, &entry[number][1]);
}

// 2.Sets the number of pixels to correct.
Cam_WriteReg(s_hCam, 0x21F2BC, 1, &number);

// 3.Set the activation of DPC function.
uint32_t dat = 1;
Cam_WriteReg(s_hCam, 0x21F29C, 1, &dat);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

	DPCNumber	DPCIndex
Minimum	0	0
Maximum	256	255

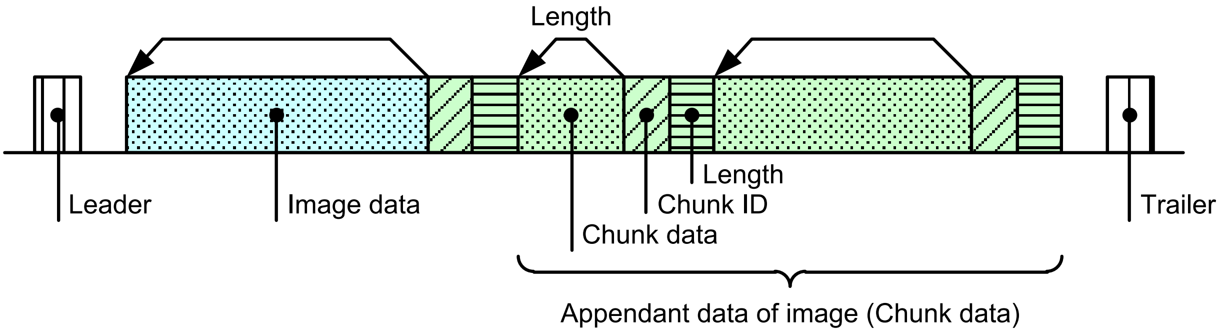
	DPCEntryX	DPCEntryY
Minimum	0	0
Maximum	WidthMax-1	HeightMax-1

Chunk

This section describes Chunk feature. Chunk data means tagged blocks of data.

The tags allow a chunk parser to dissect the data payload into its elements and to identify the content.

The length of a frame varies depending on the number of activated chunks.



Chunk

Length (B/W Model)		Length (Color Model)
	🟧 Image data	
+0x000	🟧 ChunkID = 0x00000001	+0x000
+0x004	🟧 Length = (Image data size)	+0x004
+0x008	🟧 BlockID	+0x008
+0x010	🟧 ChunkID = 0x00000100	+0x010
+0x014	🟧 Length = 0x08	+0x014
+0x018	🟩 FrameBurstTriggerCount	+0x018
+0x01C	🟩 ChunkID = 0x80001001	+0x01C
+0x020	🟩 Length = 0x04	+0x020
+0x024	🟧 ExposureTime	+0x024
+0x028	🟧 ChunkID = 0x4004003C	+0x028
+0x02C	🟧 Length = 0x04	+0x02C
+0x030	🟧 Gain	+0x030
+0x034	🟧 ChunkID = 0x4004007C	+0x034
+0x038	🟧 Length = 0x04	+0x038
	🟪 WhiteBalanceR	+0x03C
	🟪 ChunkID = 0x4005007C	+0x040
	🟪 Length = 0x04	+0x044
	🟪 WhiteBalanceB	+0x048
	🟪 ChunkID = 0x4005009C	+0x04C
	🟪 Length = 0x04	+0x050
+0x03C	🟧 LineStatusAll	+0x054
+0x040	🟧 ChunkID = 0x4009007C	+0x058
+0x044	🟧 Length = 0x04	+0x05C
+0x048	🟩 SequentialShutterNumber	+0x060
+0x04C	🟩 ChunkID = 0x80002000	+0x064
+0x050	🟩 Length = 0x04	+0x068
+0x054	🟩 SequentialShutterElement	+0x06C
+0x058	🟩 ChunkID = 0x80002001	+0x070
+0x05C	🟩 Length = 0x04	+0x074
+0x060	🟦 UserArea (256 Bytes)	+0x078
+0x160	🟦 ChunkID = 0x80000000	+0x178
+0x164	🟦 Length = 0x100	+0x17C

🟧 Always output

🟪 Always output (Color Model)

🟩 Depends on camera mode

🟦 Depends on register setting

Chunk Data Structure

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
ChunkModeActive	IBoolean	4	R/W	Sets the activation of Chunk function.
ChunkSelector	IEnumeration	4	R/W	Select which Chunk to enable or control.
ChunkEnable	IBoolean	4	R/W	Enables the inclusion of the selected Chunk data in the payload of the image.
ChunkUserAreaLength	IInteger	4	R	Indicates the length of ChunkUserAreaTable.
ChunkUserAreaTable	IString	256	R/W	Set the free user space.(Maximum:256byte)
ChunkFrameID	IInteger	8	R	Returns the Block ID of the frame included in the Chunk data.
ChunkExposureTime	IFloat	4	R	Returns the ExposureTime of the frame included in the Chunk data.
ChunkGain	IFloat	4	R	Returns the Gain of the frame included in the Chunk data.
ChunkWhiteBalanceR	IFloat	4	R	Returns the WhiteBalanceR of the frame included in the Chunk data.
ChunkWhiteBalanceB	IFloat	4	R	Returns the WhiteBalanceB of the frame included in the Chunk data.
ChunkLineStatusAll	IInteger	4	R	Returns the LineStatusAll of the frame included in the Chunk data.
ChunkFrameBurstTriggerCount	IInteger	4	R	Returns the FrameBurstTriggerCount of the frame included in the Chunk data.
ChunkSequentialShutterNumber	IInteger	4	R	Returns the SequentialShutterNumbere of the frame included in the Chunk data.
ChunkSequentialShutterElement	IInteger	4	R	Returns the SequentialShutterElement of the frame included in the Chunk data.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
ChunkModeActive	Implemented	0x21D020	[31]	R	Returns the state whether the function is implemented.
	BitWritable	0x21D02C	4	R	[0]:Off [1]:On
	Value	0x21D030	4	R/W	Sets the activation of chunk feature.
ChunkEnableOfVendorSpecific	Implemented	0x21D040	[31]	R	Returns the state whether the function is implemented.
	BitWritable	0x21D04C	4	R	[0] : BlockID [8] : FrameBurstTriggerCount [9] : SequentialShutterNumber [10] : SequentialShutterElement [24] : UserArea
	Value	0x21D050	4	R/W	Sets the activation of chunk data.
ChunkEnableOfCat4	Implemented	0x21D0E0	[31]	R	Returns the state whether the function is implemented.
	BitWritable	0x21D0EC	4	R	[0] : ExposureTime [2] : Gain
	Value	0x21D0F0	4	R/W	Sets the activation of chunk data.
ChunkEnableOfCat5	Implemented	0x21D100	[31]	R	Returns the state whether the function is implemented.
	BitWritable	0x21D10C	4	R	[2] : WhiteBalaceR [3] : WhiteBalaceB
	Value	0x21D110	4	R/W	Sets the activation of chunk data.
ChunkEnableOfCat9	Implemented	0x21D180	[31]	R	Returns the state whether the function is implemented.
	BitWritable	0x21D18C	4	R	[2] : LineStatusAll
	Value	0x21D190	4	R/W	Sets the activation of chunk data.
ChunkUserArea	Implemented	0x21D7F0	[31]	R	Returns the state whether the function is implemented.
	NumberOfElements	0x21D7FC	[30:0]	R	Returns the length (in bytes) of ChunkUserArea Value.
	Value[0]	0x21D800	length	R/W	Sets the ASCII string of ChunkUserArea.
	…	…	…		
	Value[last]	0x21D800 +NumberOf Elements -4		R/W	Sets the ASCII string of ChunkUserArea.

● Control with TeliCamSDK

Camera feature API

Control ExposureTime using dedicated API.

API name	Description
GetCamChunkModeActive	Get current statusof chunk data output mode
SetCamChunkModeActive	Set chunk data output mode

Please refer to [Controlling camera feature functions] in [TeliCamAPI Library manual] for more detail.

GenICam function API

Control Chunk using GenICam API.

◆Chunk

1. Close image stream.
2. Sets the activation of Chunk feature by ‘ChunkModeActive’.
3. Sets the activation of Chunk data by ‘ChunkSelector’ and ‘ChunkEnable’.

   Most of the Chunk data has been fixed to output.
4. Sets the user string to ‘ChunkUserAreaTable’. (optional)
5. Open image stream.
6. Capture image stream.

   TeliCamAPI provides streaming functions including command handling and buffer handling to capture image stream simply.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual]
   and [GrabStreamSimple] sample code in your TeliCamSDK install folder.
7. Extract Chunk data using GenICam API.

   7-1. Allocate memory by ‘Chunk_AttachedBuffer’ .

   7-2. Read Chunk data.

```
// GenICam node handle
CAM_NODE_HANDLE    hMode = NULL;
CAM_NODE_HANDLE    hSelector = NULL;
CAM_NODE_HANDLE    hEnable = NULL;
CAM_NODE_HANDLE    hFrameID = NULL;
CAM_NODE_HANDLE    hExposureTime = NULL;
CAM_NODE_HANDLE    hUserAreaTable = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "ChunkModeActive", &hMode);
Nd_GetNode(s_hCam, "ChunkSelector", &hSelector);
Nd_GetNode(s_hCam, "ChunkEnable", &hEnable);
Nd_GetNode(s_hCam, "ChunkFrameID", & hFrameID);
Nd_GetNode(s_hCam, "ChunkExposureTime", &hExposureTime);
Nd_GetNode(s_hCam, "ChunkUserAreaTable", &hUserAreaTable);

// 2.Activate ChunkModeActive
Nd_SetBoolValue(s_hCam, hMode, true);

//3.Enable ChunkSelector
Nd_SetEnumStrValue(s_hCam, hSelector, "ExposureTime");
Nd_SetBoolValue(s_hCam, hEnable, true);
Nd_SetEnumStrValue(s_hCam, hSelector, " UserArea");
Nd_SetBoolValue(s_hCam, hEnable, true);


// 4.Set the user string
Nd_SetStrValue(s_hCam, hUserAreaTable, "Test");

// 5.6.Open and capture image
Strm_ReadCurrentImage(hStrm, pvPayloadBuf, &uiPyldSize, &sImageInfo);

// 7-1. Attach Buffer
Chunk_AttachBuffer(s_hStrm, pvPayloadBuf, PyldSize)

// 7-2.Get FrameID of Chunk data.
int64_t fid= 0;
Nd_GetIntValue(s_hCam, hFID, &fid);

// 7-2.Get Exposure Time of Chunk data.
float64_t    exptime = 0;
Nd_GetFloatValue(s_hCam, hExposureTime, &exptime);

// 7-2.Get User Area data of Chunk data.
char userarea[256];
Uint32_t uiSize = 256;
Nd_GetStrValue(s_hCam, hUserAreaTable, &userarea, &uiSize);
```

Please refer to [INode functions], [IInteger node functions], [IBoolean node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control Chunk by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

◆Chunk

1. Close image stream.
2. Sets the activation of Chunk feature.

   Write to ‘Value’ field of ‘ChunkModeActive’ register.
3. Sets the activation of Chunk data.

   Write to ‘Value’ field of ‘ChunkEnableOfVendorSpecific‘, ‘ChunkEnableOfCat4’, ‘ChunkEnableOfCat5’ and ‘ChunkEnableOfCat9’ register.
4. Sets the user string. (optional)

   Write to ‘Value[0]’ to ‘Value[last]’ field of ‘ChunkUserArea‘ register.
5. Open image stream.
6. Capture image stream.

   TeliCamAPI provides streaming functions including command handling and buffer handling to capture image stream simply.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.

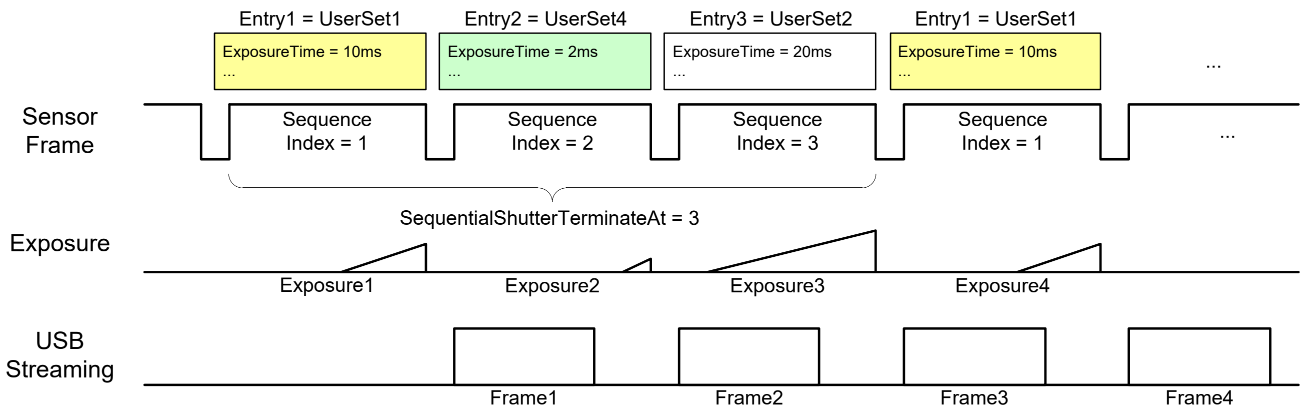
7. Extract Chunk data using GenICam API.

```
// 2.Activate ChunkModeActive
int32_t active = 1;
Cam_WriteReg(s_hCam, 0x21D030,1, &active);

// 3.Enable Chunk
int32_t cat4 = 5;
Cam_WriteReg(s_hCam, 0x21D0F0,1, &cat4);

// 5.6.Open and capture image
Strm_ReadCurrentImage(hStrm, pvPayloadBuf, &uiPyldSize, &sImageInfo);

// 7-1. Attach Buffer
Chunk_AttachBuffer(s_hStrm, pvPayloadBuf, PyldSize)

// 7-2.Get FrameID of Chunk data.
int64_t fid= 0;
Nd_GetIntValue(s_hCam, hFID, &fid);

// 7-2.Get Exposure Time of Chunk data.
float64_t    exptime = 0;
Nd_GetFloatValue(s_hCam, hExposureTime, &exptime);

// 7-2.Get User Area data of Chunk data.
char userarea[256];
uint32_t uiSize = 256;
Nd_GetStrValue(s_hCam, hUserAreaTable, &userarea, &uiSize);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

SequentialShutterControl

Sequential Shutter function performs sequential capturing with applying the settings of UserSet that have been made entry in advance. It is not supported when the ShortExposureMode is enable (ON).



Sequential Shutter

● GenICam Node

Name	Interface	Length Byte / [bit]	Access	Description
SequentialShutterEnable	IEnumeration	4	R/W	Sets the activation of Sequential Shutter function.
SequentialShutterTerminateAt	IInteger	4	R/W	Sets the number of Index to repeat the sequence.
SequentialShutterIndex	IInteger	4	R/W	Sets the sequence number to register.
SequentialShutterEntry	IInteger	4	R/W	Sets the UserSet number to register to the sequence.

● IIDC2 Register

Register	Field	Address	Length Byte / [bit]	Access	Description
SequentialShutterEnable	Implemented	0x21F300	[31]	R	Returns the state whether the function is implemented.
	ListOfElements	0x21F30C	4	R	[0] : Off [1] : On
	Value	0x21F31C	4	R/W	Sets the activation of Sequential Shutter function.
SequentialShutterTerminateAt	Implemented	0x21F320	[31]	R	Returns the state whether the function is implemented.
	Value	0x21F33C	4	R/W	Sets the number of Tables to repeat the sequence.
SequentialShutterSequenceTable	Value[0]	0x500040	4	R/W	Sets the UserSet number to register to the first sequence.
	Value[1]	0x500044	4	R/W	Sets the UserSet number to register to the second sequence.
	…	…	…	…	…
	Value[15]	0x500078	4	R/W	Sets the UserSet number to register to the 16th sequence.

● Control with TeliCamSDK

Camera feature API

Control SequentialShutter using dedicated API.

API name	Description
GetCamSequentialShutterEnable	Get activation status
SetCamSequentialShutterEnable	Set activation status
GetCamSequentialShutterTerminateAtMinMax	Get minimum and maximum value
GetCamSequentialShutterTerminateAt	Get current SequentialShutterTerminateAt value
SetCamSequentialShutterTerminateAt	Set new SequentialShutterTerminateAt value
GetCamSequentialShutterIndexMinMax	Get minimum and maximum value
GetCamSequentialShutterEntryMinMax	Get minimum and maximum value
GetCamSequentialShutterEntry	Get current SequentialShutterEntry value
SetCamSequentialShutterEntry	Set new SequentialShutterEntry value

GenICam function API

Control SequentialShutter using GenICam API.

◆SequentialShutter

1. Set the camera parameters and save them to UserSet. Repeat this as you required.
2. Set the sequence number to register to ’SequentialShutterIndex’ and Sets the UserSet number to register to the sequence to ‘SequentialShutterEntry’. Repeat this as you required.
3. Set the number of Index to repeat the sequence to ’SequentialShutterTerminateAt’.
4. Set the activation of SequentialShutter function by ‘SequentialShutterEnable’.

   Integer value and string value of Enumeration are as follows.

Integer	String
0	Off
1	On

5. Capture image stream.

   To start and stop capturing image is the same way as AcquisitionControl.

   Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.

```
// 1.Set the camera parameters and save them to UserSet.
...

// GenICam node handle
CAM_NODE_HANDLE   hIndex = NULL;
CAM_NODE_HANDLE   hEntry = NULL;
CAM_NODE_HANDLE   hTerminateAt = NULL;
CAM_NODE_HANDLE   hEnable = NULL;

// Retrieve GenICam node.
Nd_GetNode(s_hCam, "SequentialShutterIndex", &hIndex);
Nd_GetNode(s_hCam, "SequentialShutterEntry", &hEntry);
Nd_GetNode(s_hCam, "SequentialShutterTerminateAt", &hTerminateAt);
Nd_GetNode(s_hCam, "SequentialShutterEnable", &hEnable);

int64_t     entry[] = {1,4,2};       // Sequence = {UserSet1,UserSet4,UserSet2}
int64_t     index;

for(index=0; index<3; index++)
{
  // 2.Set SequentialShutterIndex and SequentialShutterEntry.
  Nd_SetIntValue(s_hCam, hIndex, index+1); // 1 origin
  Nd_SetIntValue(s_hCam, hEntry, entry[index]);
}

// 3.Set SequentialShutterTerminateAt.
Nd_SetIntValue(s_hCam, hTerminateAt, index); // 3 UserSet will be repeated.

// 4.Set the activation of SequentialShutter function by 'SequentialShutterEnable'.
Nd_SetEnumStrValue(s_hCam, hEnable, "On");
```

Please refer to [INode functions], [IInteger node functions] and [IEnumeration node functions] in [TeliCamAPI Library manual] for more detail.

Register access API

Control SequentialShutter by accessing IIDC2 registers directly.

API name	Description
Cam_ReadReg	Read register value
Cam_WriteReg	Write register value

1. Set the camera parameters and save them to UserSet. Repeat this as you required.
2. Sets the UserSet number to register to the sequence.
   Write to ‘Value[0]’ to ‘Value[15]’ of ‘SequentialShutterSequenceTable’.
   The index of ‘Value’ register means sequence index. (0 origin)
3. Set the number of Index to repeat the sequence.
   Write to ‘Value’ field of ’SequentialShutterTerminateAt’.
4. Set the activation of SequentialShutter function.
   Write to ‘Value’ field of ’SequentialShutterEnable’ register to activate LUT function.

5. Capture image stream.

To start and stop capturing image is the same way as AcquisitionControl.

Please refer to [Camera streaming functions] in [TeliCamAPI Library manual] and [GrabStreamSimple] sample code in your TeliCamSDK install folder.

```
// 1.Set the camera parameters and save them to UserSet.
...

uint64_t    addr = 0x500040;
uint32_t    entry[] = {1,4,2};        // Sequence = {UserSet1,UserSet4,UserSet2}
uint32_t    index, dat;

for(index=0; index<3; index++,addr+=4)
{
  // 2.Set SequentialShutterIndex and SequentialShutterEntry.
  Cam_WriteReg(s_hCam, addr, 1, &entry[index]);
}

// 3.Set SequentialShutterTerminateAt.
Cam_WriteReg(s_hCam, 0x21F33C, 1, &index); // 3 UserSet will be repeated.

// 4.Set the activation of SequentialShutter function.
dat = 1;
Cam_WriteReg(s_hCam, 0x21F31C, 1, &dat);
```

Please refer to [Camera functions] in [TeliCamAPI Library manual] for more detail.

◆Minimum/Maximum Value

SequentialShutterIndex	setting value
Minimum (*)	1
Maximum	16

* initial factory setting

SequentialShutterEntry	setting value
Minimum (*)	1
Maximum	15

* initial factory setting

- **Note**

The following table is the list of registers applied to “SequentialShutter”.

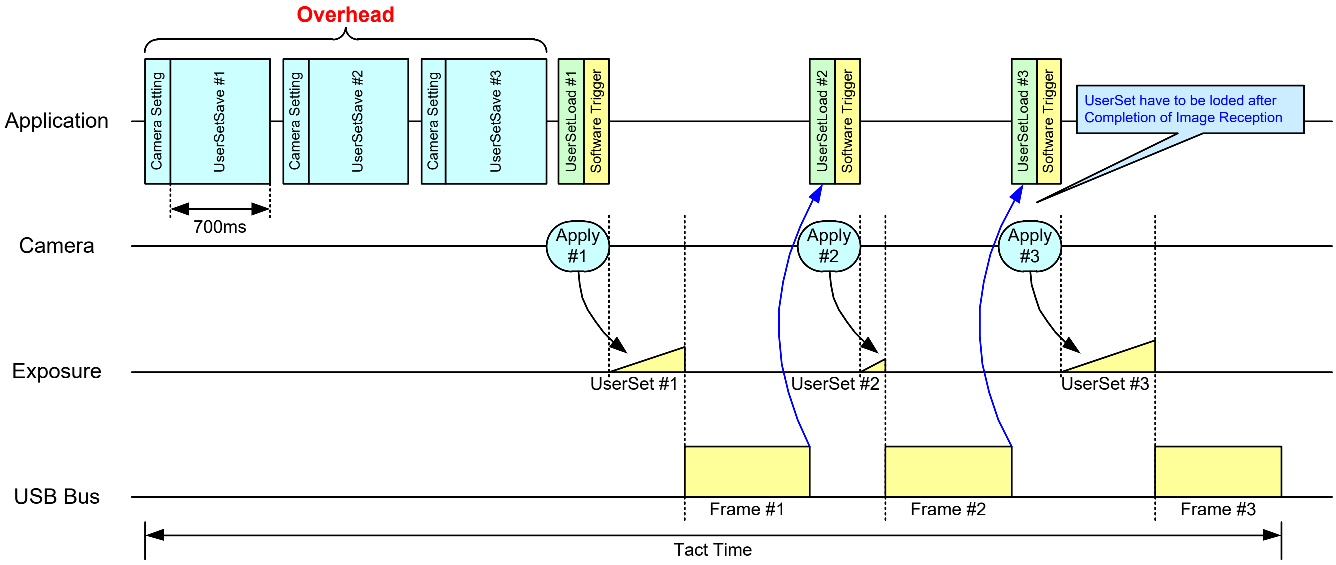
Table: List of registers to be applied to SequentialShutter

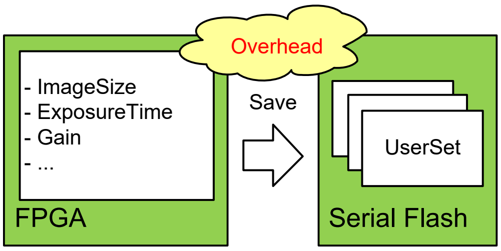
Category	Register
ImageFormatControl	OffsetX
	OffsetY
ExposureControl	ExposureTime
DigitalIOControl	UserOutputValueAll
	LineSource
CounterAndTimerControl	TimerDuration
	TimerDelay
AnalogControl	Gain
	BlackLevel
	Gamma
	Sharpness
	BalanceRatio
	ColorCorrectionMatrix
LUTControl	LUTEnable

Appendix

UserSetSave and UserSetQuickSave difference

If you want to change multiple Camera Settings (e.g. ROI setting) at once, UserSet feature is useful.
Application needs to save the camera settings in UserSet memory in advance or in initialization stage.
By **UserSetSave**, UserSet is stored to non-volatile flash memory.
It takes about **700ms** to execute UserSetSave because Non-volatile flash memory needs to be erased before writing. If you want to switch over multiple different settings in run time, overhead time of **UserSetSave** is inevitable.

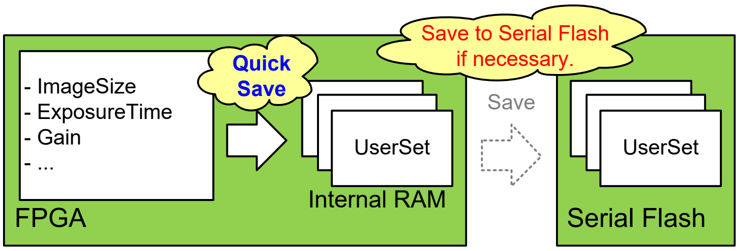




By **UserSetQuickSave**, UserSet is stored to internal RAM. It takes less than **100us** in execution.

It can reduce the overhead time of **UserSetSave** greatly.

You can also save UserSets to Serial Flash if necessary by **UserSetSave**.



MultiFrame and Bulk function difference

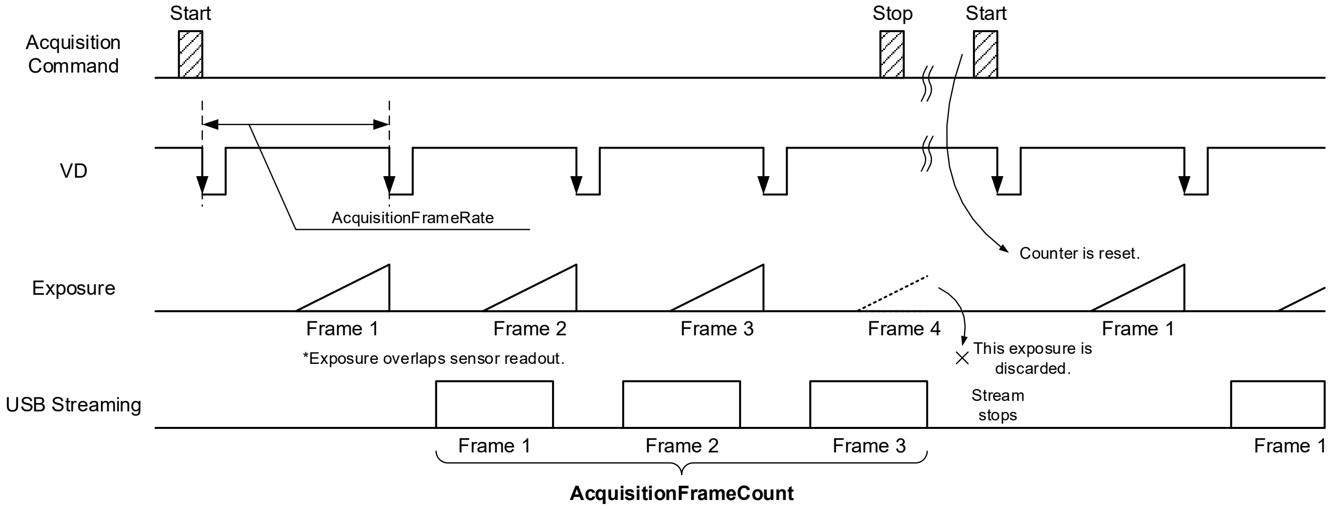
This section describes the difference between MultiFrame function of AcquisitionMode and Bulk function of TriggerSequence.

- MultiFrame function sets limits to the number of frames to transfer with AcquisitionFrameCount register.
- Bulk function sets limits to the number of frames to exposure with TriggerAdditionalParameter register.

- MultiFrame function in Normal Shutter mode (TriggerMode = Off)

Camera transfers “AcquisitionFrameCount” frame(s).

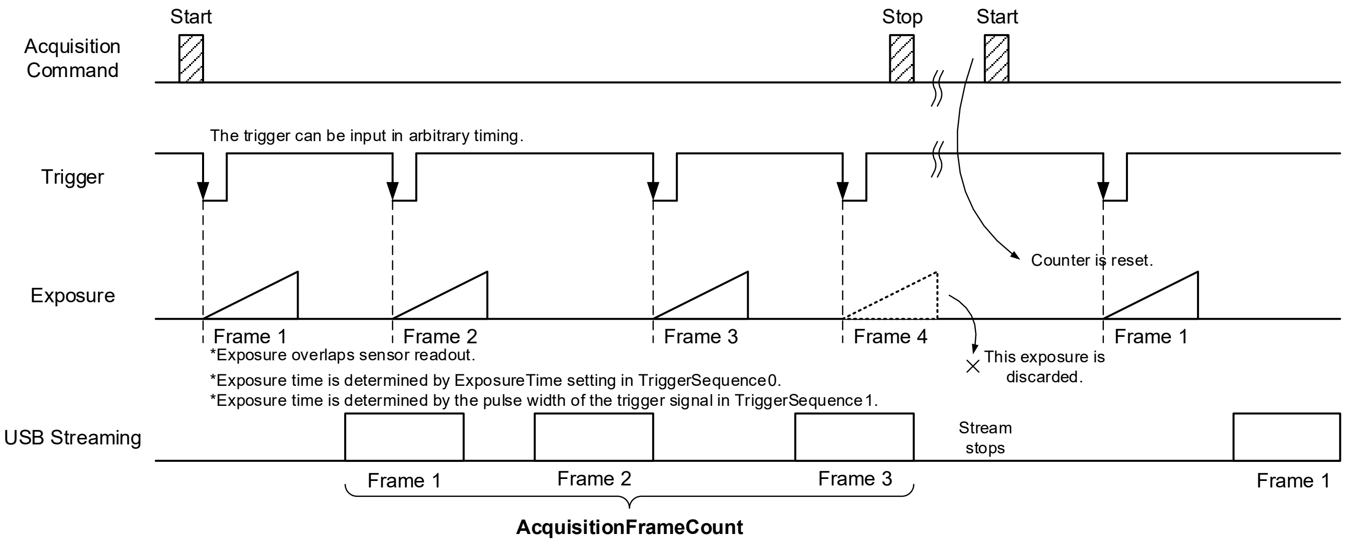
AcquisitionMode = MultiFrame
AcquisitionFrameCount = 3
TriggerMode = Off



- MultiFrame function in Random Trigger Shutter mode (TriggerMode = On, TriggerSequence = 0 or 1)

Camera transfers “AcquisitionFrameCount” frame(s). It requires “AcquisitionFrameCount” time(s) of trigger.

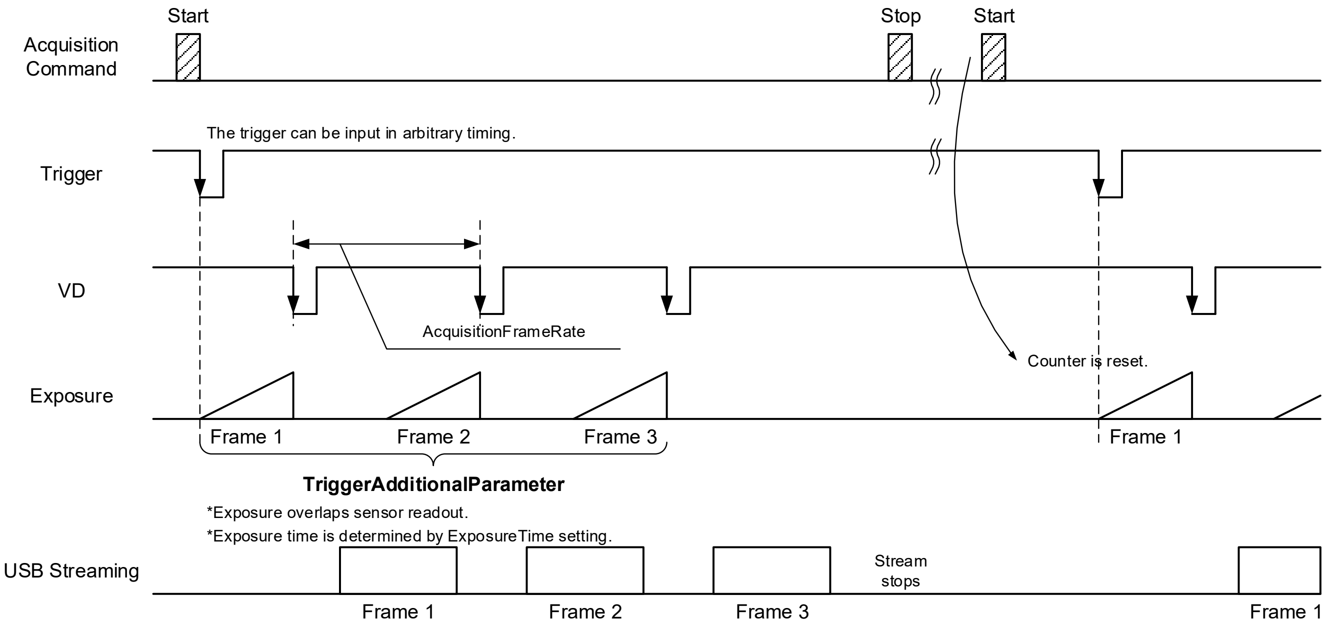
AcquisitionMode = MultiFrame
AcquisitionFrameCount = 3
TriggerMode = On
TriggerSequence = 0 or 1


Start
Stop
Start
Acquisition Command
The trigger can be input in arbitrary timing.
Trigger
Counter is reset.
Exposure
Frame 1
Frame 2
Frame 3
Frame 4
Frame 1
*Exposure overlaps sensor readout.
This exposure is discarded.
*Exposure time is determined by ExposureTime setting in TriggerSequence0.
*Exposure time is determined by the pulse width of the trigger signal in TriggerSequence1.
USB Streaming
Stream stops
Frame 1
Frame 2
Frame 3
Frame 1
AcquisitionFrameCount

- Bulk function (TriggerMode = On, TriggerSequence = 6)

Camera transfers “TriggerAdditionalParameter” frame(s) by a single trigger.

AcquisitionMode = Continuous
TriggerMode = On
TriggerSequence = 6
TriggerAdditionalParameter = 3

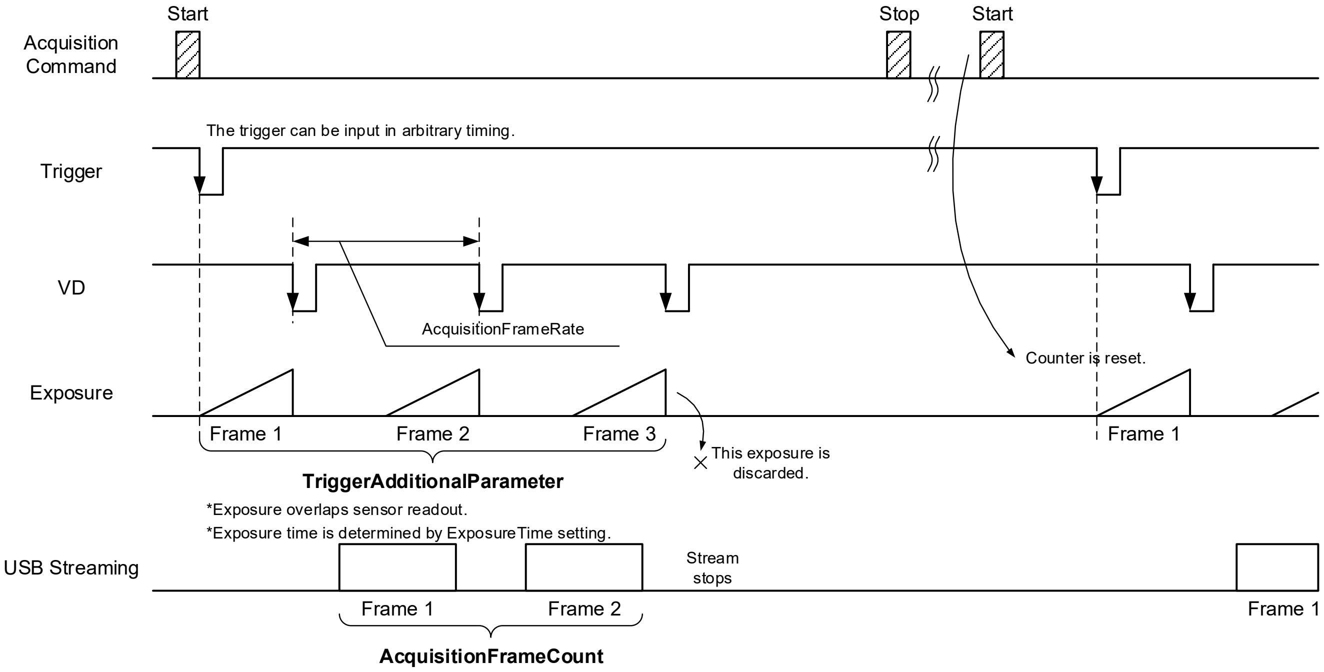


● Note

In Bulk function, you may want to set “Continuous” to “AcquisitionMode” register.

You can set “MultiFrame”, but acquisition frame number is limited by “AcquisitionFrameCount.

AcquisitionMode = MultiFrame
AcquisitionFrameCount = 2
TriggerMode = On
TriggerSequence = 6
TriggerAdditionalParameter = 3


Start
Stop
Start
Acquisition Command
The trigger can be input in arbitrary timing.
Trigger
VD
AcquisitionFrameRate
Counter is reset.
Exposure
Frame 1
Frame 2
Frame 3
Frame 1
This exposure is discarded.
TriggerAdditionalParameter
*Exposure overlaps sensor readout.
*Exposure time is determined by ExposureTime setting.
USB Streaming
Stream stops
Frame 1
Frame 2
Frame 1
AcquisitionFrameCount

Warranty rules

● Warranty term

Warranty term is 36 months after your purchase. We may assume the date of the purchase from our shipping date when the date is unidentified.

● Limited Warranty

Free warranty is not applicable for the troubles, damages or losses caused by the cases of the followings, even if it is during the warranty term.

1. Natural exhaust, wear or degradation of a component parts
2. Handling against the instructions and conditions described in the instruction manual
3. Remodeling, adjustment and the part exchange. (including the opening of the enclosure box and the alteration)
4. Using the accessories not included with the product or our non-designated optional articles
5. Damages caused during the transportation or deficiency of the handling such as drop or fall of the products after the products having been transferred to customers, leaving the products to corrosive environment such as sunlight, fire, sand, soil, heat, moisture, or an inappropriate storing method
6. A fire, an earthquake, a flood, a lightning, or other natural disasters, pollution and a short circuit, abnormal voltage, excessive physical pressure, theft, other accident
7. When connected to a product which is not recommended
8. When connected to the power supply which is not suitable
9. Forgery product, products which does not have proper serial number, products of which serial number is forged, damaged or deleted
10. All defects that happened after the expiration for a warranty term

Repair

● Repair methods

Exchange to a replacement or an equal function product.

● Repair request methods

On the occasion of a repair request, please download the "Failure situation report sheet" from our website, fill in the necessary items and return it together with the defective product.

Repair Request Methods
http://www.toshiba-teli.co.jp/en/support/contact/failure_situation.htm

1. Please read the following instructions carefully.Please return our product alone, taking out of your equipment in case that our product is installed to an equipment
2. We are unable to return the information such as your own serial numbers, control number, the identification seal, if it is attached to the returned products. Please keep record before you return the product.
3. As the data saved in the camera will not be kept after the repair, please take out data before return.
4. We are unable to accept the cancellation after the repair request by the customer's reason.
5. About the repair product shipping expenses, please bear the charges when you return the product to us. We bear the charges to you from us only for a warranty period.
6. We are unable to accept your request of a delivery date and time of the product return, or the delivery method.
7. We are unable to accept a trouble factor investigation, the request of the repair report.
8. We accept a repair of out of warranty product, if it is reparable.
9. The proprietary rights of the repair request products after the exchange repair belong to us.
10. The immunity from responsibility of the product is applied in the repair completion products.

* Please refer for the inquiry about the software to our homepage or sales personnel.