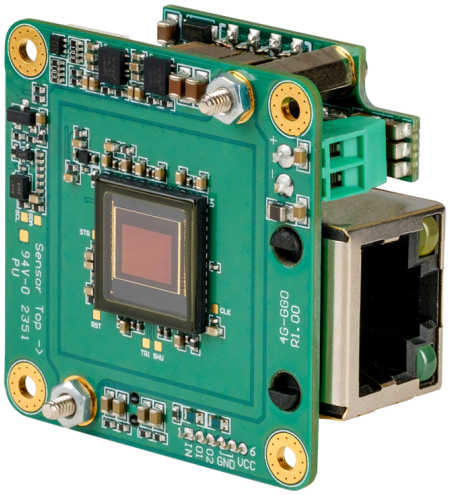
Technical Details



# DFM 47GX900-MLP
# Technical Reference Manual

THE IMAGINGSOURCE
TECHNOLOGY BASED ON STANDARDS

# Table of Contents


1. Quick Facts 5

2. Dimensional Diagrams 7

2.1 DFM 47GX900-MLP Board Camera with PoE ..... 7

3. I/O Connector 8

3.1 6-Pin I/O Connector ..... 8

4. Spectral Characteristics 9

4.1 Spectral Sensitivity - IMX900 ..... 9

5. Camera Controls 10

5.1 Sensor Readout Control ..... 10

5.1.1 Pixel Format ..... 10

5.1.1.1 8-Bit Bayer Raw ..... 11

5.1.1.2 12-Bit Packed Bayer Raw ..... 11

5.1.1.3 16-Bit Bayer Raw ..... 11

5.1.1.4 YUV 4:2:2 ..... 11

5.1.1.5 YUV 4:1:1 ..... 12

5.1.1.6 RGB24 ..... 12

5.1.1.7 8-Bit Monochrome ..... 12

5.1.2 Resolution ..... 12

5.1.3 Readout Modes ..... 13

5.1.4 Frame Rate ..... 14

5.1.5 Partial Scan Offset ..... 16

5.1.6 Image Flipping ..... 17

5.2 Image Sensor Control ..... 18

5.2.1 Exposure Time ..... 18

5.2.2 Gain ..... 18

5.2.3 Black Level ..... 19

5.3 Automatic Exposure and Gain Control ..... 19

5.3.1 Auto Exposure ..... 19

5.3.2 Auto Gain ..... 20

5.3.3 Auto Reference Value ..... 20

5.3.4 Highlight Reduction ..... 20

5.3.5 Auto Exposure Limits ..... 21

5.3.6 Auto Gain Limits ..... 21

5.4 Trigger ..... 22

5.4.1 Trigger Mode ..... 22

5.4.2 Trigger Polarity ..... 22

5.4.3 Software Trigger ..... 23

5.4.4 Trigger Exposure Mode ..... 23

5.4.5 Trigger Burst Count ..... 24

5.4.6 Trigger Source ..... 24

5.4.7 IMX Trigger Timing ..... 24
5.4.8 Trigger Overlap ..... 25
5.4.9 Trigger Delay ..... 25
**5.5 Trigger Timing Parameters ..... 26**
5.5.1 Trigger Debounce Time ..... 26
5.5.2 Trigger Mask Time ..... 26
5.5.3 Trigger Noise Suppression Time ..... 27
**5.6 Digital I/O ..... 27**
5.6.1 Line Selector ..... 27
5.6.2 Line Name ..... 28
5.6.3 Line Mode ..... 28
5.6.4 Line Inverter ..... 28
5.6.5 Line Status ..... 29
5.6.6 Line Status Invalidate ..... 29
5.6.7 Line Source ..... 29
5.6.8 Line Format ..... 30
5.6.9 User Output Selector ..... 30
5.6.10 User Output Value ..... 31
**5.7 Strobe ..... 31**
5.7.1 Strobe Enable ..... 31
5.7.2 Strobe Polarity ..... 31
5.7.3 Strobe Operation ..... 32
**5.8 White Balance ..... 32**
5.8.1 Auto White Balance ..... 32
5.8.2 White Balance Mode ..... 33
5.8.3 Manual White Balance ..... 34
**5.9 Image Processing ..... 36**
5.9.1 Gamma ..... 36
5.9.2 Lookup Table ..... 37
**5.10 Color Processing ..... 38**
5.10.1 Hue ..... 38
5.10.2 Saturation ..... 38
5.10.3 Color Correction Matrix ..... 39
**5.11 Region of Interest for Auto Functions ..... 42**
5.11.1 Auto Functions ROI Enable ..... 42
5.11.2 Auto Functions ROI Preset ..... 42
5.11.3 Auto Functions ROI Custom Rectangle ..... 43
**5.12 User Sets ..... 44**
5.12.1 User Set Selector ..... 44
5.12.2 Load User Set ..... 45
5.12.3 Save User Set ..... 45
5.12.4 Default User Set ..... 45
**5.13 Timestamp Control ..... 45**

5.13.1 Timestamp Latch ..... 46
5.13.2 Timestamp Latch Value ..... 46
5.13.3 Timestamp Latch String ..... 46
5.13.4 Timestamp Reset ..... 46
**5.14 Precision Time Protocol ..... 47**
5.14.1 PTP Enable ..... 47
5.14.2 PTP Status ..... 47
**5.15 Action Scheduler ..... 48**
5.15.1 Action Selector ..... 48
5.15.2 Action Scheduler Time ..... 48
5.15.3 Action Scheduler Interval ..... 49
5.15.4 Action Scheduler Commit ..... 49
5.15.5 Action Scheduler Cancel ..... 49
**5.16 Events ..... 50**
5.16.1 Event Selector ..... 50
5.16.2 Event Notification ..... 50
5.16.3 Event Exposure End ..... 51
5.16.4 Event Frame Trigger Missed ..... 51
5.16.5 Event Line1 Falling Edge ..... 52
5.16.6 Event Line1 Rising Edge ..... 52
5.16.7 Event Test ..... 53
**5.17 Chunk Data ..... 53**
5.17.1 Chunk Mode Active ..... 54
5.17.2 Chunk Selector ..... 54
5.17.3 Chunk Enable ..... 54
5.17.4 Chunk Exposure Time ..... 55
5.17.5 Chunk Gain ..... 55
**5.18 Bandwidth Control ..... 55**
5.18.1 Device Link Throughput Limit Mode ..... 55
5.18.2 Device Link Throughput Limit ..... 56

**6. Temperature Measurement Point 57**

**7. Status LEDs 58**

**7.1 On-Board LED ..... 58**
**7.2 LEDs on the RJ45 Connector ..... 58**

**8. Revision History 59**

# 1 Quick Facts

| General | |
|---|---|
| Vision Standard | GigE Vision |
| Dynamic Range | 12 bit |
| Resolution | 2048x1536 |
| Frame Rate at Full Resolution | 36 |
| Pixel Formats | 8-Bit Bayer (RG)<br>12-Bit Bayer Packed (RG)<br>16-Bit Bayer (RG)<br>YUV 4:2:2<br>YUV 4:1:1<br>YUV 4:1:1<br>RGB24<br>8-Bit Monochrome |

| Optical Interface | |
|---|---|
| IR-Cut filter | No |
| Sensor Type | Sony IMX900 |
| Shutter Type | Global |
| Sensor Format | 1/3.1 inch |
| Pixel Size | 2.25 µm |
| Lens Mount | M12 |

| Electrical Interface | |
|---|---|
| Interface | GigE |
| Supply Voltage | PoE: 48 VDC to 56 VDC |
| Power Consumption | approx 2.3 W PoE |

| Mechanical Data | |
|---|---|
| Dimensions | H: 36 mm, W: 38 mm, L: 18 mm |
| Mass | 20 g |

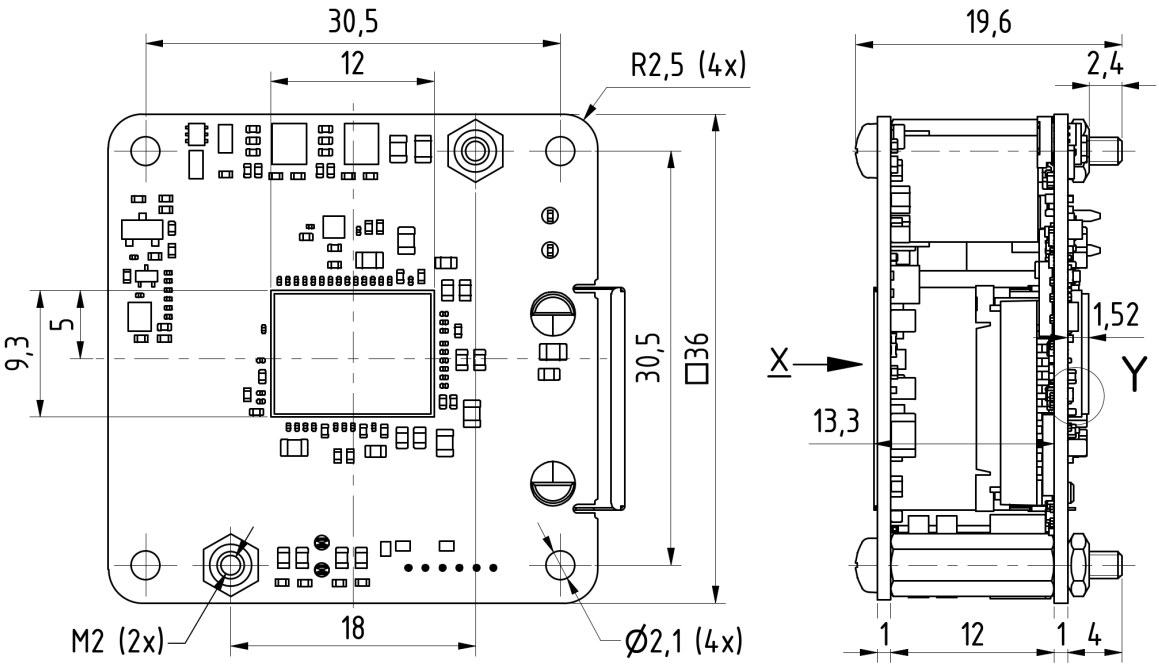
| Adjustments | |
|---|---|
| Shutter | 15 µs to 1 s |
| Gain | 0 dB to 48 dB |

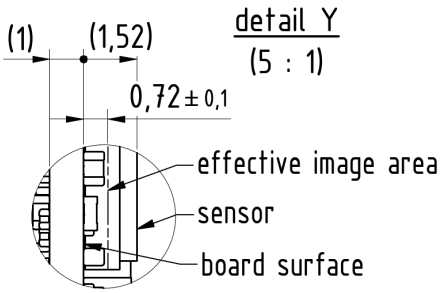
| Environmental Conditions | |
|---|---|
| Board Temperature (operating) * | 0 °C to 70 °C |
| Temperature (storage) | -20 °C to 80 °C |
| Humidity (operating) | 20 % to 80 % (non-condensing) |
| Humidity (storage) | 20 % to 95 % (non-condensing) |

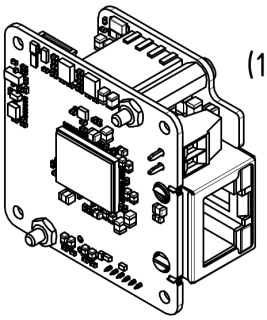
*) See section Temperature Measurement Point for details.

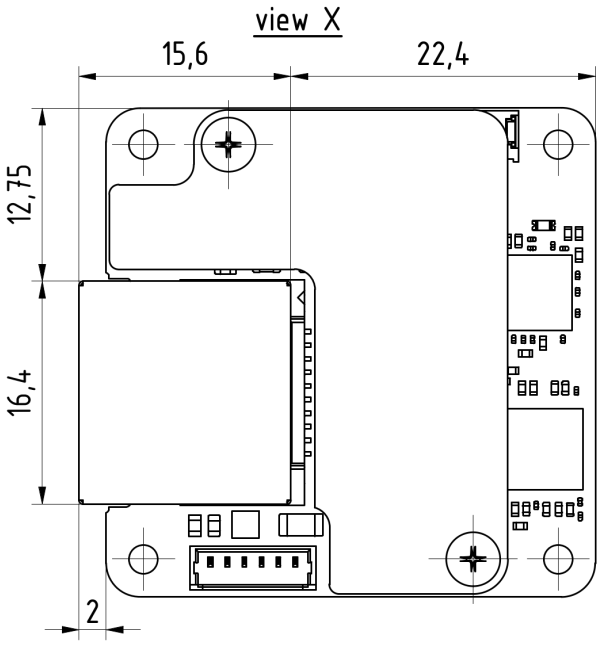
# 2 Dimensional Diagrams

## 2.1 DFM 47GX900-MLP Board Camera with PoE



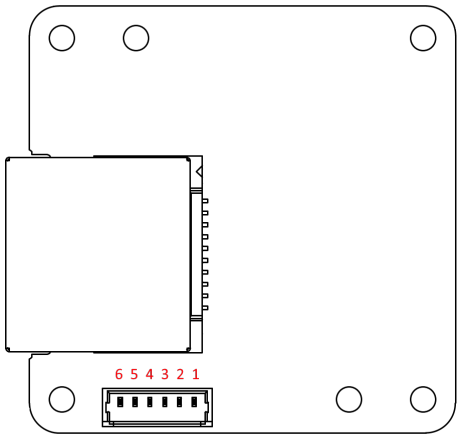




Dimensions: mm
Tolerances: DIN ISO 2768-f

306-23-11-00-00

# 3 I/O Connector

## 3.1 6-Pin I/O Connector




Rear view of camera

| Pin | Signal | Type[1] | Remarks | Min | Max | LineSelector[2] |
|---|---|---|---|---|---|---|
| 1 | TRIGGER_IN | I | min 0.15mA @ 3.0V | 3.0 V | 25 V | Line1 |
| 2 | GPOUT1 | O | Open Drain, max 0.5A | | 25 V | Line0 |
| 3 | GPOUT2 | O | Open Drain, max 0.5A | | 25 V | Line2 |
| 4 | GND | G | | | | - |
| 5 | GND | G | | | | - |
| 6 | | N/C | Do not connect | | | - |

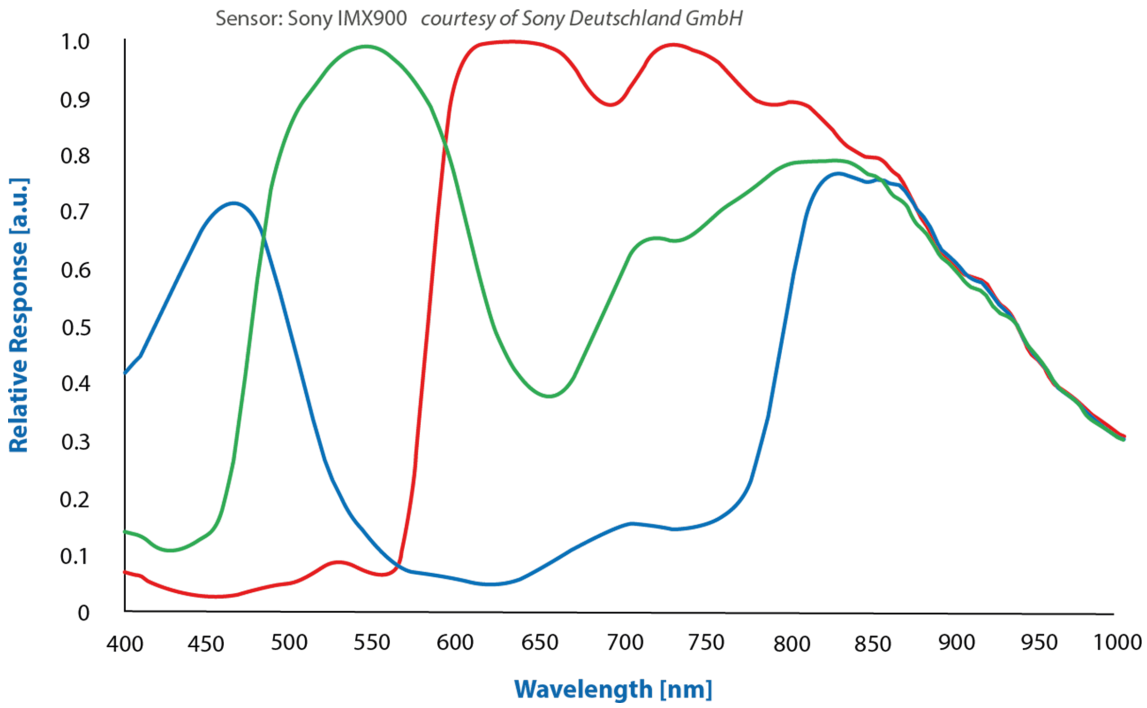
[1] G: Ground, O: Output, I: Input

[2] Set Line Selector to configure or query this pin.

The part number of this PicoBlade connector is Molex-53047-0610. To create an I/O cable, a PicoBlade connector (Molex-51021-0600) or a PicoBlade cable assembly (e.g. Molex-15134-0600) is required.

# 4 Spectral Characteristics

## 4.1 Spectral Sensitivity - IMX900



# 5 Camera Controls

This section describes the parameters available for the DFM 47GX900-MLP camera.

The actual name of the parameter depends on the driver technology used to access the camera. Parameter names are listed for the most common ways to access the cameras:

- *GigE Vision* (*IC Imaging Control 4*, or via 3rd party driver)
- *IC Imaging Control 3.x* (on Windows, via *Device Driver for The Imaging Source GigE Cameras*)

## 5.1 Sensor Readout Control

### 5.1.1 Pixel Format

The pixel format defines the data type of the pixels transmitted to the computer. The bits per pixel needed for a particular pixel format influence the required bandwith.

The driver technology used to access the camera significantly impacts the way the pixel format is controlled:

- When using *IC Imaging Control 4* or *GigE Vision*, the pixel format is controlled through the `PixelFormat` *GenICam* feature.
- When using *IC Imaging Control 3.x*, the pixel format is part of the video format - a parameter which combines pixel format, resolution and readout mode. For more information, refer to the *IC Imaging Control 3.x* documentation sections on `VideoFormat` and `VideoFormatDesc`.

The DFM 47GX900-MLP color camera supports multiple pixel formats with variable bits-per-pixel settings. The names of the pixel formats and the way to select them depends on the driver used to control the camera. The following table contains a short overview of all possible formats followed by a more detailed description.

| Pixel Format | Bits Per Pixel | GigEVision | TIS GigE Driver |
|---|---|---|---|
| 8-Bit Bayer (RG) | 8 | `BayerRG8` | `RGB32, Y800` |
| 12-Bit Packed Bayer (RG) | 12 | `BayerRG12p` | `RGB64, Y16` |
| 16-Bit Bayer (RG) | 16 | `BayerRG16` | `RGB64, Y16` |
| YUV 4:2:2 | 16 | `YUV422_8` | `YUY2` |
| YUV 4:1:1 (a) | 12 | `YCbCr411_8` | |
| YUV 4:1:1 (b) | 12 | `YCbCr411_8_CbYYCrYY` | `Y411` |
| RGB24 | 24 | `BGR8` | `RGB24` |
| 8-Bit Monochrome | 8 | `Mono8` | |

#### 5.1.1.1 8-Bit Bayer Raw

This format transmits data using one byte for each pixel.

*GigEVision* drivers see this pixel format as one of the 8-bit bayer raw formats (`BayerGR8`, `BayerRG8`, `BayerGB8`, `BayerBG8`).

The *Device Driver for The Imaging Source GigE Cameras* simplifies this variety of possible pixel formats and offers two video formats instead: `RGB32` and `Y800`. The `RGB32` format results from the driver's automatic debayering of the raw image data, while the `Y800` format contains the raw data which is reinterpreted as monochrome.

#### 5.1.1.2 12-Bit Packed Bayer Raw

This format transmits data using 3 bytes for each pair of 2 consecutive pixels.

*GigEVision* drivers see this pixel format as one of the 12-bit packed bayer raw formats (`BayerGR12Packed`, `BayerRG12Packed`, `BayerGB12Packed`, `BayerBG12Packed`).

The *Device Driver for The Imaging Source GigE Cameras* simplifies this variety of possible pixel formats and offers two video formats instead: `RGB64` and `Y16`. The `RGB64` format results from the driver's automatic debayering of the raw image data, while the `Y16` format contains the raw data which is reinterpreted as monochrome. Since the DFM 47GX900-MLP camera offers both 12-bit packed and 16-bit bayer raw pixel formats, the driver uses the 12-bit packed format when the `RGB64` or `Y16` video format is selected.

#### 5.1.1.3 16-Bit Bayer Raw

This format transmits data using 2 bytes for each pixel.

The sensor of the DFM 47GX900-MLP camera is not able to provide 16-bit data output. The pixel data is transmitted in the most significant bits. This allows application programs to ignore the sensor-specific data type and to treat the data as if the sensor outputs 16 bits.

*GigEVision* drivers see this pixel format as one of the 16-bit packed bayer raw formats (`BayerGR16`, `BayerRG16`, `BayerGB16`, `BayerBG16`).

The *Device Driver for The Imaging Source GigE Cameras* simplifies this variety of possible pixel formats and offers two video formats instead: `RGB64` and `Y16`. The `RGB64` format results from the driver's automatic debayering the raw image data, while the `Y16` format contains the raw data which is reinterpreted as monochrome.

#### 5.1.1.4 YUV 4:2:2

This format transmits an 8-bit luminance value for every pixel and a pair of chrominance values for every two pixels.

The camera converts the raw pixel data from the sensor to YUV 4:2:2. Since YUV 4:2:2 averages to 2 bytes per pixel, the YUV 4:2:2 pixel format requires two times the bandwidth of the 8-bit raw format, thus reducing the maximum possible frame rate.

*GigEVision* drivers see this pixel format as `YUV422_8`.

The *Device Driver for The Imaging Source GigE Cameras* offers this pixel format as the `YUY2` video format.

#### 5.1.1.5 YUV 4:1:1

This format transmits an 8-bit luminance value for every pixel and a pair of chrominance values for every four pixels.

The two YUV 4:1:1 pixel formats only differ in their byte order.

The camera converts the raw pixel data from the sensor to YUV 4:2:2. Since YUV 4:2:2 averages to 1.5 bytes per pixel, the YUV 4:2:2 pixel format requires two 50% more bandwidth than the 8-bit raw format, thus reducing the maximum possible frame rate.

*GigEVision* drivers see this pixel format as `YCbCr411_8` and `YCbCr411_8_CbYYCrYY`.

The *Device Driver for The Imaging Source GigE Cameras* offers this pixel format as the `Y411` video format.

#### 5.1.1.6 RGB24

The camera converts the raw pixel data from the sensor to RGB24. Since RGB24 uses 3 bytes to store 1 pixel, the RGB24 pixel format requires three times the bandwidth of the 8-bit raw format, thus reducing the maximum possible frame rate.

*GigEVision* drivers enumerate this pixel format as `BGR8`.

The *Device Driver for The Imaging Source GigE Cameras* enumerates this pixel format as the `RGB24` video format.

#### 5.1.1.7 8-Bit Monochrome

The camera generates the pixel data by debayering the raw pixel data and transmitting only the luminance information.

*GigEVision* drivers enumerate this pixel format as `Mono8`.

### 5.1.2 Resolution

The DFM 47GX900-MLP allows the user to specify which rectangular region of the image sensor to read out during camera operation. The size of this rectangle determines the number of pixels that have to be transferred for each frame and has a significant influence on the required network bandwidth.

Lowering the resolution also often allows the image sensor to operate at a higher frame rate. Changes in the vertical resolution have more effect on the maximum frame rate than changes in the horizontal direction.

The way the resolution is controlled varies greatly between the driver technology used to access the camera:

- When using *GigE Vision*, the resolution is controlled through the GenICam features Width and Height.

- When using *IC Imaging Control*, the resolution is part of the video format, a parameter that combines pixel format, resolution and readout mode. For more information, refer to the *IC Imaging Control* documentation sections on `VideoFormat` and `VideoFormatDesc`.

| Parameter | Horizontal Resolution |
|---|---|
| Minimum | 256 |
| Maximum | 2048 |
| Increment | 8 |
| GigEVision: GenICam | `Width` |
| Type | Integer |

| Parameter | Vertical Resolution |
|---|---|
| Minimum | 256 |
| Maximum | 1536 |
| Increment | 8 |
| GigEVision: GenICam | `Height` |
| Type | Integer |

### 5.1.3 Readout Modes

The DFM 47GX900-MLP camera offers different sensor readout modes.

By default, the camera outputs all pixels of the image sensor.

In *skipping* modes, the image sensor regularly skips a number of rows and/or columns during readout. The resulting output image is smaller but contains the same field of view than the image produced by default readout mode.

The way the readout modes are controlled depends greatly upon which driver technology is used to access the camera:

- When using *GigE Vision*, the readout mode is controlled through the *GenICam* features `DecimationHorizontal`, `DecimationVertical`, `BinningHorizontal` and `BinningVertical`.
- When using *IC Imaging Control*, the readout mode is part of the video format - a parameter that which combines pixel format, resolution and readout mode. For more information, refer to the *IC Imaging Control* documentation sections on `VideoFormat` and `VideoFormatDesc`.

The following readout modes are supported:

- Default
- Skipping 2X

| Parameter | Skipping Horizontal |
|---|---|
| Valid Values | 1, 2 |
| GigEVision: GenICam | `DecimationHorizontal` |
| Type | Integer |

| Parameter | Skipping Vertical |
|---|---|
| Valid Values | 1, 2 |
| GigEVision: GenICam | `DecimationVertical` |
| Type | Integer |

### 5.1.4 Frame Rate

The frame rate is specified in frames per second and determines the camera's operating speed.

The way the frame rate is controlled depends greatly upon which driver technology is used to access the camera:

- When using *GigEVision*, the frame rate is controlled through the *GenICam* feature `AcquisitionFrameRate`.
- When using IC Imaging Control, the frame rate is selected from a list of available frame rates through APIs such as `Grabber::setFPS` or `ICImagingControl.DeviceFrameRate`.

The range of available frame rates depends upon other camera settings such as well, pixel format, resolution and readout modes.

| Parameter | Frame Rate |
|---|---|
| Minimum | Depending on Pixel Format, Resolution, and Readout Mode |
| Maximum | Depending on Pixel Format, Resolution, and Readout Mode |
| GigEVision: GenICam | `AcquisitionFrameRate` |
| Type | Float |

**Please note:** Enabling Device Link Throughput Limit Mode can limit the frame rate below the configured value.

The following tables show the maximum frame rate for some combinations of pixel format and resolution.

**8-Bit Bayer Raw**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 36 |
| 1920 | 1080 | 54 |
| 640 | 480 | 175 |

**12-Bit Packed Bayer Raw**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 24 |
| 1920 | 1080 | 36 |
| 640 | 480 | 150 |

**16-Bit Bayer Raw**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 18 |
| 1920 | 1080 | 27 |
| 640 | 480 | 150 |

**YUV 4:2:2**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 18 |
| 1920 | 1080 | 27 |
| 640 | 480 | 175 |

**YUV 4:1:1 (a)**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 24 |
| 1920 | 1080 | 36 |
| 640 | 480 | 175 |

**YUV 4:1:1 (b)**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 24 |
| 1920 | 1080 | 36 |
| 640 | 480 | 175 |

**RGB24**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 12 |
| 1920 | 1080 | 18 |
| 640 | 480 | 122 |

**8-Bit Monochrome**

| Width | Height | Maximum Frame Rate |
|---|---|---|
| 2048 | 1536 | 36 |
| 1920 | 1080 | 54 |
| 640 | 480 | 175 |

### 5.1.5 Partial Scan Offset

If the selected resolution is smaller than the sensor size, the part of the sensor that is actually read out can be specified by the *Partial Scan Offset X* and *Partial Scan Offset Y* parameters. By default, the camera automatically positions the offsets so that the center of the sensor is used.

| Parameter | Partial Scan Offset X |
|---|---|
| Minimum | 0 |
| Maximum | 1792 |
| Increment | 8 |
| GigEVision: GenICam | `OffsetX` |
| Type | Integer |
| VCD Property | `VCDID_PartialScanOffset` `\VCDElement_PartialScanOffsetX` |

| Parameter | Partial Scan Offset Y |
|---|---|
| Minimum | 0 |
| Maximum | 1280 |
| Increment | 8 |
| GigEVision: GenICam | `OffsetY` |
| Type | Integer |
| VCD Property | `VCDID_PartialScanOffset \VCDElement_PartialScanOffsetY` |

If *Partial Scan Offset X* or *Partial Scan Offset Y* is configured to a value that would be invalid with the current setting of Resolution setting, the camera uses the maximum possible value.

| Parameter | Partial Scan Auto Center |
|---|---|
| `On` | Automatically configure Partial Scan Offset X/Y so that the center area of the sensor is read out |
| `True` | |
| `Off` | Control Partial Scan Offset X/Y manually |
| `False` | |
| GigEVision: GenICam | `OffsetAutoCenter` |
| Type | Boolean |
| VCD Property | `VCDID_PartialScanOffset \VCDElement_PartialScanAutoCenter` |

### 5.1.6 Image Flipping

The DFM 47GX900-MLP camera allows flipping of the image data horizontally, vertically, or both. The *Flip Horizontal* and *Flip Vertical* parameters can be used to control this sensor feature.

| Parameter | Flip Horizontal |
|---|---|
| `True` | Image data is flipped horizontally |
| `False` | Image data is not flipped horizontally |
| GigEVision: GenICam | `ReverseX` |
| Type | Boolean |
| VCD Property | `VCDID_FlipHorizontal\VCDElement_Value` |

| Parameter | Flip Vertical |
|---|---|
| `True` | Image data is flipped vertically |
| `False` | Image data is not flipped vertically |
| GigEVision: GenICam | `ReverseY` |
| Type | Boolean |
| VCD Property | `VCDID_FlipVertical\VCDElement_Value` |

## 5.2 Image Sensor Control

### 5.2.1 Exposure Time

The *Exposure Time* parameter defines the time the camera opens its (electronic) shutter when it is taking an image.

| Parameter | Exposure Time |
|---|---|
| Minimum | Dynamic, as low as 15 µs |
| Maximum | 1 s |
| Default | auto |
| GigEVision: GenICam | `ExposureTime` |
| Type | Float |
| VCD Property | `VCDID_Exposure\VCDElement_Value` |

**Dynamic minimum:** The minimum value depends on other settings such as IMX Trigger Timing, Trigger Mode, *Width*, and *Pixel Format*.

### 5.2.2 Gain

The *Gain* parameter defines the amplification that is applied to the image at sensor level.

| Parameter | Gain |
|---|---|
| Minimum | 0 dB |
| Maximum | 48 dB |
| Default | auto |
| GigEVision: GenICam | `Gain` |
| Type | Float |
| VCD Property | `VCDID_Gain\VCDElement_Value` |

### 5.2.3 Black Level

The *Black Level* parameter defines the lowest possibly intensity value of the image sensor.

| Parameter | Black Level |
|---|---|
| Minimum | 0 |
| Maximum | 4095 |
| Default | 0 |
| GigEVision: GenICam | `BlackLevel` |
| Type | Float |
| VCD Property | `VCDID_Brightness\VCDElement_Value` |

## 5.3 Automatic Exposure and Gain Control

The DFM 47GX900-MLP camera can automatically control gain and exposure time. These automatic functions are enabled by default.

In order to optimize image parameters, a region of interest can be specified for automatic functions. Specifying a region of interest enables fine-grained control over the image area for which the image parameters are optimized. A selection of pre-defined area presets is available, but the user can also specify the coordinates of a custom rectangle.

In certain situations, it is desirable to limit the range of the auto-controlled parameters. For example, one might want to avoid high gain settings in order to keep noise levels low. Other applications require limiting the maximum exposure time so that movements do not get blurred. Therefore, the ranges of the gain and exposure parameter can be limited.

If both auto exposure and auto gain are active, the camera tries to lower the gain value in order to reduce noise and improve the image quality.

### 5.3.1 Auto Exposure

| Parameter | Auto Exposure |
|---|---|
| `Continuous` | Enable Auto Exposure |
| `True` | |
| `Off` | Disable Auto Exposure |
| `False` | |
| GigEVision: GenICam | `ExposureAuto` |
| Type | Enumeration |
| VCD Property | `VCDID_Exposure\VCDElement_Auto` |

### 5.3.2 Auto Gain

| Parameter | Auto Gain |
|---|---|
| `Continuous` | Enable Auto Gain |
| `True` | |
| `Off` | Disable Auto Gain |
| `False` | |
| GigEVision: GenICam | `GainAuto` |
| Type | Enumeration |
| VCD Property | `VCDID_Gain\VCDElement_Auto` |

### 5.3.3 Auto Reference Value

The *Auto Reference Value* parameter specifies the target brightness for both auto exposure and auto gain.

| Parameter | Auto Reference Value |
|---|---|
| Minimum | 0 |
| Maximum | 255 |
| Default | 64 |
| GigEVision: GenICam | `ExposureAutoReference` |
| Type | Integer |
| VCD Property | `VCDID_Exposure\VCDElement_AutoReference` |

### 5.3.4 Highlight Reduction

Enabling *Highlight Reduction* lets the auto exposure and auto gain functions reduce overexposed areas in the output image. This feature is particularly useful when using 10/12/16-bit output images and a tone mapping algorithm in post-processing when using higher bit depths as the dark areas still contain a lot of useful information.

| Parameter | Highlight Reduction |
|---|---|
| `True` | Try to reduce overexposed areas |
| `False` | Ignore overexposed areas and focus on matching the image brightness to the selected Auto Reference Value. |
| GigEVision: GenICam | `ExposureAutoHighlightReduction` |
| Type | Boolean |
| VCD Property | `VCDID_HighlightReduction\VCDElement_Value` |

### 5.3.5 Auto Exposure Limits

The *Auto Exposure Lower Limit* parameter determines the minimum possible value that can be set by the auto exposure algorithm.

| Parameter | Auto Exposure Lower Limit |
|---|---|
| Minimum | 15 µs |
| Maximum | 1 s |
| GigEVision: GenICam | `ExposureAutoLowerLimit` |
| Type | Float |

The *Auto Exposure Upper Limit* parameter determines the maximum possible value that can be set by the auto exposure algorithm.

| Parameter | Auto Exposure Upper Limit |
|---|---|
| Minimum | 15 µs |
| Maximum | 1 s |
| GigEVision: GenICam | `ExposureAutoUpperLimit` |
| Type | Float |
| VCD Property | `VCDID_Exposure\VCDElement_AutoMaxValue` |

If the *Auto Exposure Upper Limit Auto* parameter is enabled, the value of *Auto Exposure Upper Limit* is automatically kept at the maximum possible value for the current frame rate.

| Parameter | Auto Exposure Upper Limit Auto |
|---|---|
| `On` | Select *Auto Exposure Upper Limit* automatically |
| `Off` | Let the user control *Auto Exposure Upper Limit* |
| GigEVision: GenICam | `ExposureAutoUpperLimitAuto` |
| Type | Enumeration |
| VCD Property | `VCDID_Exposure\VCDElement_AutoMaxValueAuto` |

### 5.3.6 Auto Gain Limits

The *Auto Gain Lower Limit* parameter determines the minimum possible value that can be set by the auto gain algorithm.

| Parameter | Auto Gain Lower Limit |
|---|---|
| Minimum | 0 dB |
| Maximum | 48 dB |
| GigEVision: GenICam | `GainAutoLowerLimit` |
| Type | Float |

The *Auto Gain Upper Limit* parameter determines the maximum possible value that can be set by the auto gain algorithm.

| Parameter | Auto Gain Upper Limit |
|---|---|
| Minimum | 0 dB |
| Maximum | 48 dB |
| GigEVision: GenICam | `GainAutoUpperLimit` |
| Type | Float |

## 5.4 Trigger

The trigger mode can be used to take images at very specific points in time which are specified by an electrical signal connected to the TRIGGER_IN pin of the I/O connector of the camera.

### 5.4.1 Trigger Mode

The *Trigger Mode* parameter enables the trigger mode.

| Parameter | Trigger Mode |
|---|---|
| `On` | Enable Trigger Mode |
| `True` | |
| `Off` | Disable Trigger Mode |
| `False` | |
| GigEVision: GenICam | `TriggerMode` |
| Type | Enumeration |
| VCD Property | `VCDID_Trigger\VCDElement_Value` |

### 5.4.2 Trigger Polarity

The *Trigger Polarity* parameter controls whether a trigger event is accepted on the rising or falling edge of the signal connected to the TRIGGER_IN line.

| Parameter | Trigger Polarity |
|---|---|
| `RisingEdge` | Accept rising edge as trigger signal |
| `True` | |
| `FallingEdge` | Accept falling edge as trigger signal |
| `False` | |
| GigEVision: GenICam | `TriggerActivation` |
| Type | Enumeration |
| VCD Property | `VCDID_Trigger\VCDElement_Polarity` |

### 5.4.3 Software Trigger

The *Software Trigger* function can be used to simulate a trigger pulse, in turn causing one image to be exposed and delivered to the host computer.

| Parameter | Software Trigger |
|---|---|
| Execute | Simulate one trigger pulse |
| GigEVision: GenICam | `TriggerSoftware` |
| Type | Command |
| VCD Property | `VCDID_Trigger\VCDElement_SoftwareTrigger` |

### 5.4.4 Trigger Exposure Mode

The *Trigger Exposure Mode* parameter sets the way in which the exposure time is controlled in trigger mode. By default, the exposure time is controlled through the Exposure Time parameter. The exposure time can also be controlled by the time the TRIGGER_IN line is kept in active state.

| Parameter | Trigger Exposure Mode |
|---|---|
| `FrameStart` | The exposure time is controlled by the Exposure Time parameter |
| `Timed` | |
| `ExposureActive` | The exposure time is controlled by the pulse length on the TRIGGER_IN pin |
| `Trigger Width` | |
| GigEVision: GenICam | `TriggerSelector` |
| Type | Enumeration |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerExposureMode` |

### 5.4.5 Trigger Burst Count

The *Trigger Burst Count* parameter allows the camera to be configured to take more than one image per trigger pulse. By default, this parameter is set to 1 so that exactly one image is acquired. The images are taken as rapidly as possible given current settings (i.e. as limited by the current exposure time and frame rate settings).

| **Parameter** | **Trigger Burst Count** |
|---|---|
| Minimum | 1 |
| Maximum | 1000 |
| Default | 1 |
| GigEVision: GenICam | `AcquisitionBurstFrameCount` |
| Type | Integer |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerBurstCount` |

### 5.4.6 Trigger Source

The *Trigger Source* parameter allows the camera to be configured to only accept trigger signals from a specified source, e.g. only software trigger.

| **Parameter** | **Trigger Source** |
|---|---|
| `Any` | Allow trigger signals from any source |
| `Line1` | Allow only hardware trigger |
| `Software` | Allow only software trigger |
| GigEVision: GenICam | `TriggerSource` |
| Type | Enumeration |

### 5.4.7 IMX Trigger Timing

The *IMX Trigger Timing* parameter configures timing details of the trigger operation.

| Parameter | IMX Trigger Timing |
|---|---|
| `MinimizeTriggerDelay` | This mode allows the lowest possible Trigger Delay, but the minimum Exposure Time at full resolution increases to 208 µs. |
| `MinimizeExposureTime` | This mode allows the lowest possible Exposure Time, but the minimum Trigger Delay at full resolution increases to 412 µs. |
| `AllowTriggerOverlap` | This mode allows Trigger Overlap, but will introduce a few microseconds of jitter into Trigger Delay.<br>The typical minimum trigger delay time at full resolution is 412 µs. |
| GigEVision: GenICam | `IMXTriggerTiming` |
| Type | Enumeration |
| VCD Property | `VCDID_Trigger\VCDElement_IMXTriggerTiming` |

### 5.4.8 Trigger Overlap

The *Trigger Overlap* feature gives information on when a new trigger pulse is accepted in the trigger -> exposure -> readout sequence.

| Parameter | Trigger Overlap |
|---|---|
| `Off` | The next trigger pulse is only accepted once the previous frame has been read out from the sensor |
| `Readout` | The next trigger pulse is accepted during readout as long as the remaining readout time is shorter than the exposure time |
| GigEVision: GenICam | `TriggerOverlap` |
| Type | Enumeration |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerOverlap` |

### 5.4.9 Trigger Delay

The *Trigger Delay* parameter specifies a time for which the camera waits between receiving a trigger signal and starting the exposure of an image. Simulated trigger pulses generated through the *Software Trigger* function are not delayed by this parameter.

| Parameter | Trigger Delay |
|---|---|
| Minimum | Dynamic, as low as 0.8 µs |
| Maximum | 1 s |
| GigEVision: GenICam | `TriggerDelay` |
| Type | Float |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerDelay` |

**Dynamic minimum:** The minimum value depends on other settings such as IMX Trigger Timing, *Width*, *PixelFormat*, and Trigger Polarity.

## 5.5 Trigger Timing Parameters

The 47G series camera series offers several options for dealing with bad trigger signals. By using the *Debounce Time*, *Denoise Time* and *Mask Time* parameters, the camera can be configured to ignore pulses on its TRIGGER_IN line under certain conditions.

Useful values for these parameters are application-specific. They depend on the expected trigger frequency, the exposure time and assumptions about the input signal quality.

The default values for all these parameters is 0 µs, assuming perfect signal quality.

### 5.5.1 Trigger Debounce Time

The *Trigger Debounce Time* parameter specifies the time for which the trigger input has to be low in order for the next trigger signal to be accepted.

| Parameter | Trigger Debounce Time |
|---|---|
| Minimum | 0 s |
| Maximum | 1 s |
| Default | 0 s |
| GigEVision: GenICam | `TriggerDebouncer` |
| Type | Float |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerDebouncer` |

### 5.5.2 Trigger Mask Time

The *Trigger Mask Time* parameter specifies the time for which trigger pulses are ignored after accepting a trigger signal.

| Parameter | Trigger Mask Time |
|---|---|
| Minimum | 0 s |
| Maximum | 1 s |
| Default | 0 s |
| GigEVision: GenICam | `TriggerMask` |
| Type | Float |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerMask` |

### 5.5.3 Trigger Noise Suppression Time

The *Trigger Noise Suppression Time* parameter specifies the time for which trigger input has to be active in order to be accepted as a trigger signal.

| Parameter | Trigger Noise Suppression Time |
|---|---|
| Minimum | 0 s |
| Maximum | 1 s |
| Default | 0 s |
| GigEVision: GenICam | `TriggerDenoise` |
| Type | Float |
| VCD Property | `VCDID_Trigger\VCDElement_TriggerDenoise` |

## 5.6 Digital I/O

47G series cameras have one digital input and two digital outputs. The digital input can be used as a Trigger input, and the current status can also examined directly.

The digital outputs can be configured as a Strobe output to signal the exact moment when the image sensor is sensitive to light so that external light sources can be synchronized to its operation cycle.

See I/O Connector for a description of the electric properties of the I/O connector.

### 5.6.1 Line Selector

The *Line Selector* feature selects the physical line (or pin) of the external device connector to configure.

| Parameter | Line Selector |
|---|---|
| `Line0` | Select Line 0 |
| `Line1` | Select Line 1 |
| `Line2` | Select Line 2 |
| GigEVision: GenICam | `LineSelector` |
| Type | Enumeration |

If a line is selected, all other Line* features will be applied to its associated device pin:

- Line Name
- Line Mode
- Line Inverter
- Line Status
- Line Status Invalidate
- Line Source
- Line Format

### 5.6.2 Line Name

*Line Name* returns the name of the line selected by Line Selector.

| Parameter | Line Name |
|---|---|
| GigEVision: GenICam | `LineName` |
| Type | String |

### 5.6.3 Line Mode

*Line Mode* is a read-only feature indicating the direction of the line selected by Line Selector.

| Parameter | Line Mode |
|---|---|
| `Input` | The selected line is an input. |
| `Output` | The selected line is an output. |
| GigEVision: GenICam | `LineMode` |
| Type | Enumeration |

### 5.6.4 Line Inverter

The *Line Inverter* feature can be used to invert the output signal of the line selected by Line Selector.

*Line Inverter* is only available if the Line Mode of the selected line is set to *Output*.

| Parameter | Line Inverter |
|---|---|
| `False` | The line signal is not inverted. |
| `True` | The line signal is inverted. |
| GigEVision: GenICam | `LineInverter` |
| Type | Boolean |

### 5.6.5 Line Status

The *Line Status* feature returns the current status of the line selected by Line Selector.

The status of the signal is taken after the signal was possible inverted by Line Inverter.

*Line Status* is only available if the Line Mode of the selected line is set to *Input*.

| Parameter | Line Status |
|---|---|
| `True` | The level of the line signal is High. |
| `False` | The level of the line signal is Low. |
| GigEVision: GenICam | `LineStatus` |
| Type | Boolean |

### 5.6.6 Line Status Invalidate

*Line Status Invalidate* invalidates the value of Line Status, causing it to be re-read from the device.

| Parameter | Line Status Invalidate |
|---|---|
| Execute | Invalidates the value of *LineStatus*. |
| GigEVision: GenICam | `LineStatusInvalidate` |
| Type | Command |

### 5.6.7 Line Source

*Line Source* selects which internal I/O source signal is output on the line selected by Line Selector.

The *Line Source* feature is only available if the Line Mode of the selected line is set to *Output*.

| Parameter | Line Source |
|---|---|
| `UserOutput0` | The line outputs the User Output Bit 0 state as defined by User Output Value. |
| `UserOutput1` | The line outputs the User Output Bit 1 state as defined by User Output Value. |
| `UserOutput2` | The line outputs the User Output Bit 2 state as defined by User Output Value. |
| `ExposureActive` | Device is doing exposure of a frame. |
| GigEVision: GenICam | `LineSource` |
| Type | Enumeration |

### 5.6.8 Line Format

*Line Format* indicates the electrical format of the selected line selected by Line Selector.

| Parameter | Line Format |
|---|---|
| `TTL` | The Line is currently accepting or sending TTL level signals. |
| `OptoCoupled` | The Line is opto-coupled. |
| `OpenDrain` | The Line is Open Drain. |
| GigEVision: GenICam | `LineFormat` |
| Type | Enumeration |

### 5.6.9 User Output Selector

*User Output Selector* selects which bit of the user output register will be set by User Output Value.

The bits in the user output register can be selected as source signal in Line Source.

| Parameter | User Output Selector |
|---|---|
| `UserOutput0` | Selects bit 0 of the user output register. |
| `UserOutput1` | Selects bit 1 of the user output register. |
| `UserOutput2` | Selects bit 2 of the user output register. |
| GigEVision: GenICam | `UserOutputSelector` |
| Type | Enumeration |

### 5.6.10 User Output Value

The *User Output Value* feature sets the value of the bit selected by User Output Selector.

The bits in the user output register can be selected as source signal in Line Source.

| Parameter | User Output Value |
|---|---|
| `False` | Set the bit to 0. |
| `True` | Set the bit to 1. |
| GigEVision: GenICam | `UserOutputValue` |
| Type | Boolean |

## 5.7 Strobe

The strobe function controls the automatic generation of output pulses on the STROBE_OUT pin which is synchronized to the image sensor's exposure time.

### 5.7.1 Strobe Enable

The *Strobe Enable* parameter enables the automatic generation of strobe pulses.

| Parameter | Strobe Enable |
|---|---|
| `On` | Strobe enabled |
| `True` | |
| `Off` | Strobe disabled |
| `False` | |
| GigEVision: GenICam | `StrobeEnable` |
| Type | Enumeration |
| VCD Property | `VCDID_Strobe\VCDElement_Value` |

### 5.7.2 Strobe Polarity

The *Strobe Polarity* parameter can be used to invert the strobe pulse output.

| Parameter | Strobe Polarity |
|---|---|
| `ActiveHigh` | The STROBE_OUT pin is logically high during the exposure time |
| `True` | |
| `ActiveLow` | The STROBE_OUT pin is logically low during the exposure time |
| `False` | |
| GigEVision: GenICam | `StrobePolarity` |
| Type | Enumeration |
| VCD Property | `VCDID_Strobe\VCDElement_StrobePolarity` |

### 5.7.3 Strobe Operation

The *Strobe Operation* parameter specifies how the length of the strobe pulses are controlled.

| Parameter | Strobe Operation |
|---|---|
| `Exposure` | Output pulse duration is equal to the current exposure time |
| `True` | |
| GigEVision: GenICam | `StrobeOperation` |
| Type | Enumeration |
| VCD Property | `VCDID_Strobe\VCDElement_StrobeMode` |

## 5.8 White Balance

The DFM 47GX900-MLP color camera can perform white balance by digital multiplication of the appropriate raw pixel values.

### 5.8.1 Auto White Balance

DFM 47GX900-MLP can perform *Auto White Balance*. It can be enabled continuously when changes in lighting conditions are expected. Under controlled lighting conditions, however, one-time white balance calibration is the recommended approach.

To perform one-time white balance calibration, place a reference white sheet in front of the camera and set `BalanceWhiteAuto` to *Once*. The camera will then find the correct white balance settings for the current light conditions and keep them active until instructed otherwise.

| Parameter | Auto White Balance |
|---|---|
| `Continuous` | Enable automatic white balance |
| `True` | |
| `Once` | Enable automatic white balance until the color channels have been balanced once |
| `One Push` | |
| `Off` | Disable automatic white balance |
| `False` | |
| GigEVision: GenICam | `BalanceWhiteAuto` |
| Type | Enumeration |
| VCD Property | `VCDID_WhiteBalance\VCDElement_Auto`<br>`VCDID_WhiteBalance\VCDElement_OnePush` |

### 5.8.2 White Balance Mode

The default auto white balance algorithm is *Gray World*. It is selected by setting the *White Balance Mode* accordingly. The *Gray World* algorithm operates under the assumption that the average color of a scene is near gray and adjusts the white balance coefficients so the average color of the resulting image is gray as well. In the case where this assumption is wrong (e.g. the camera is pointing at a blue carpet), the auto white balance algorithm will select coefficients that do not result in a visually accurate image.

To avoid the problems posed by the *Gray World* algorithm, the *Temperature* mode can be used. This mode operates by trying to determine the color temperature of the light source.

| Parameter | White Balance Mode |
|---|---|
| `Gray World` | Use the *Gray World* model |
| `Temperature` | Use the *Temperature* model |
| GigEVision: GenICam | `BalanceWhiteMode` |
| Type | Enumeration |
| VCD Property | `VCDID_WhiteBalance`<br>`\VCDElement_WhiteBalanceMode` |

To assist the automatic white balance temperature detection, *White Balance Auto Preset* can be used to limit the range of valid color temperatures.

| Parameter | White Balance Auto Preset |
|---|---|
| `Any` | Allow any color temperature |
| `Auto Warm White` | Limit color temperature to approximately 2500-4000 K |
| `Auto Cool White` | Limit color temperature to approximately 4000-5800 K |
| `Auto Daylight` | Limit color temperature to approximately 5700-7500 K |
| `Auto Deep Shade` | Limit color temperature to approximately 7500-10000 K |
| GigEVision: GenICam | `BalanceWhiteAutoPreset` |
| Type | Enumeration |
| VCD Property | `VCDID_WhiteBalance \VCDElement_WhiteBalanceAutoPreset` |

### 5.8.3 Manual White Balance

If *Auto White Balance* is not enabled, the amplifiers for the red, green and blue channels can be controlled manually.

| Parameter | White Balance Red |
|---|---|
| Minimum | 0 |
| Maximum | 3.984375 |
| Default | auto |
| GigEVision: GenICam | `BalanceRatio[BalanceRatioSelector=Red]` |
| Type | Float |
| VCD Property | `VCDID_WhiteBalance \VCDElement_WhiteBalanceRed` |

| Parameter | White Balance Green |
|---|---|
| Minimum | 0 |
| Maximum | 3.984375 |
| Default | auto |
| GigEVision: GenICam | `BalanceRatio[BalanceRatioSelector=Green]` |
| Type | Float |
| VCD Property | `VCDID_WhiteBalance \VCDElement_WhiteBalanceGreen` |

| Parameter | White Balance Blue |
|---|---|
| Minimum | 0 |
| Maximum | 3.984375 |
| Default | auto |
| GigEVision: GenICam | `BalanceRatio[BalanceRatioSelector=Blue]` |
| VCD Property | `VCDID_WhiteBalance \VCDElement_WhiteBalanceBlue` |

If *White Balance Mode* is set to *Temperature*, the color temperature can be set directly. Additionally, a light source can be selected from a list of presets.

| Parameter | White Balance Temperature |
|---|---|
| Minimum | 2500 K |
| Maximum | 10000 K |
| Default | auto |
| Increment | 100 |
| GigEVision: GenICam | `BalanceWhiteTemperature` |
| Type | Integer |
| VCD Property | `VCDID_WhiteBalance \VCDElement_WhiteBalanceTemperature` |

| Parameter | White Balance Temperature Preset |
|---|---|
| Sodium-Vapor Lamp | 2500 K |
| Warm Light | 2700 K |
| Halogen Light | 3000 K |
| Neutral White | 3500 K |
| Cool White Fluorescent Light | 4000 K |
| Cool White/Daylight (CFL) | 5000 K |
| Daylight | 6000 K |
| Cool White LED | 7500 K |
| Blue Sky | 10000 K |
| GigEVision: GenICam | BalanceWhiteTemperaturePreset |
| Type | Enumeration |
| VCD Property | VCDID_WhiteBalance \VCDElement_TemperaturePreset |

## 5.9 Image Processing

### 5.9.1 Gamma

The 47G series cameras can perform gamma correction on the image data. If the pixel format of a color camera is set a YUV mode, gamma correction is applied to the Y channel. In bayer raw and RGB24 modes, all color channels are subject to the same gamma correction.

| Parameter | Gamma |
|---|---|
| Minimum | 0.3 |
| Maximum | 1.2 |
| Default | 0.55 |
| GigEVision: GenICam | Gamma |
| Type | Float |
| VCD Property | VCDID_Gamma\VCDElement_Value |

### 5.9.2 Lookup Table

The 47G series allows the user to specify a lookup table (LUT) that can transform pixel values. This lookup table can be used to perform tasks like histogram equalization inside the camera.

If the pixel format of a color camera is set to one of the YUV modes, the lookup table is applied only to the Y channel. In bayer raw and RGB24 modes, all color channels are subject to the same lookup operation.

The lookup table has 257 16-bit entries. The upper 8 bits of the image data are used to select the appropriate entry in the lookup table, while the remaining bits are used to interpolate between the selected entry and one of its neighbors.

The *LUT Enable* feature enables the lookup table.

| **Parameter** | **LUT Enable** |
|---|---|
| `True` | Enable lookup table operation |
| `False` | Disable lookup table operation |
| GigEVision: GenICam | `LUTEnable` |
| Type | Boolean |

The lookup table values can be programmed using the *LUT Index* and *LUT Value* features.

| **Parameter** | **LUT Index** |
|---|---|
| Minimum | 0 |
| Maximum | 256 |
| GigEVision: GenICam | `LUTIndex` |
| Type | Integer |

| **Parameter** | **LUT Value** |
|---|---|
| Minimum | 0 |
| Maximum | 65535 |
| GigEVision: GenICam | `LUTValue` |
| Type | Integer |

The whole lookup table can be quickly reprogrammed through the *LUT Value All* feature.

| Parameter | LUT Value All |
|---|---|
| `Value` | 1028 bytes of memory containing 257 LUT entries. Each 16-bit LUT entry is aligned on a 32-bit boundary |
| GigEVision: GenICam | `LUTValueAll` |
| Type | Register |

**Important:** Since gamma correction is implemented using the same hardware as the lookup table, gamma correction is disabled when the lookup table is enabled.

## 5.10 Color Processing

The DFM 47GX900-MLP color camera allows color adjustments with the *Hue* and *Saturation* parameters. These adjustments work with all output pixel formats.

**Important:** When a bayer raw pixel format is selected, *Hue* and *Saturation* still work. However, when at least one of them is enabled (set to non-default value), the resulting image is slightly blurred because of the way these parameters are implemented internally.

### 5.10.1 Hue

The *Hue* parameter controls the color angle of the output image.

| Parameter | Hue |
|---|---|
| Minimum | -180 |
| Maximum | 180 |
| Default | 0 |
| GigEVision: GenICam | `Hue` |
| Type | Float |
| VCD Property | `VCDID_Hue\VCDElement_Value` |

### 5.10.2 Saturation

The *Saturation* parameter controls the color saturation of the output image.

| Parameter | Saturation |
|---|---|
| Minimum | 0 |
| Maximum | 255 |
| Default | 64 |
| GigEVision: GenICam | `Saturation` |
| Type | Float |
| VCD Property | `VCDID_Saturation\VCDElement_Value` |

### 5.10.3 Color Correction Matrix

As seen in the Spectral Characteristics diagram, the frequencies at which the red, green and blue pixels respond to incoming light overlap. This causes the colors of objects in captured images to differ from the colors of the same objects seen in reality.

A RGB to RGB color correction matrix can be used to change the pixel data output of the camera so that the colors in captured image are a closer match to what is expected by human observers.

| Parameter | Enable Color Correction Matrix |
|---|---|
| `True` | Enable Color Correction Matrix |
| `False` | Disable Color Correction Matrix |
| GigEVision: GenICam | `ColorTransformationEnable` |
| Type | Boolean |
| VCD Property | `VCDID_ColorMatrix\VCDElement_Value` |

By default, the values of the color matrix coefficients are initialized with values optimized for skin tones and other everyday objects.

If necessary, the matrix coefficients can be controlled manually.

When using *GenICam*, select a coefficient using `ColorTransformationValueSelector` and then change it via the `ColorTransformationValue`.

When using *IC Imaging Control*, the matrix coefficients are controlled via the `VCDElement_ColorMatrix[RGB][RGB]` elements of `VCDID_ColorMatrix`.

| Parameter | Red contribution to the red pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain00]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixRR` |

| Parameter | Green contribution to the red pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain01]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixRG` |

| Parameter | Blue contribution to the red pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain02]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixRB` |

| Parameter | Red contribution to the green pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain10]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixGR` |

| Parameter | Green contribution to the green pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain11]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixGG` |

| Parameter | Blue contribution to the green pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain12]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixGB` |

| Parameter | Red contribution to the blue pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain20]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixBR` |

| Parameter | Green contribution to the blue pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain21]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixBG` |

| Parameter | Blue contribution to the blue pixel |
|---|---|
| Minimum | -3.9375 |
| Maximum | 3.9375 |
| GigEVision: GenICam | `ColorTransformationValue [ColorTransformationValueSelector=Gain22]` |
| Type | Float |
| VCD Property | `VCDID_ColorMatrix \VCDElement_ColorMatrixBB` |

## 5.11 Region of Interest for Auto Functions

The 47G series cameras allow for the setting of a region of interest that is used while controlling the Auto Exposure, Auto Gain and Auto White Balance functions.

### 5.11.1 Auto Functions ROI Enable

The *Auto Functions ROI Enable* parameter allows for the use of a region of interest for auto functions.

| Parameter | Auto Functions ROI Enable |
|---|---|
| `True` | Auto functions only take a specified part of the image into account |
| `False` | Auto functions look at the whole image |
| GigEVision: GenICam | `AutoFunctionsROIEnable` |
| Type | Boolean |
| VCD Property | `VCDID_AutoRoi\VCDElement_Value` |

### 5.11.2 Auto Functions ROI Preset

The *Auto Functions ROI Preset* parameter lets the user select from a list of ROI presets.

| Parameter | Auto Functions ROI Preset |
|---|---|
| `Full Sensor` | Select the full output image |
| `Center 50%` | Select a rectangular area in the center of the output image |
| `Center 25%` | Select a small rectangular area in the center of the output image |
| `Bottom Half` | Select the bottom half of the output image |
| `Top Half` | Select the top half of the output image |
| `Custom Rectangle` | Select a custom rectangle for fine-grained control |
| GigEVision: GenICam | `AutoFunctionsROIPreset` |
| Type | Enumeration |
| VCD Property | `VCDID_AutoRoi\VCDElement_AutoRoiPreset` |

### 5.11.3 Auto Functions ROI Custom Rectangle

When *Auto Functions ROI Preset* is set to *Custom Rectangle*, the user can manually set the position and size of the region of interest. The coordinates are relative to the origin of the current video format as specified by Partial Scan Offset.

| Parameter | Auto Functions ROI Left |
|---|---|
| Minimum | 0 |
| Maximum | 2032 |
| Increment | 2 |
| GigEVision: GenICam | `AutoFunctionsROILeft` |
| Type | Integer |
| VCD Property | `VCDID_AutoRoi \VCDElement_AutoRoiLeftRelative` |

| Parameter | Auto Functions ROI Top |
|---|---|
| Minimum | 0 |
| Maximum | 1520 |
| Increment | 2 |
| GigEVision: GenICam | `AutoFunctionsROITop` |
| Type | Integer |
| VCD Property | `VCDID_AutoRoi \VCDElement_AutoRoiTopRelative` |

| Parameter | Auto Functions ROI Width |
|---|---|
| Minimum | 16 |
| Maximum | 2048 |
| Increment | 2 |
| GigEVision: GenICam | `AutoFunctionsROIWidth` |
| Type | Integer |
| VCD Property | `VCDID_AutoRoi \VCDElement_AutoRoiWidthRelative` |

| Parameter | Auto Functions ROI Height |
|---|---|
| Minimum | 16 |
| Maximum | 1536 |
| Increment | 2 |
| GigEVision: GenICam | `AutoFunctionsROIHeight` |
| Type | Integer |
| VCD Property | `VCDID_AutoRoi \VCDElement_AutoRoiHeightRelative` |

## 5.12 User Sets

The 47G series cameras can store their complete configuration into built-in non-volatile memory. The camera configuration can be saved into and restored from one of two available memory slots upon user's request. Additionally, the camera can be configured to load one of the user's camera configurations at startup.

### 5.12.1 User Set Selector

The *User Set Selector* parameter selects the memory slot on which consecutive load and save commands are executed.

| Parameter | User Set Selector |
|---|---|
| `UserSet1` | First of the two memory slots for user configurations |
| `UserSet2` | Second of the two memory slots for user configurations |
| `Default` | A special memory slot that always contains the camera's factory default configuration. It cannot be overwritten |
| GigEVision: GenICam | `UserSetSelector` |
| Type | Enumeration |

### 5.12.2 Load User Set

By using the *Load User Set* command, the user can restore the camera configuration from the memory slot selected by *User Set Selector*.

| Parameter | Load User Set |
|---|---|
| Execute | Restore the camera configuration from the memory slot specified by *User Set Selector* |
| GigEVision: GenICam | `UserSetLoad` |
| Type | Command |

### 5.12.3 Save User Set

By using the *Save User Set* command, the user can save the camera configuration into the memory slot selected by *User Set Selector*.

| Parameter | Save User Set |
|---|---|
| Execute | Save the current camera configuration into the memory slot specified by *User Set Selector* |
| GigEVision: GenICam | `UserSetSave` |
| Type | Command |

### 5.12.4 Default User Set

The *Default User Set* parameter controls the camera's startup behavior. It selects which memory slot is used for device initialization when the camera is powered up.

| Parameter | Default User Set |
|---|---|
| `UserSet1` | Load configuration from the first memory slot |
| `UserSet2` | Load configuration from the second memory slot |
| `Default` | Load factory default configuration |
| GigEVision: GenICam | `UserSetDefault` |
| Type | Enumeration |

## 5.13 Timestamp Control

47G series cameras run an internal timestamp clock. The clock starts at zero when the camera is powered up.

The source of the timestamp clock has a frequency tolerance of ±20 ppm.

All image buffers sent by the camera have a timestamp attached to them.

Notification for asynchronous Events also contain a timestamp value.

The time stamp can be synchronized to other network devices via Precision Time Protocol.

### 5.13.1 Timestamp Latch

The *Timestamp Latch* command copies the current value of the camera's timestamp counter into the Timestamp Latch Value so that it can be read later.

| Parameter | Timestamp Latch |
|---|---|
| Execute | Copies the current value of the camera's timestamp into Timestamp Latch Value. |
| GigEVision: GenICam | `TimestampLatch` |
| Type | Command |

### 5.13.2 Timestamp Latch Value

The *Timestamp Latch Value* feature contains the value of camera's timestamp counter at the time of the previous execution of the Timestamp Latch command.

A human-readable interpretation of the value can be read from Timestamp Latch String.

| Parameter | Timestamp Latch Value |
|---|---|
| GigEVision: GenICam | `TimestampLatchValue` |
| Type | Integer |
| Unit | µs |

### 5.13.3 Timestamp Latch String

The *Timestamp Latch String* feature contains a human-readable interpretation of the value of Timestamp Latch Value.

| Parameter | Timestamp Latch String |
|---|---|
| GigEVision: GenICam | `TimestampLatchString` |
| Type | String |

### 5.13.4 Timestamp Reset

The *Timestamp Reset* command resets the camera's timestamp counter to start from zero.

| Parameter | Timestamp Reset |
|---|---|
| Execute | Resets the camera's timestamp counter. |
| GigEVision: GenICam | `TimestampReset` |
| Type | Command |

**Important:** Resetting the timestamp counter is not possible if PTP Enable is enabled.

## 5.14 Precision Time Protocol

The 47G series can synchronize its internal clock to other devices via *Precision Time Protocol* (PTP, IEEE 1588).

When multiple PTP devices are connected through the same network, they will synchronize their clocks. This allows the image timestamps to remain comparable between cameras.

A camera running with PTP enabled can also schedule actions to be executed at a specified time point in the future through *Scheduled Actions*.

PTP is enabled by default.

### 5.14.1 PTP Enable

The *PTP Enable* parameter enables or disables the camera's usage of the *Precision Time Protocol*.

| Parameter | PTP Enable |
|---|---|
| `True` | Use Precision Time Protocol |
| `False` | PTP disabled |
| GigEVision: GenICam | `PtpEnable` |
| Type | Boolean |

### 5.14.2 PTP Status

*PTP Status* is a read-only feature that reports the current status of the *Precision Time Protocol* algorithm.

| Parameter | PTP Status |
|---|---|
| `Master` | The camera acts as PTP master clock. |
| `Slave` | The camera synchronizes its time stamp to a PTP master clock. |
| `Uncalibrated` | PTP was recently enabled, or a recent change in the network configuration initiated a resynchronization. |
| `Disabled` | PTP is disabled. |
| GigEVision: GenICam | `PtpStatus` |
| Type | Enumeration |

## 5.15 Action Scheduler

When *Precision Time Protocol* is enabled, the action scheduler can be used to plan and run actions at some point in the future.

The only action the camera can perform right now is triggering images.

Action triggers are only accepted when Trigger Source is set to `Action0` or `Any`.

### 5.15.1 Action Selector

The *Action Selector* feature selects the action that the Action Scheduler Time, Action Scheduler Interval, Action Scheduler Commit and Action Scheduler Cancel features act on.

| Parameter | Action Selector |
|---|---|
| `0` | `Action0`, which is hard-wired to trigger images |
| GigEVision: GenICam | `ActionSelector` |
| Type | Enumeration |

### 5.15.2 Action Scheduler Time

The *Action Scheduler Time* is the time for when an action is scheduled, specified as a PTP timestamp in nanosecond units.

Scheduling the action to a time in the past will cause the action to be performed immediately.

Scheduling a repeated action to start in the past will ignore missed actions and continue as if it had been scheduled earlier.

| Parameter | Action Scheduler Time |
|---|---|
| GigEVision: GenICam | `ActionSchedulerTime` |
| Type | Integer |

### 5.15.3 Action Scheduler Interval

The *Action Scheduler Interval* feature allows instructing the action scheduler to repeat the scheduled action in a fixed interval, indefinitely.

If the interval is set to 0, the action is performed once and not repeated.

The interval is specified in microseconds.

| Parameter | Action Scheduler Interval |
|---|---|
| Minimum | 0 |
| Maximum | `4294967295 us` |
| GigEVision: GenICam | `ActionSchedulerInterval` |
| Type | Integer |

### 5.15.4 Action Scheduler Commit

Executing the *Action Scheduler Commit* command commits an action to the queue of scheduled actions.

The action is planned to be performed at the time specified by Action Scheduler Time.

If Action Scheduler Interval has a value other than 0, the action will be repeated with the specified interval indefinitely, or until the Action Scheduler Cancel command is executed.

There is a finite number of queue slots available, trying to queue too many actions will fail. The maximum length of the queue is reported by the `ActionQueueSize` feature.

| Parameter | Action Scheduler Commit |
|---|---|
| Execute | Schedule the action specified by Action Selector to be performed at a specified time in the future. |
| GigEVision: GenICam | `ActionSchedulerCommit` |
| Type | Command |

### 5.15.5 Action Scheduler Cancel

Executing the *Action Scheduler Cancel* command cancels all scheduled actions of the action type selected by Action Selector.

| Parameter | Action Scheduler Cancel |
|---|---|
| Execute | Cancel all scheduled actions of the type specified by Action Selector. |
| GigEVision: GenICam | `ActionSchedulerCancel` |
| Type | Command |

## 5.16 Events

47G series devices can send asynchronous event packets, notifying the application about certain events during device operation.

To enable a specific event, select the event to enable in Event Selector and then set Event Notification to *On*.

The way to receive the event notifications depends on the used camera SDK. In most cases, a notification handler has to be registered on the feature named after the event, e.g. *EventTest* or *EventExposureEnd*. Additional event data can often be queried via additional features with a derived name, e.g. *EventTestTimestamp* or *EventExposureEndFrameID*.

### 5.16.1 Event Selector

The *Event Selector* parameter selects an event to configure.

| Parameter | Event Selector |
|---|---|
| `ExposureEnd` | Selects the ExposureEnd event |
| `FrameTriggerMissed` | Selects the FrameTriggerMissed event |
| `Line1FallingEdge` | Selects the Line1FallingEdge event |
| `Line1RisingEdge` | Selects the Line1RisingEdge event |
| `Test` | Selects the Test event |
| GigEVision: GenICam | `EventSelector` |
| Type | Enumeration |

### 5.16.2 Event Notification

The *Event Notification* parameter enables or disables the event selected by Event Selector.

| Parameter | Event Notification |
|---|---|
| `On` | Enables notifications for the selected event |
| `Off` | Disables notifications for the selected event |
| GigEVision: GenICam | `EventNotification` |
| Type | Enumeration |

### 5.16.3 Event Exposure End

The *Event Exposure End* feature is notified when the camera completes the exposure of an image.

It can be useful to synchronize camera exposure with real-world events, e.g. moving parts in front of the camera.

The value of this feature has no meaning and should not be read.

| Parameter | Event Exposure End |
|---|---|
| GigEVision: GenICam | `EventExposureEnd` |
| Type | Integer |

The *Event Exposure End Timestamp* parameter identifies the point in time when the current exposure end event occurred.

Please refer to the documentation of your camera SDK on how to safely read the timestamp associated with the current event when a notification is received.

| Parameter | Event Exposure End Timestamp |
|---|---|
| GigEVision: GenICam | `EventExposureEndTimestamp` |
| Type | Integer |

The *Event Exposure End Frame ID* parameter the frame number of the frame whose exposure was completed.

Please refer to the documentation of your camera SDK on how to safely read the timestamp associated with the current event when a notification is received.

| Parameter | Event Exposure End Frame ID |
|---|---|
| GigEVision: GenICam | `EventExposureEndFrameID` |
| Type | Integer |

### 5.16.4 Event Frame Trigger Missed

The *Event Frame Trigger Missed* feature is notified when the camera receives a trigger pulse, but the sensor was not in a state where a new exposure could be started.

The value of this feature has no meaning and should not be read.

| Parameter | Event Frame Trigger Missed |
|---|---|
| GigEVision: GenICam | `EventFrameTriggerMissed` |
| Type | Integer |

The *Event Frame Trigger Missed Timestamp* parameter identifies the point in time when the current frame trigger missed event occurred.

Please refer to the documentation of your camera SDK on how to safely read the timestamp associated with the current event when a notification is received.

| **Parameter** | **Event Frame Trigger Missed Timestamp** |
|---|---|
| GigEVision: GenICam | `EventFrameTriggerMissedTimestamp` |
| Type | Integer |

### 5.16.5 Event Line1 Falling Edge

The *Event Line1 Falling Edge* feature is notified when the input on Line1 changes from high to low.

The value of this feature has no meaning and should not be read.

| **Parameter** | **Event Line1 Falling Edge** |
|---|---|
| GigEVision: GenICam | `EventLine1FallingEdge` |
| Type | Integer |

The *Event Line1 Falling Edge Timestamp* parameter identifies the point in time when the current line1 falling edge event occurred.

Please refer to the documentation of your camera SDK on how to safely read the timestamp associated with the current event when a notification is received.

| **Parameter** | **Event Line1 Falling Edge Timestamp** |
|---|---|
| GigEVision: GenICam | `EventLine1FallingEdgeTimestamp` |
| Type | Integer |

### 5.16.6 Event Line1 Rising Edge

The *Event Line1 Rising Edge* feature is notified when the input on Line1 changes from high to low.

The value of this feature has no meaning and should not be read.

| **Parameter** | **Event Line1 Rising Edge** |
|---|---|
| GigEVision: GenICam | `EventLine1RisingEdge` |
| Type | Integer |

The *Event Line1 Rising Edge Timestamp* parameter identifies the point in time when the current line1 rising edge event occurred.

Please refer to the documentation of your camera SDK on how to safely read the timestamp associated with the current event when a notification is received.

| **Parameter** | **Event Line1 Rising Edge Timestamp** |
|---|---|
| GigEVision: GenICam | `EventLine1RisingEdgeTimestamp` |
| Type | Integer |

### 5.16.7 Event Test

The *Event Test* feature is notified when a test event occurs.

To generate a test event, use the *Test Event Generate* command.

The value of this feature has no meaning and should not be read.

| **Parameter** | **Event Test** |
|---|---|
| GigEVision: GenICam | `EventTest` |
| Type | Integer |

The *Event Test Timestamp* parameter identifies the point in time when the current test event occurred.

Please refer to the documentation of your camera SDK on how to safely read the timestamp associated with the current event when a notification is received.

| **Parameter** | **Event Test Timestamp** |
|---|---|
| GigEVision: GenICam | `EventTestTimestamp` |
| Type | Integer |

The *Test Event Generate* command can be used to generate a test event.

| **Parameter** | **Test Event Generate** |
|---|---|
| Execute | Generate one test event |
| GigEVision: GenICam | `TestEventGenerate` |
| Type | Command |

## 5.17 Chunk Data

Some 47G series devices can send chunk data alongside the capture images, containing additional information about the captured frames.

To enable the transmission of chunk data, Chunk Mode Active has to be enabled.

To enable specific information to be included as part of the chunk data, select the chunk to enable in Chunk Selector and then enable Chunk Enable. Please note that certain chunkdata parts can be stuck to always-on.

The way to receive the chunk depends on the used camera SDK. In most cases, a received frame has to be attached to the device's property interface. After that, the chunk's values are readable like other device properties.

### 5.17.1 Chunk Mode Active

The *Chunk Mode Active* parameter enables the transmission of chunk data alongside the image data.

Please note that enabling chunk data changes the transmitted image data format. The chunk format is not necessarily supported by all GigEVision drivers.

| **Parameter** | **Chunk Mode Active** |
|---|---|
| `True` | Enable chunk data transmission |
| `False` | Disable chunk data transmission |
| GigEVision: GenICam | `ChunkModeActive` |
| Type | Boolean |

### 5.17.2 Chunk Selector

The *Chunk Selector* parameter selects a chunkdata item to configure.

| **Parameter** | **Chunk Selector** |
|---|---|
| `ExposureTime` | Selects ExposureTime chunk |
| `Gain` | Selects Gain chunk |
| GigEVision: GenICam | `ChunkSelector` |
| Type | Enumeration |

### 5.17.3 Chunk Enable

The *Chunk Enable* parameter enables or disables the chunkdata item selected by *Chunk Selector*.

| Parameter | Chunk Enable |
|---|---|
| `True` | Enables the selected chunkdata part |
| `False` | Disables the selected chunkdata part |
| GigEVision: GenICam | `ChunkEnable` |
| Type | Boolean |

### 5.17.4 Chunk Exposure Time

The *Chunk Exposure Time* parameter contains the exposure time used when capturing the image.

| Parameter | Chunk Exposure Time |
|---|---|
| Minimum | 15 µs |
| Maximum | 1 s |
| GigEVision: GenICam | `ChunkExposureTime` |
| Type | Float |

### 5.17.5 Chunk Gain

The *Chunk Gain* parameter contains the gain value used when capturing the image.

| Parameter | Chunk Gain |
|---|---|
| Minimum | 0 dB |
| Maximum | 48 dB |
| GigEVision: GenICam | `ChunkGain` |
| Type | Float |

## 5.18 Bandwidth Control

When using multiple 47G series devices connected to the same network adapter, the data rate has to be considered to allow consistent operation.

To limit the bandwith of a camera, set Device Link Throughput Limit Mode to `On`, and set Device Link Throughput Limit to the desired maximum data rate.

### 5.18.1 Device Link Throughput Limit Mode

The *Device Link Throughput Limit Mode* parameter enables or disables the effect of Device Link Throughput Limit.

If set, the camera's internal frame rate is automatically reduced to fit into the available bandwith.

An inter-packet delay is added between streaming packets to ensure the network bandwidth does not exceed the limit.

| Parameter | Device Link Throughput Limit Mode |
|---|---|
| `Off` | The bandwith the device uses for streaming is unlimited. |
| `On` | The maximum bandwidth the device uses for streaming is defined by Device Link Throughput Limit. |
| GigEVision: GenICam | `DeviceLinkThroughputLimitMode` |
| Type | Enumeration |

### 5.18.2 Device Link Throughput Limit

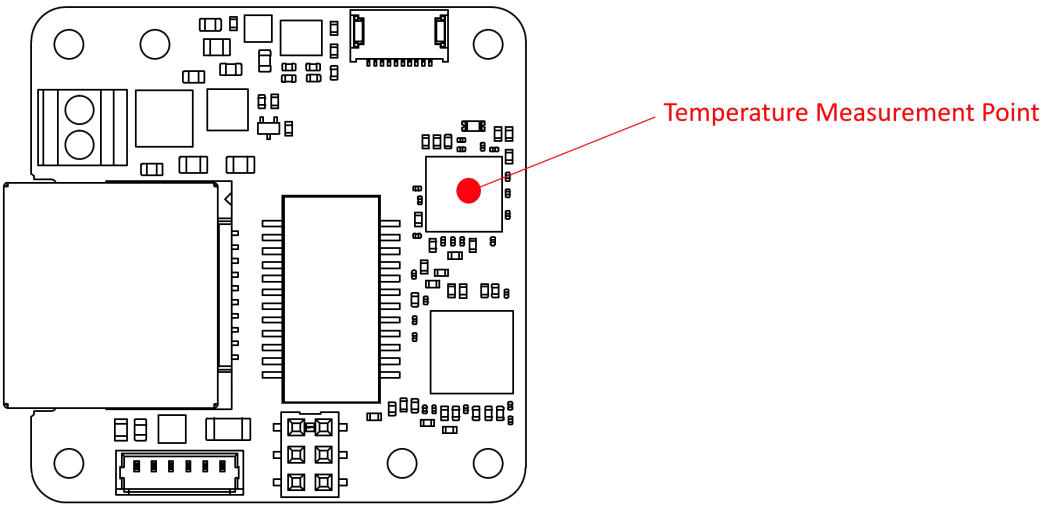
The *Device Link Throughput Limit* parameter defines the maximum bandwidth used by the device to stream data.

Device Link Throughput Limit Mode must be set to `On` for *Device Link Throughput Limit* to have an effect.

| Parameter | Device Link Throughput Limit |
|---|---|
| Minimum | 1000000 Mbps |
| Maximum | 125000000 Mbps |
| GigEVision: GenICam | `DeviceLinkThroughputLimit` |
| Type | Integer |

# 6 Temperature Measurement Point

Device temperature in operating state is measured on the back side of the circuit board of DFM 47GX900-MLP:



# 7 Status LEDs

## 7.1 On-Board LED

There is a single orange LED on the back side of the sensor circuit board.

| LED Status | Description |
|---|---|
| Orange | Startup or acquisition active |
| Off | Idle |

## 7.2 LEDs on the RJ45 Connector

The LEDs on the RJ45 connector show the network connection status and network activity.

| LED | Description |
|---|---|
| Green | On when connected to Gigabit Ethernet |
| Orange | Network activity indicator |

# 8 Revision History

| Date | Version | Description |
|---|---|---|
| 2025/2/14 | | Initial Release of this document |

# DFM 47GX900-MLP Technical Reference Manual

All product and company names in this document may be trademarks and tradenames of their respective owners and are hereby acknowledged.

The Imaging Source Europe GmbH cannot and does not take any responsibility or liability for any information contained in this document. The source code presented in this document is exclusively used for didactic purposes. The Imaging Source does not assume any kind of warranty expressed or implied, resulting from the use of the content of this document or the source code.

The Imaging Source Company reserves the right to make changes in specifications, function or design at any time and without prior notice.

Last update: April 2025
© 2025 The Imaging Source
All rights reserved. Reprint, also in parts, only allowed with permission of The Imaging Source Europe GmbH.

All weights and dimensions are approximate. Unless otherwise specified, the lenses shown in the context of cameras are not shipped with these cameras.

**Headquarters:**

*The Imaging Source Europe GmbH*
*Überseetor 18, D-28217 Bremen, Germany*
*Phone: +49 421 33591-0*

**North & South America:**

*The Imaging Source, LLC*
*4600 Park Road, Suite 470, Charlotte, NC 28209, United States*
*Phone: +1 877-462-4772*

**Asia Pacific:**

*The Imaging Source Asia Co., Ltd.*
*3F., No. 43-7/8, Zhongxing Road*
*Xizhi District, New Taipei City 221012, Taiwan*
*Phone: +886 2-2792-3153*

**www.theimagingsource.com**