bertin technologies

Ref.: 22459-800-DU003-G

CamSight HD

Interface Control Document CamSight High Definition (60 Hz configuration)

Approved by	Julien RANNOU	Camsight Activity Manager	Julien RANNOU Signature numérique de Julien RANNOU Date : 2026.04.02 10:26:05 +02'00'
Checked by	Emmanuel SOULIE	Camsight Technical Manager	Signature numérique de SOULIE Date : 2026.04.02 10:07:42 +02'00'
Written by	Batuhan KOSEOGLU	Optronic Field Application Engineer	Signature numérique de Batuhan KOSEOGLU Date : 2026.04.01 17:02:31 +02'00'
Written by	Antoine LANGUILLE	Software Embedded Engineer	LANGUILLE Antoine Signature numérique de LANGUILLE Antoine Date : 2026.04.02 09:09:19 +02'00'
Written by	Guillaume PHAM	Software Embedded Engineer	PHAM Guillaume Signature numérique de PHAM Guillaume Date : 2026.04.01 21:10:14 +02'00'
	Name	Title	Visa

bertin technologies
SAS (French Simplified Joint Stock Company) with registered capital of 18 000 000 Euros ▰ 422 511 204 RCS VERSAILLES ▰
French APE business sector code 7112B
Head Office: Parc d'Activités du Pas du Lac ▰ 10 bis, avenue Ampère Montigny-le-Bretonneux (78) ▰ France ▰ +33 (0)1 39 30 60 00
Postal Address: BP 284 ▰ 78053 Saint-Quentin-en-Yvelines Cedex ▰ FRANCE
https://www.bertin-technologies.com

LIST OF REVISIONS

Rev.	Date	**Modifications** (main reasons, sections and pages affected)	Writer / Checker
A	09/01/2025	First issue	JRa, MVi, Dor / DDi
B	28/02/2025	Rework the I2C table	RRi / GLa
C	23/05/2025	Add Trigger_mode §, feature available in FW versions ≥ 4.6.0.0. I2C interface: Add trigger_mode I2C register, update page layout, add missing reg description. Update J5 electrical information	SGe / GLa
D	23/06/2025	Update J6 / UART connector pinout after PCB update (version B): ⇨ ! MIPI interface board (022459-313-PCB) version A (first 11 cameras) and B have different pinout! Add missing information about I2C integrated pull-up	SGe / GLa
E	15/01/2025	FW evolution for MIPI 60Hz: - i2c bus address set at address 0x30 - Camera Serial Number extended to 4bytes for i2c Mipi messages Feature available in FW versions ≥ 4.7.2.0	GPh / ESo
F	20/02/2026	Update with the addition of new interfaces and lenses: - Description of the core camera - HD-SDI - GigE Vision	BKo
G	16/03/2026	Frequency-based (25 Hz or 60 Hz) separations of ICDs	BKo

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

This Interface Control Document (ICD) provides detailed information about the CamSight High Definition (HD) 60 Hz camera interfaces. This document’s purpose is to provide guidance for integrating this camera into other video equipment and systems, by describing the camera's powering requirements, interfaces, and connectors pinouts.

This document is intended to be used by engineers and technicians in charge of integrating the camera into their systems. It assumes a basic understanding of video signal processing, electrical engineering, and communication protocols. This document should be used as a general guide, and any specific requirements or questions should be addressed to Bertin Technologies directly.

1.1. Documentation

AD001	Compagnon software user manual	Upcoming Document
AD002	Updater software user manual	Upcoming Document

1.2. Acronyms

BP	Bad Pixel
CLAE	Contrast-Limited Adaptive Histogram Equalization
CLHE	Contrast-Limited Histogram Equalization
FoV	Field of View
HD	High Definition
MMCX	Micro-Miniature Coaxial
NUC	Non-Uniformity Correction
SDI	Serial Digital Interface
SWaP	Size, Weight and Power
UART	Universal Asynchronous Receiver Transmitter
VFE	Video Front End
WIP	Work In Progress

2. OVERVIEW

CamSight HD is an Original Equipment Manufacturer (OEM) digital core camera developed by Bertin Photonics. It has been designed to meet user's requirements of Low SWaP cameras offering easy integration within optronic systems while providing thermal infrared.

Figure 1: Picture of CamSight HD OEM version

The CamSight HD camera is based around a thermal infrared sensor detailed in Table 1.

Based on patented Bertin Photonics' shutterless, it allows for low latency and freeze-free real time image delivery in thermal infrared, making it suitable for a wide range of applications.

Sensor/Camera	Thermal IR
Manufacturer	LYNRED
Reference	ATTO1280-02
Spectrum	LWIR
Resolution	1280x1024
Pixel pitch	12 microns

Table 1 : CamSight HD sensor characteristics

CamSight HD 60 Hz is available in 7 different assemblies, each corresponding to a specific Field of View and SWaP requirement, mechanical blueprints and characteristics are given for each module.

The CamSight HD OEM version is composed of three main parts:

- At the center, core camera which contains the sensor and the fundamentals printed circuit boards. This component gives you access to the high sensitivity option.
- The lens, located just in front of the core camera. These lenses are specifically designed for the LWIR infrared range. They are integrated and calibrated using a patented method developed by Bertin Technologies, making them highly effective.
- An interface board, located on the back of the camera. They can provide additional connectivity options and allow easier integration with existing systems.

All these components are described in detail in the following sections.

3. MECHANICAL INTERFACE

Mechanical step files are available, please contact Bertin Technologies if needed.

3.1. Core camera

3.1.1. Core

The core camera weighs 115 g.

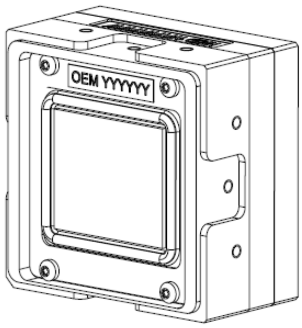

Figure 2: Drawing of the CamSight HD core

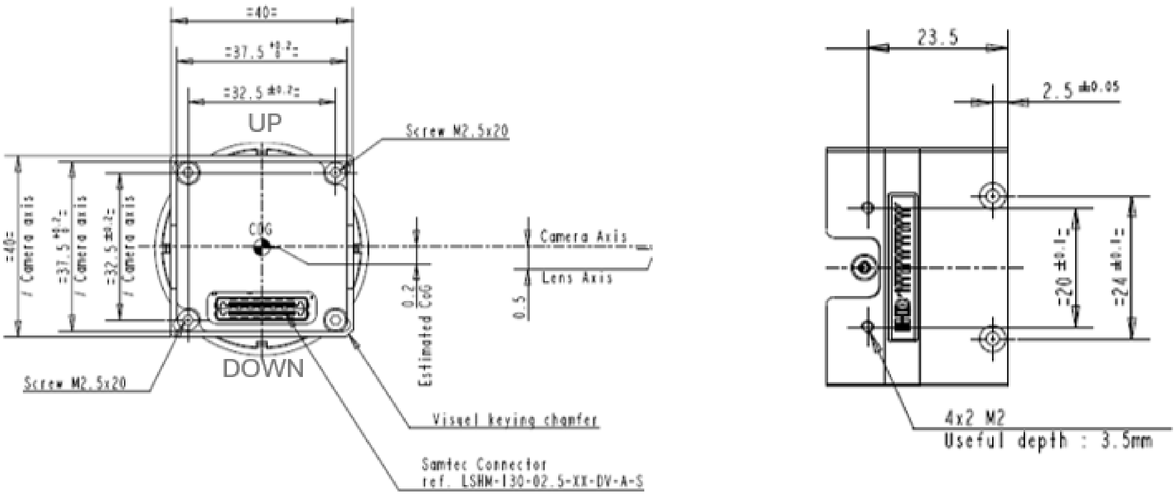

Figure 3: Additional drawing of the CamSight HD core

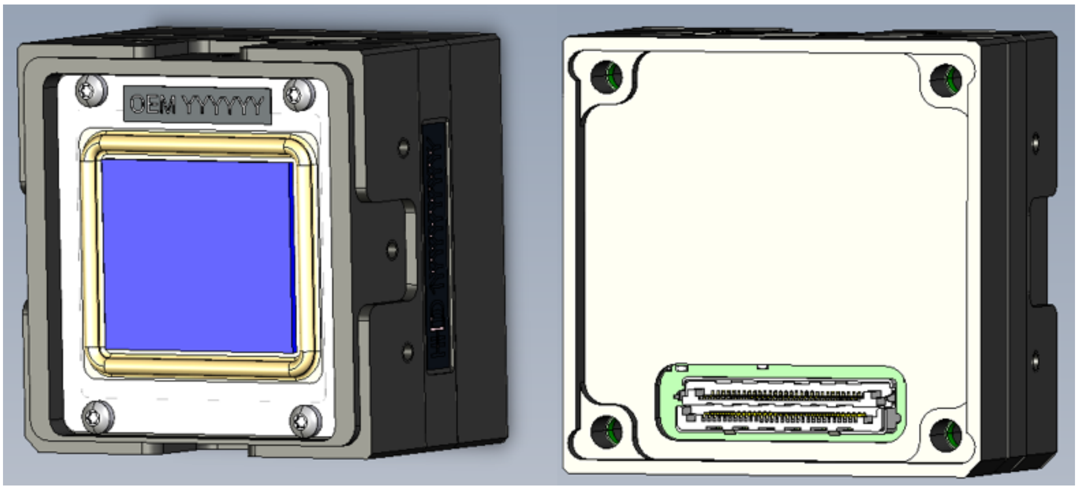

Figure 4: 3D rendering of the CamSight HD core

3.1.2. Shutter

The core shutter version of the CamSight HD camera requires mechanical support to maintain the lens and hide the parasitic infrared light. Mechanical support is necessary to ensure proper alignment of the lens and to reduce the impact of parasitic infrared light on the image quality. The core camera weighs 128 g.

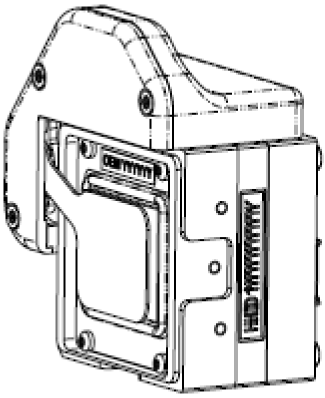

Figure 5: Drawing of the CamSight HD core with shutter

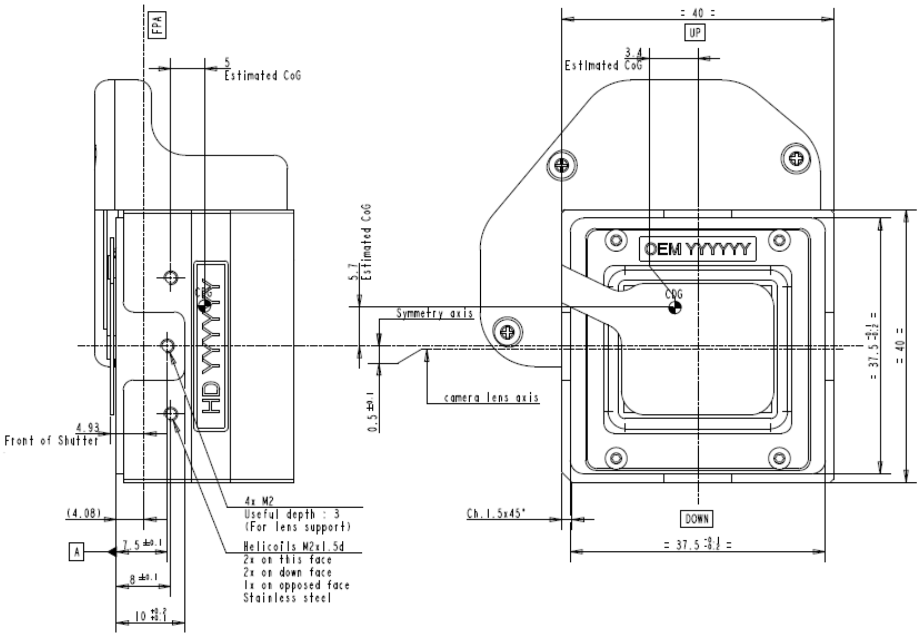

Figure 6: Additional drawing of the Camsight HD core with shutter

Figure 7: 3D rendering of the CamSight HD core with shutter

3.2. Lens

The OEM version of the CamSight HD camera includes a lens that requires mechanical support to maintain it and hide the parasitic infrared light. It is necessary to ensure proper alignment of the lens and to reduce the impact of parasitic infrared light on the image quality.
This section contains partial information. Other lens configurations are available upon request.

3.2.1. Lens 9.3mm

The camera supports the following lens:

Figure 8: 3D rendering of the 9.3mm lens with mechanical support

Dimension	Value
Diameter	55 mm
Length	84.7 mm
Weight	301 gr
Field of view	95.0° x 76.0°
Mounting screws	M2

Table 2: 9.3mm Lens characteristics

3.2.2. Lens 15mm

The camera supports the following lens:

Figure 9: 3D rendering of the 15mm lens with mechanical support

Dimension	Value
Diameter*	43 mm
Length	77.4 mm
Weight	226 gr
Field of view	60.0° x 47.5°
Mounting screws	M2

Table 3: 15mm Lens characteristics

**Diameter of the thickest ring*

3.2.3. Lens 19mm

The camera supports the following lens:

Figure 10: 3D rendering of the 19mm lens with mechanical support

Dimension	Value
Diameter	46.4 mm
Length	85.1 mm
Weight	271 g
Field of view	46.4° x 37.1°
Mounting screws	M2

Table 4: 19mm Lens characteristics

3.2.4. Lens 35mm

The camera supports the following lens:

Figure 11: 3D rendering of the 35mm lens with mechanical support

Dimension	Value
Diameter	45 mm
Length	51 mm
Weight	144 g
Field of view	24.6° x 19.9°
Mounting screws	M2

Table 5: 35mm Lens characteristics

3.2.5. Lens 55mm

The camera supports the following lens:

Figure 12: 3D rendering of the 55mm lens with mechanical support

Dimension	Value
Diameter	68 mm
Length	79.8 mm
Weight	304 g
Field of view	15.8° x 12.7°
Mounting screws	M2

Table 6: 55mm Lens characteristics

3.2.6. Lens 75mm

The camera supports the following lens:

Figure 13: 3D rendering of the 75mm lens with mechanical support

Dimension	Value
Diameter	86.0 mm
Length	97.4 mm
Weight	491 g
Field of view	11.6° x 9.3°
Mounting screws	M2

Table 7: 75mm Lens characteristics

3.2.7. Lens 100mm

The camera supports the following lens:

Figure 14: 3D rendering of the 100mm lens with mechanical support

Dimension	Value
Diameter	112.6 mm
Length	117 mm
Weight	925 g
Field of view	9.0° x 7.0°
Mounting screws	M2

Table 8: Lens 100mm characteristics

3.3. Interfaces

Several additional electronic boards are available to adapt the video output and better meet users' needs. As a reminder, interface boards are add-on modules that connect directly to the core camera.

3.3.1. MIPI CSI-2 interface board

The MIPI interface board provides image output on an Amphenol connector. Depending on the core camera configuration, you will have an output with 8 bits or 14 bits.

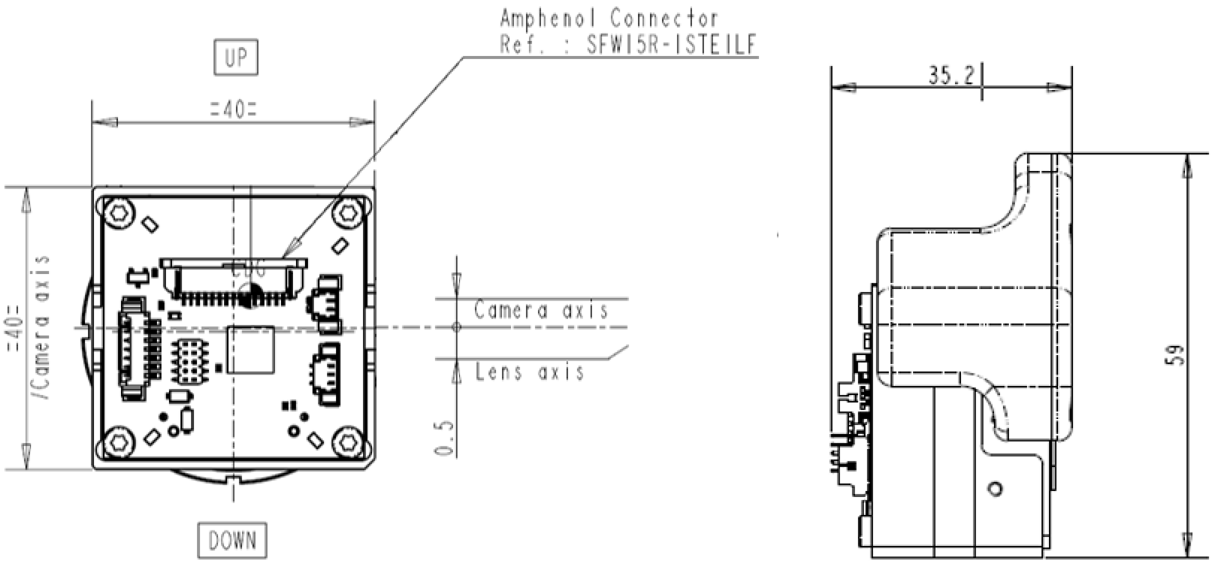

Figure 15: Drawing of MIPI interface board

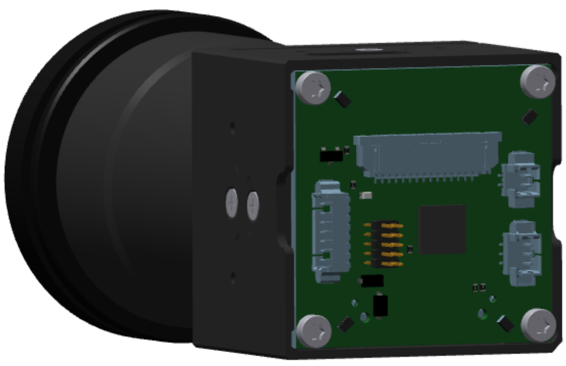

Figure 16: 3D rendering of the CamSight HD with an 8.5 mm lens and the MIPI interface board

3.3.2. GigE Vision interface board

The GigE Vision interface board provides image output on an RJ45 connector with 8 bits. It is made of:

- A interface board
- a Pleora stack boards (NTx-GigE Vision) composed of :
  - a processing board
  - an Ethernet board
- an optional flat flexible connector (recommended) between the two boards to distance the Pleora boards from the camera in order to avoid heat emission from Pleora boards on the sensor.

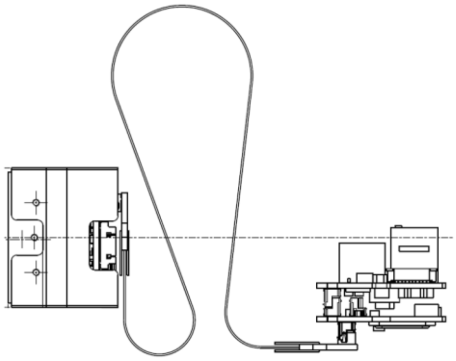

Figure 17: Drawing of the CamSight HD core with GigE Vision interface board

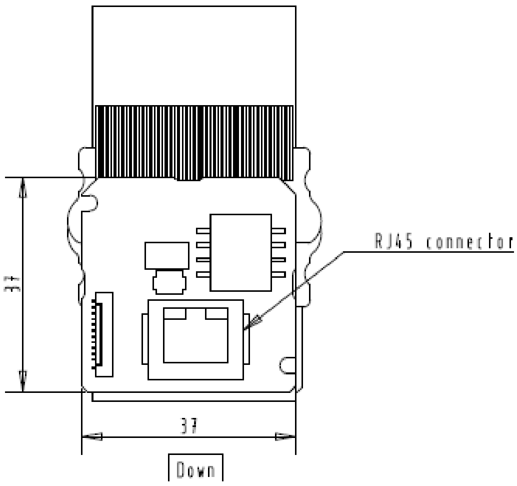

Figure 18: Drawing of the GigE Vision interface boards

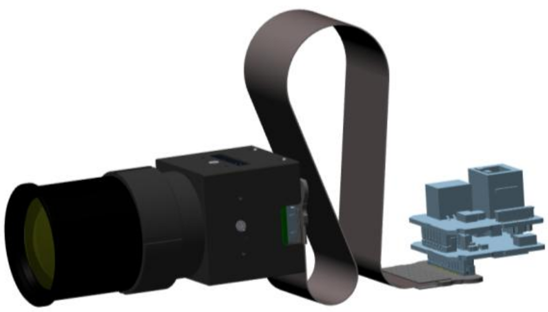

Figure 19: 3D rendering of the CamSight HD with a 15 mm lens and the GigE Vision interface

4. ELECTRICAL INTERFACE

4.1. Boards connectors

4.1.1. Core camera

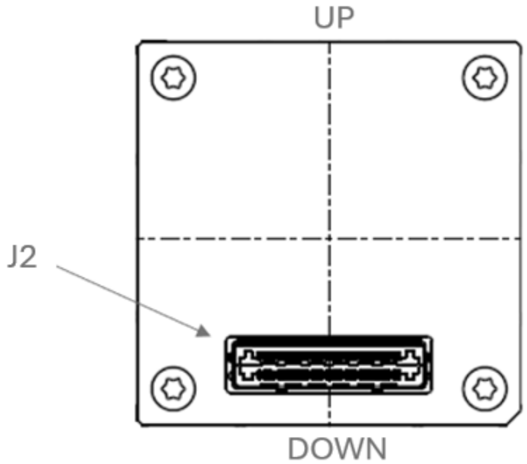

Figure 20: Drawing of the core camera connector

Connector	Connector reference	Mating connector
J2	SAMTEC LSHM-130-02.5-XX-DV-A-S	SAMTEC LSHM-130-02.5-XX-DV-A-S

Table 9: Core board connector reference

J1 – 0.50MM TERMINAL/SOCKET COMBO,60PINS (LSHM-130-02.5-XX-DV-A-S)							
Voltage	**I/O**	**Name**	**Pin**	**Pin**	**Name**	**I/O**	**Voltage**
SDI	O	RESERVED	1	2	GND	IO	-
-	IO	GND	3	4	RESERVED	I	1V8
1V8	I	RESERVED	5	6	RESERVED	O	1V8
1V8	I	RESERVED	7	8	RESERVED	I	1V8
-	IO	GND	9	10	GND	IO	-
5V	I	5V	11	12	UART0_RX	I	3V3
5V	I	5V	13	14	UART0_TX	O	3V3
-	IO	GND	15	16	GND	IO	
3V3	O	VIDEO_OUT_HSYNC	17	18	VIDEO_OUT_DE	O	3V3
3V3	O	VIDEO_OUT_VSYNC	19	20	VIDEO_OUT_D6	O	3V3
3V3	O	VIDEO_OUT_TRIG	21	22	VIDEO_OUT_D8	O	3V3
3V3	O	VIDEO_OUT_PCLK	23	24	VIDEO_OUT_D13	O	3V3
	IO	GND	25	26	VIDEO_OUT_D2	O	3V3
3V3	O	VIDEO_OUT_D14	27	28	VIDEO_OUT_D5	O	3V3
3V3	O	VIDEO_OUT_D12	29	30	VIDEO_OUT_D3	O	3V3
3V3	O	VIDEO_OUT_D0	31	32	VIDEO_OUT_D1	O	3V3
3V3	O	VIDEO_OUT_D9	33	34	VIDEO_OUT_D7	O	3V3
3V3	O	VIDEO_OUT_D4	35	36	VIDEO_OUT_D10	O	3V3
3V3	O	VIDEO_OUT_D15	37	38	VIDEO_OUT_D11	O	3V3
3V3	O	3V3	39	40	GND	IO	-
-	IO	GND	41	42	RESERVED	O	3V3
	I	SDI REFCLK_N	43	44	RESERVED	I	3V3
	I	SDI REFCLK_P	45	46	RESERVED	O	3V3
3V3	O	RESERVED	47	48	GND	IO	-
3V3	O	RESERVED	49	50	RESERVED	O	3V3
3V3	O	RESERVED	51	52	RESERVED	O	3V3
	IO	GND	53	54	RESERVED	O	3V3
3V3	O	I2C1_SCL	55	56	RESERVED	O	3V3
3V3	IO	I2C1_SDA	57	58	1V8	O	1V8
3V3	O	DRIVER ENABLE	59	60	GND	IO	-

Table 10: Core board J1 connector pinout

4.1.2. MIPI Interface Board

When using the MIPI interface board, 14-bit payload is aligned on the 16-bit parallel output LSB:

- Bit15 & bit14 = “00”
- Bit13...bit0 = mono14 stream

The output bits depth is therefore 16.

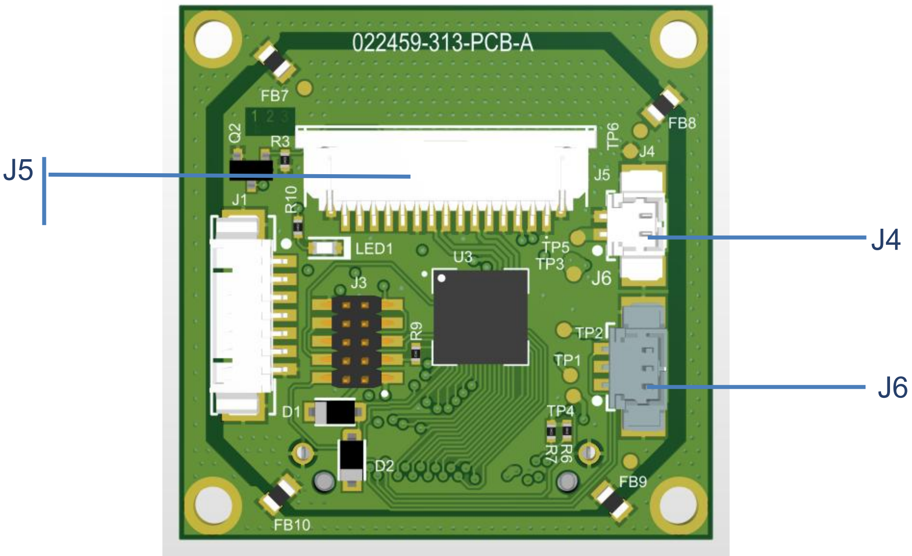

Figure 21: Drawing of MIPI interface board connectors

NOTE: J1 and J3 are factory reserved connectors.

Connector	Connector reference	Mating connector
J4	MOLEX 53398-0271	MOLEX 510210200
J5	SFW15R1STE1LF	FFC cable
J6	MOLEX 53398-0371	MOLEX 510210300

Table 11: MIPI interface board connectors reference

J4 – POWER SUPPLY (Molex 53398-0271)			
Pin	**Name**	**I/O**	**Voltage**
1	POWER	I	4.8v to 5.4V
2	POWER GND	I	N/A

Table 12: MIPI board J4 connector pinout

J6 – UART (022459-313-PCB-B or higher version) (Molex 53261-0371)			
Pin	**Name**	**I/O**	**Voltage**
1	UART **TX**	O	3.3V
2	GND	IO	-
3	UART **RX**	I	3.3V

Table 13: MIPI board J6 connector pinout (Camera output)

J5 – MIPI CSI-2 (SFW15R-1STE1LF)			
Pin	**Name**	**I/O**	**Voltage**
1	GND	IO	
2	CSI_D0_N	O	MIPI
3	CSI_D0_P	O	MIPI
4	GND	IO	
5	CSI_D1_N	O	MIPI
6	CSI_D1_P	O	MIPI
7	GND	IO	
8	CSI_CLK_N	O	MIPI
9	CSI_CLK_P	O	MIPI
10	GND	IO	
11	POWER_EN	I	POWER_EN turn on threshold > 1.9V POWER_EN turn off threshold < 0.6V Operating range [0V...6.5V]
12	TRIGGER	I	V_{min}=0V – Vmax=+5V V_{IH} = +1.5V
13	CSI_I2C_SCL		Open collector +3.3V with integrated (R11) 4.7kΩ pull-up resistor
14	CSI_I2C_SDA		Open collector +3.3V with integrated (R12) 4.7kΩ pull-up resistor
15	Unused		

Table 14: MIPI board J5 connector pinout (camera output)

All MIPI signals comply with MIPI CSI-2 D-PHY standard.

4.1.3. GigE Vision interface board

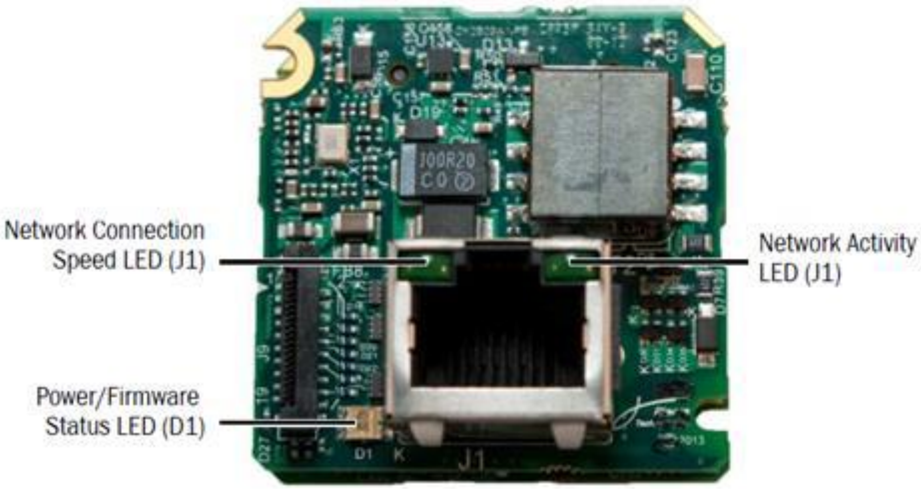

Figure 22: Picture of the GigE Vision interface board

RJ45 connector for Ethernet Network connection, as specified in IEEE 802.3 standard.

The Ethernet interface can operate at 100 or 1000 Mbps and supports Internet Protocol Version 4 (IPv4). Bitrate to be chosen depending on core camera's configuration (8 or 14 bits, frequency of 60Hz).

4.2. Power Supplies

4.2.1. Core camera

Input pin	Requirement	Min	Max
VCC input	Voltage	4.8 V*	5.4 V*
	Current	1 A	-
	Ripple	-	50 mV

Table 15: Core camera power supplies requirements

*: absolute minimum and maximum over temperature range.

4.2.2. MIPI interface Board

Input pin	Requirement	Min	Max
VCC Digital	Voltage	4.8 V*	5.4 V*
	Current	1 A	-
	Ripple	-	50 mV

Table 16: MIPI interface board power supplies requirements

*: absolute minimum and maximum over temperature range.

4.2.3. GigE Vision interface Board

This is a PoE Ethernet powered interface in accordance with IEEE 802.3af standard, up to 7 Watts.

4.3. Interface configurations

4.3.1. Serial electrical interface

To configure and communicate with CamSight HD cameras, a serial UART interface is available on all electronic boards and extension boards.

WARNING: Depending on the electronic board you are connecting to, the logic levels can be either 1.8V or 3.3V, refer to your board electrical interface for the specific line voltage to use.

UART Parameter	Value
Default baud rate	115200 baud/s
Start bit	1
Stop bit	1
Number of data bits	8
Parity bit	None

Table 17: UART Serial interface parameters

4.3.2. I2C MIPI interface

To configure and communicate with CamSight HD cameras with a MIPI CSI-2 output, an I2C interface is available on the MIPI electronics boards.

I2C Parameter	Value
Clock speed	400kHz
Camera address	0x30

Table 18: I2C MIPI interface parameters

5. IMAGE PROCESSING

Depending on your core camera's configuration (M8 or M14 bits), the image processing pipeline will be different. Therefore, you will have access to different features. The associated features are described in paragraph5.2.

Note that switching from one configuration to another is not dynamic and requires reprogramming the firmware of the CSHD camera core.

5.1. Camera pipeline overview

5.1.1. CamSight HD M8 bits pipeline

The CamSight HD M8 bits image processing pipeline is the following:

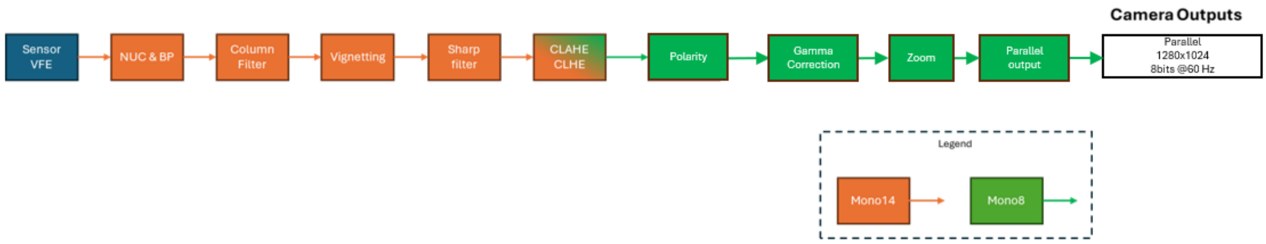

Figure 23: **Video stream pipeline** of the CamSight HD core 60 Hz M8

5.1.2. CamSight HD M14 pipeline

The CamSight HD M14 bits image processing pipeline is the following:

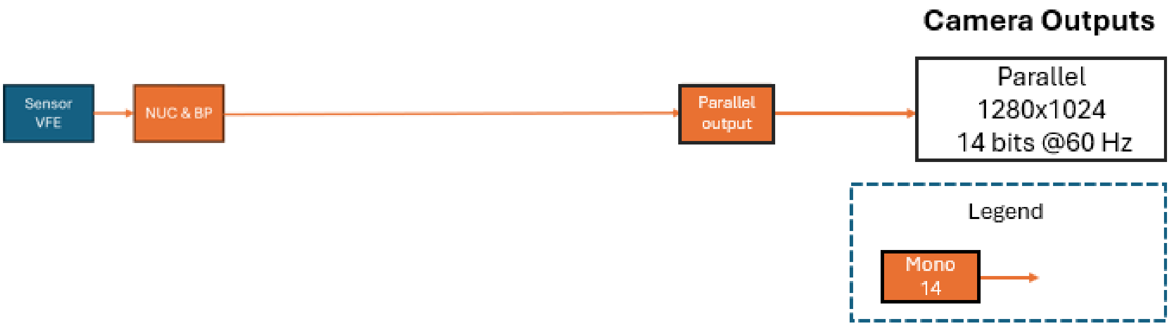

Figure 24: **Video stream pipeline** of the CamSight HD core 60 Hz M14

5.1.3. Other interfaces

The different interfaces you can choose from only affect the camera's output. The camera's processing pipeline remains the same regardless of the interface you use. You can find more information about the outputs of the different interfaces in the paragraph 4. The table below summarizes the various possible configurations.

	9Hz	25Hz	30Hz	50Hz	60Hz
Parallel	X	14 bits	X	X	8bits and 14 bits
HD-SDI	X	8bits	X	X	X
MIPI	X	X	X	X	8bits and 16 bits
GigE	X	14 bits	X	X	8 bits and 14 bits

Table 19: CamSight HD interfaces framerate

5.2. Features overview

5.2.1. Bad pixels

The CamSight HD integrates the factory list of bad pixels and corrects them in real time with each frame.

5.2.2. Shutterless operation

The CamSight HD camera features shutterless technology that allows for low-latency and freeze-free image delivery without a mechanical shutter. This technology compensates for thermal drift and other sources of noise that can affect image quality in traditional shuttered cameras.

In addition to the shutterless technology, the CamSight HD camera is designed to operate optimally over a wide temperature range, from -40°C to +70°C. To achieve this, the camera has several operating points, each optimized for a specific temperature range. When transition occurs between two temperature operating points some artifacts can be observed on the image during typically less than half a second.

5.2.3. Less shutter operation

The CamSight HD camera features an internal shutter that improves image quality. This operation is performed manually at the operator’s request.

5.2.4. Column noise correction

The column noise is a type of non-uniformity that can be induced by the variation in the response of the reference bolometers in the sensor. However, this noise can be effectively reduced by using the “Column noise correction” algorithm available in the camera. This algorithm works by estimating the amount of noise present in each column of the image and then removing it. However, it is important to note that this algorithm may sometimes create artefacts on vertical structures when they are too big in the image. Nonetheless, the Column noise correction algorithm is an important tool for improving the overall image quality of the camera.

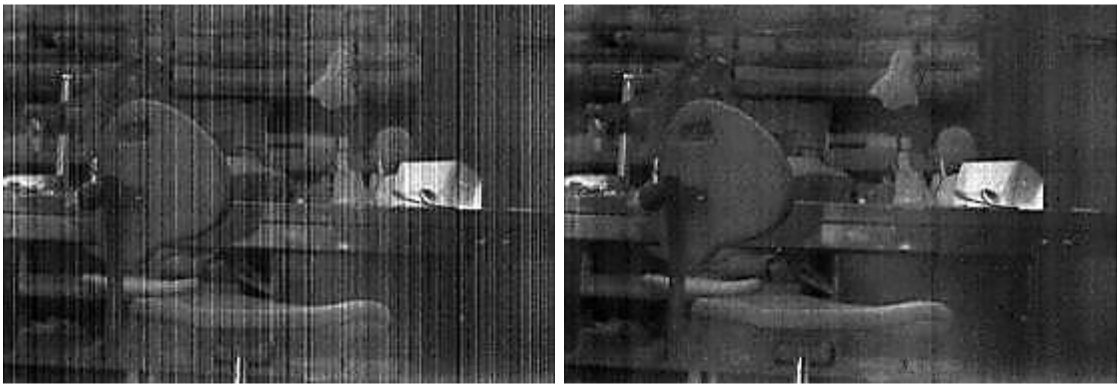

Figure 25: Left picture-column noise correction OFF/Right picture-column noise correction ON

Sharpness is increased by 75% on both images to better observe the effect on thumbnails.

5.2.5. Vignetting correction (flattening filter)

The Vignetting correction algorithm is a powerful tool to reduce non-uniformities induced by temperature gradients on the sensor, also called vignetting. This non-uniformity is a common issue in shutterless cameras and can significantly degrade the image quality. The algorithm works by analyzing the image to detect areas with different temperature gradients and applying a correction factor to each pixel based on its location. This correction factor compensates for the non-uniformities and restores the uniformity of the image.

However, it is important to note that the Vignetting correction algorithm can sometimes create artefacts on temperature uniform surfaces, such as the sky. This is because the algorithm may interpret the uniform temperature as a temperature gradient and apply a correction factor that is not needed. Overall, the Vignetting correction algorithm is a valuable tool for improving the image quality.

Figure 26: Left picture - vignetting correction OFF / Right picture - vignetting correction ON

5.2.6. Edge enhancement

The camera is equipped with an edge enhancement algorithm which can improve the image quality by enhancing the edges. In sharpening mode, the edges of the objects in the image are enhanced, resulting in a sharper image.

Figure 27: Left picture - sharpening OFF / Right picture - Sharpening ON

5.2.7. Histogram dynamic range enhancement

The camera includes two dynamic range correction algorithms that optimize the histogram processing of the scene.

There are two algorithms:

- CLHE (Contrast-Limited Histogram Equalization)
- CLAHE (Contrast-Limited Adaptive Histogram Equalization).

A parameter allows adjusting these algorithms to be more linear or more histogram equalization. While the CLHE algorithm works on the histogram computed over a single ROI (Region of Interest) defined by the user, CLAHE divides the image into multiple ROI and processes the histogram of each part of the image.

Depending on the scenes encountered of the final application, CLHE or CLAHE should be selected for best results.

Figure 28: Left picture - CLHE / Right picture – CLAHE

5.2.8. Polarity

The camera features an image algorithm to invert image polarity (White Hot vs Black Hot)

Figure 29: Left picture - White Hot / Right picture - Black Hot

5.2.9. Gamma correction

A gamma correction can be applied to the image to modify its luminosity and contrast. Gamma values higher than 1 result in an overall darkening of the image while a gamma value less than 1 results in an overall lightening of the image.

The gamma correction value can be set by the user depending on the imaged scene.

<u>NOTE:</u> The gamma computation is approximated, resulting in a lack of brightness boosting for low values of gamma (3rd order approximation of $(1+x)^{\gamma}$).

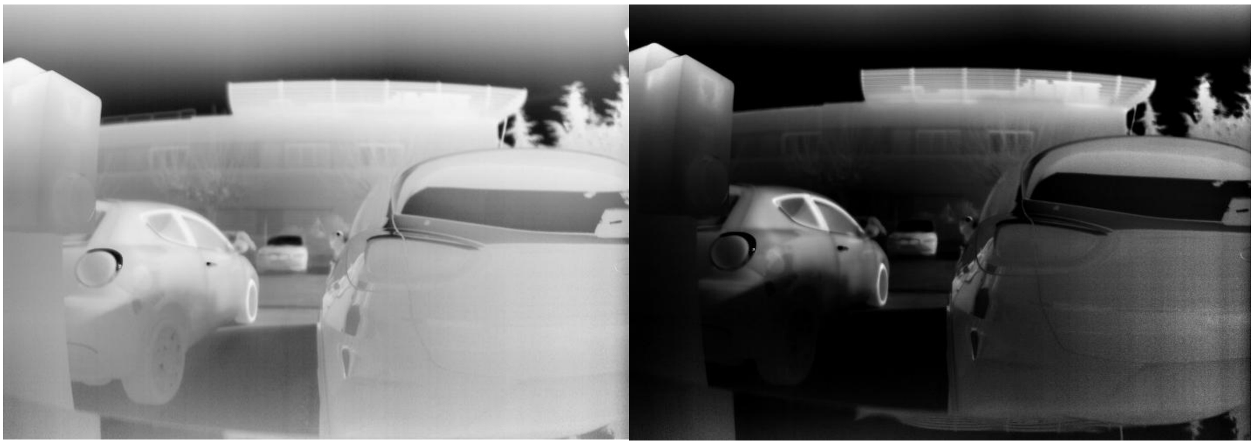

Figure 30: Left picture – Low gamma correction / Right picture – High gamma correction

5.2.10. Zoom

The camera features a digital zoom up to x8. Two zoom methods are available: Nearest and Bilinear.

In the Nearest mode, the new pixel value is obtained by copying the value of the nearest neighboring pixel.

In the Bilinear mode, the new pixel value is computed from its surrounding neighbors whose values are linearly interpolated.

Figure 31: CamSight HD with 8.5mm lens: Left picture – Zoom x1 / Right picture – Zoom x8

6. TRIGGER

6.1. Camera trigger modes

The CamSight HD can operate in two different trigger modes:

- Freerun mode: the camera generates its own trigger signal to trigger IR sensor image generation. No external action is required.
- External trigger mode: the camera uses an external signal to trigger IR sensor image generation. An external valid trigger signal, matching both electrical and logical requirements, is needed.

One image is generated for each trigger signal, either internal (freerun mode) or external.

Trigger signal rising edge is used.

Trigger mode can be changed dynamically using i2c interface.

Camera trigger mode is described in below diagram.

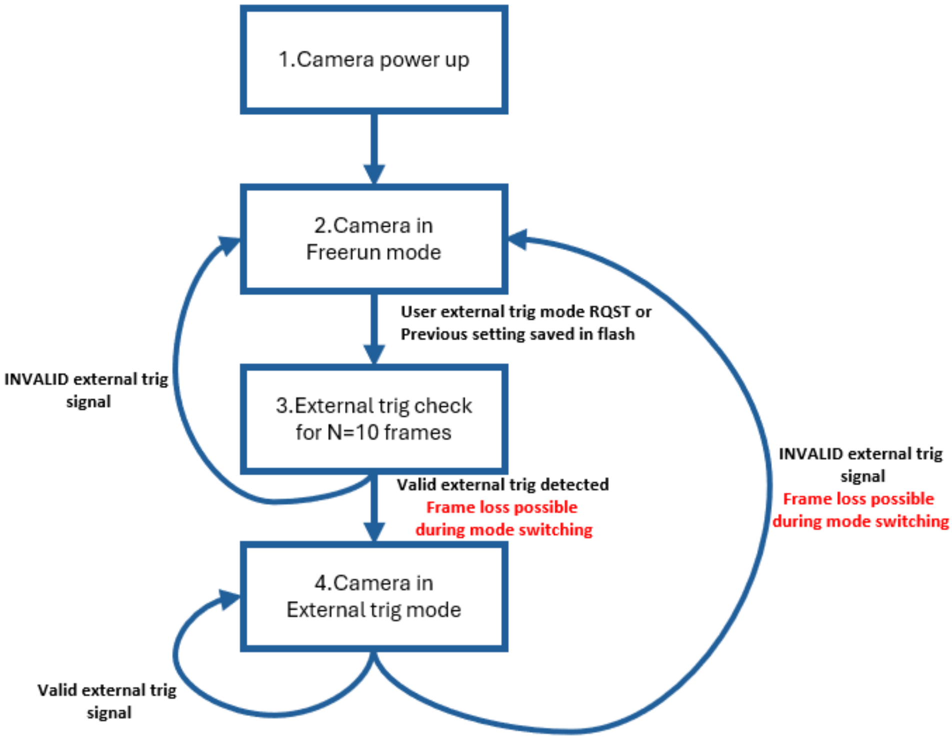

Figure 32: Camera trigger mode behaviour

When camera is set in external trigger mode, either at power up or dynamically by user, camera trigger management is described below:

- At power up, camera starts in freerun mode.
- Then if the user requests switching to the external trigger mode or if this setting was saved during last power cycle, camera starts checking for a valid trigger signal.
- If external trigger signal meets the following requirements during at least 10 frames, the camera switches to external trigger mode. A frame loss might occur during mode switching.
- The camera stays in external trigger mode as long as the external trigger signal is valid (see following requirements).
- If the external trigger signal becomes invalid, the camera falls back in freerun mode to ensure image delivery. A frame loss might occur during mode switching.
- The camera starts checking for an external trigger signal again.

6.2. External trigger signal specification

External signal must respect:

- Frequency: must be +/- 0.1% of camera FPS
- Duty cycle: 50% recommended, > 10% required
- Electrical levels: specified in Table 9, VIDEO_OUT_TRIG pin.

External trigger signal is continuously monitored and when detected as invalid (out of range frequency or duty cycle), camera falls back to freerun mode to ensure image generation.

The camera automatically switches back to external trigger mode if the signal is detected as valid again.

A status bit is available in the trigger mode register to check that external trigger signal is detected and valid. Check corresponding I2C table ***TRIGGER_MODE.***

Example:

A valid configuration for CSHD 60Hz:

- Frequency: 60Hz (16.66666 ms)
- Duty cycle: 6ms ON / 10ms Off

7. SOFTWARE INTERFACE

7.1. SERIAL communication interface

7.1.1. MavLink Protocol

The CamSight MAVLink Protocol is based on MAVLink v2 protocol. All transactions are composed of 1 command packet sent by the master device and 1 response packet sent by the camera.

7.1.2. MAVLink versioning

The CamSight MAVLink Protocol is versioned. All devices implementing this protocol must ensure compatibility by adhering to the protocol implementation guidelines.

The `GET_CAMERA_PROTOCOL_VERSION` command is used by the master device to identify the available camera command set.

Historical versions of the CamSight MAVLink Protocol is not versioned (doesn't implement the `GET_CAMERA_PROTOCOL_VERSION` command) and is identified in this document as version 0.

7.1.3. MAVLink generator

Pymavgen is used to generate automatically the MAVLink code libraries to interface with the Camera: https://github.com/mavlink/mavlink

This tool can generate the MAVLink protocol implementation in a variety of languages, from embedded C to Python or C#, Rust etc.

The library generator program 'mavgenerate.py' takes as input an XML file describing the command set to be implemented in the library. An XML file is available for each version of CamSight MAVLink Protocol on request.

The output of the MAVLink library generator is a file or directory of files depending on the target language.

WARNING: The generated output should be used directly as a library and not modified, if regenerated, the modified files will be overwritten. Prefer creating wrappers instead of modifying code directly.

The MAVLink library sits as a data encoder/decoder, the programmer must use the higher level API to send and receive messages as well as indicate to the generated library how to interface with the hardware serial communication (read/write functions).

Here is an example of this implementation for a C language target:

```
#ifndef MAVLINK_H
#define MAVLINK_H
//----------------------------------------------------------------------------
// HRo: configuration part for MavLink messaging
#include "mavlink_types.h"
#include "serial_interface.h"

#define MAVLINK_USE_CONVENIENCE_FUNCTIONS          1

#define MAVLINK_SEND_UART_BYTES                    serial_interface_send_bytes
extern mavlink_system_t mavlink_system;
//----------------------------------------------------------------------------
```

7.1.4. MAVLink transactions

The communication between the camera and the host is structured as a "Request/Answer" or "Command/Response" exchange. The camera is in a slave role and only responds to commands sent by the host as shown in Figure .

The answers for SET commands are `MESSAGE_ACK/MESSAGE_NACK` packets for SET messages. For GET messages, the same message is echoed with the PAYLOAD section filled by the camera.

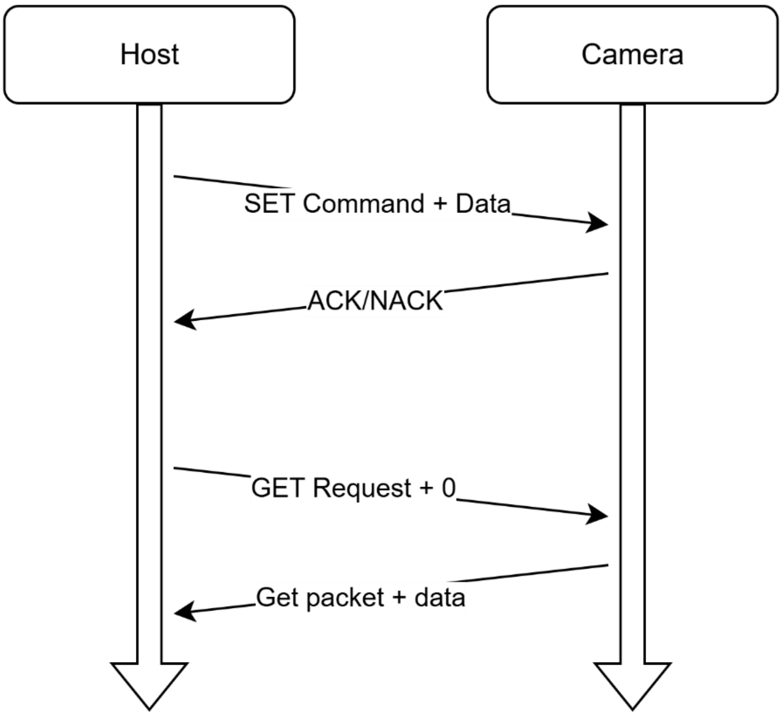

Figure 33: MAVLink communication flow

7.1.4.1. Master to Slave command/request

Requests/Commands are sent from the host to the camera. Two types or Requests/Commands exist:

- GET commands, asking the camera to send back information
- SET commands, setting parameters values and configuration

All requests fall into these two categories, even if they don't contain the words "Get" or "Set".

Every Request/Command is structured as indicated by Table .

B0	B1	B2-B3	B4	B5-B6	B7-B9	B10-B(N+9)	B(N+10)-B(N+11)
0xFD	LEN	RESERVED	SEQ	RESERVED	MID	PAYLOAD	CRC16

Table 20: MAVLink command format

Bytes:

- 0xFD: Magic number/start byte for MAVLink V2 communication.
- LEN: Length of the PAYLOAD, expressed in bytes; can be from 0 to 255 bytes.
- RESERVED: Must be kept 0.
- SEQ: Sequence number, incremented at each message sent/received.
- RESERVED: Must be kept 0.
- MID: Message ID (24 bits) of the sent command.
- PAYLOAD of LEN bytes:
  - Command data depends on the command type.
  - 0 in case of GET commands
- CRC16: 16-bit CRC (MCRF4XX) computed over B1 to B(N+9) bytes.

NOTE: Each MAVLink packet is coded in little endian format.

7.1.4.2. Slave to Master acknowledgement/response

Every request command is followed after some time by an acknowledgement or an answer (in the case of a GET command) whose structure is indicated by Table .

B0	B1	B2-B3	B4	B5-B6	B7-B9	B10-(N+9)	B(N+10)-B(N+11)
0xFD	LEN	RESERVED	SEQ	RESERVED	MID	PAYLOAD	CRC16

Table 21: MAVLink answer format

Bytes:

- 0xFD: Magic number/start byte for MAVLink V2 communication.
- LEN: Length of the PAYLOAD, expressed in bytes; can be from 0 to 255 bytes.
- RESERVED: Reserved value, must be kept 0.
- SEQ: Sequence number, incremented at each message sent/received.
- RESERVED: Reserved value, must be kept 0.
- MID: Message ID (24 bits) of the answered Request/Command.
- PAYLOAD of LEN bytes:
  - Requested data for GET commands
  - MESSAGE_ACK packet data for acknowledgement
- CRC16: 16-bit CRC (MCRF4XX) computed over all messages bytes.

NOTE: Each MAVLink packet is coded in little endian format.

7.1.4.3. Timing and errors recovery

The MESSAGE_ACK message is used as the default answer for SET commands. Unknown/Unimplemented messages are answered with a MESSAGE_NACK packet.

If a command fails to execute/raise an error, a MESSAGE_NACK packet is also emitted.

Value	Min	Mean	Max
Time between message and answer		100ms	1.5s
Number of recommended retries	1	3	

Table 22: Error recovery recommendations

7.1.4.4. Streaming protocol

A streaming protocol is used to allow transferring huge amounts of data through 2 specific streaming commands. These streams can be either read or write streams.

The write and read streaming transfers are built around MAVLink transactions extended protocol messages: WRITE_STREAM and READ_STREAM.

7.1.4.4.1. Write Stream Transfer format

The MAVLink master always initiates the transaction with a WRITE_STREAM_START indicating the type of data transferred, its length and destination address, the slave acknowledges the stream start by sending back a MESSAGE_ACK. If the slave sends a MESSAGE_NACK packet instead, the stream is aborted.

The stream packets are then sent by the master and individually acknowledged or not the slave device responding with MESSAGE_ACK or MESSAGE_NACK packet.

If a MESSAGE_ACK packet is received by the master, the next write packet is sent with its SEQ control byte incremented.

However, if a MESSAGE_NACK packet is received, the master must resend the same stream packet with same SEQ control byte. If the slave device keeps sending a MESSAGE_NACK after this 2^{nd} try, it must at least try 2 more times before cancelling the transfer.

At the end of a successful transfer, the master sends a STREAM_END packet containing the 32 bits CRC computed over all sent data bytes (PAYLOAD bytes only).

Note that in case of a failed transaction (MESSAGE_NACK response), the packet data should still be counted in the final CRC, as the local CRC is updated with each incoming packet, regardless of the transaction success.

The slave device computes and sends back the CRC32 of all received data (PAYLOAD bytes only) and sends it back via a STREAM_END packet.

The sequence of packets in a Read Stream Transfer is detailed in Table .

Master request	Slave response
WRITE_STREAM_START	MESSAGE_ACK
WRITE_STREAM	MESSAGE_ACK
...	...
WRITE_STREAM	MESSAGE_ACK
STREAM_END	STREAM_END

Table 23: MAVLink Stream write transactions

7.1.4.4.2. Read Stream Transfer format

The MAVLink master always initiates the transaction with a READ_STREAM_START indicating the type of data transferred, its length and starting address, the slave acknowledges the stream start by sending back a MESSAGE_ACK. If the slave sends a MESSAGE_NACK packet instead, the stream is aborted.

The stream packets are then sent by the master and individually acknowledged or not the slave device responding with a READ_STREAM containing the read data byte or a MESSAGE_NACK packet indicating an error has occurred.

If a READ_STREAM packet is received by the master, the next read packet is sent with its SEQ control byte incremented.

However, if a MESSAGE_NACK packet is received, the master must resend the same stream packet with same SEQ control byte. If the slave device keeps sending a MESSAGE_NACK after this 2nd try, it must at least try 2 more times before cancelling the transfer.

At the end of a successful transfer, the master sends a STREAM_END packet containing the 32 bits CRC computed over all sent data bytes (PAYLOAD bytes only).

Note that in case of a failed transaction (MESSAGE_NACK response), the packet data should still be counted in the final CRC, as the local CRC is updated with each incoming packet, regardless of the transaction success.

The slave device computes and sends back the CRC32 of all received data (PAYLOAD bytes only) and sends it back via a STREAM_END packet.

The sequence of packets in a Read Stream Transfer is detailed in Table .

Master request	Slave response
READ_STREAM_START	MESSAGE_ACK
READ_STREAM	READ_STREAM
...	...
READ_STREAM	READ_STREAM
STREAM_END	STREAM_END

Table 24: MAVLink Stream read transactions

7.1.5. MAVLink reference

7.1.5.1. CAMERA COMMON USER

7.1.5.1.1. MESSAGE_ACK

[RESERVED] Generic message to acknowledge messages and receive errors

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	COMMAND				VALUE		

B16	B17	B18	B19
VALUE	RESULT	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 9 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 8192 (0x2000)
- **B9-B12:** COMMAND Of type `uint32_t`. Acknowledged message ID
- **B13-B16:** VALUE Of type `uint32_t`, defaults to 0. Optional additional information.
- **B17:** RESULT Of type `uint8_t`. Message result
  Allowed values are defined by the MESSAGE_ACK_RESULT enum:
  - 0: MESSAGE_ACK_OK Message processed successfully
  - 1: MESSAGE_ACK_NOK Message processing failed

▶ **B18-B19:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.2. GET_SERIALNUMBER

[GET] Get the camera serial number (SN)

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	SERIAL_NUMBER				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 8194 (0x2002)
- **B9-B12:** SERIAL_NUMBER Of type `uint32_t`. Camera serial number
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.3. SHUTTER_CONTROL

[SET] Direct camera shutter control

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	COMMAND	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 8206 (0x200E)
- **B9:** COMMAND Of type `uint8_t`. Shutter command
  Allowed values are defined by the SHUTTER_CONTROL_CMD enum:
  - 0: SHUTTER_CONTROL_CMD_OPEN Open the camera shutter
  - 1: SHUTTER_CONTROL_CMD_CLOSE Close the camera shutter
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.4. SHUTTER_CHECK_PRESENCE

[GET] Check if the camera is equipped with a mechanical shutter

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	IS_PRESENT	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 9000 (0x2328)
- **B9:** IS_PRESENT Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.5. GET_CAMERA_PROT_VERS

[GET] Get the MAVLink protocol version

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	CAMERA_PROTOCOL_VERSION	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC — Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN — Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED — Reserved value, must be kept 0
- **B4:** SEQ — Sequence number, incremented at each command
- **B5-B6:** RESERVED — Reserved value, must be kept 0
- **B7-B8:** MID — Message ID, here equal to 12287 (0x2FFF)
- **B9:** CAMERA_PROTOCOL_VERSION — Of type `uint8_t`. MAVLINK camera communication protocol version
  Allowed values are defined by the CAMERA_PROTOCOL_VERSION enum:
  - 6: PROTOCOL_V6 — Camera communication protocol version 6
  - 5: PROTOCOL_V5 — Camera communication protocol version 5
  - 4: PROTOCOL_V4 — Camera communication protocol version 4
  - 3: PROTOCOL_V3 — Camera communication protocol version 3
  - 2: PROTOCOL_V2 — Camera communication protocol version 2
  - 1: PROTOCOL_V1 — Camera communication protocol version 1
  - 0: PROTOCOL_V0 — Camera communication protocol version 0
- **B10-B11:** CRC16 — 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.6. SET_GAMMA

[SET] Set the gamma contrast curve on IR sensor

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	VALUE				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12290 (0x3002)
- **B9-B12:** VALUE Of type `uint32_t`, between 327678 (min) and 163840 (max).
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.7. SET_CONTRAST

[SET] Set the histogram clip value for the CLHE/CLAHE algorithms

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	VALUE				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12292 (0x3004)
- **B9-B12:** VALUE Of type `uint32_t`, between 0 (min) and 30000 (max). CLHE/CLAHE clip threshold
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.8. INVERT_POLARITY

[SET] Set the camera contrast polarity

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12294 (0x3006)
- **B9:** ENABLE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.9. ROI_CONTROL

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	X_START		X_END		Y_START		Y_END

B16	B17	B18
Y_END	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 8 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12297 (0x3009)
- **B9-B10:** X_START Of type `uint16_t`. ROI top-left corner horizontal distance from image top-left corner
- **B11-B12:** X_END Of type `uint16_t`. ROI bottom-right corner horizontal distance from image bottom-right corner
- **B13-B14:** Y_START Of type `uint16_t`. ROI top-left corner vertical distance from image top-left corner
- **B15-B16:** Y_END Of type `uint16_t`. ROI bottom-right corner vertical distance from image bottom-right corner
- **B17-B18:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.10. CONTRAST_CONTROL

[SET] Change the image processing contrast algorithm in use

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	TYPE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12300 (0x300C)
- **B9:** TYPE Of type `uint8_t`. Contrast algorithm used
  Allowed values are defined by the CONTRAST_TYPE enum:
  - 0: CONTRAST_CLHE Contrast algorithm used is CLHE (Contrast Limited Histogram Equalization)
  - 1: CONTRAST_CLAHE Contrast algorithm used is CLAHE (Contrast Limited Adaptative Histogram Equalization)
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.11. CAMERA_STATUS

[GET] Get complete status of the camera

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	CONTRAST				LUMINOSITY		

B16	B17	B18	B19	B20	B21	B22	B23
LUMINOSITY	FOCUS_ERROR	SHUTTER_ERROR	FOCUS_MODE	FOCUS_ACTION	FOCUS_POSITION		

B24	B25	B26	B27	B28	B29
FOCUS_POSITION	NUC_MODE	NUC_STATUS	IR_POLARITY	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 19 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12303 (0x300F)
- **B9-B12:** CONTRAST Of type `uint32_t`. Current CLHE/CLAHE clip threshold, see [SET_CONTRAST]
- **B13-B16:** LUMINOSITY Of type `uint32_t`. Current gamma value, see [SET_GAMMA]
- **B17:** FOCUS_ERROR Of type `uint8_t`.
- **B18:** SHUTTER_ERROR Of type `uint8_t`. Shutter status
- **B19:** FOCUS_MODE Of type `uint8_t`.

- **B20:** FOCUS_ACTION Of type `uint8_t`.
- **B21-B24:** FOCUS_POSITION Of type `uint32_t`, between -100 (min) and 100 (max). [Deprecated] Previously used for cameras with focusing abilities
- **B25:** NUC_MODE Of type `uint8_t`. Current NUC mode, see [NUC_CONTROL]
- **B26:** NUC_STATUS Of type `uint8_t`.
- **B27:** IR_POLARITY Of type `uint8_t`. Current image polarity, see [INVERT_POLARITY]
- **B28-B29:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.12. SET_CUSTOM_SPEED

[SET] Enable/Disable custom MAVLink UART speed

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12308 (0x3014)
- **B9:** ENABLE Of type `int8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.13. SET_ZOOM_PARAMS

[SET] Set the camera zoom parameters

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	X_FACTOR				Y_FACTOR		

B16	B17	B18	B19	B20	B21	B22	B23
Y_FACTOR	X_CENTER				Y_CENTER		

B24	B25	B26
Y_CENTER	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 16 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12310 (0x3016)
- **B9-B12:** X_FACTOR Of type `uint32_t`, between 65536 (min) and 524288 (max).
- **B13-B16:** Y_FACTOR Of type `uint32_t`, between 65536 (min) and 524288 (max).
- **B17-B20:** X_CENTER Of type `uint32_t`. X zoom center, horizontal distance in pixels from image top-left corner
- **B21-B24:** Y_CENTER Of type `uint32_t`. Y zoom center, vertical distance in pixels from image top-left corner
- **B25-B26:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.14. SET_ZOOM_METHOD

[SET] Set the zoom algorithm used

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	METHOD	CRC16	

<u>NOTE:</u> Each MAVLink packet is coded in little endian format.

<u>Bytes:</u>

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12311 (0x3017)
- **B9:** METHOD Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.15. GET_ROI

[GET] Get the Region Of Interest's coordinates used for the CLHE histogram calculation

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	X1		X2		Y1		Y2

B16	B17	B18
Y2	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 8 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12315 (0x301B)
- **B9-B10:** X1 Of type `uint16_t`. ROI top-left corner horizontal distance from image top-left corner
- **B11-B12:** X2 Of type `uint16_t`. ROI bottom-right corner horizontal distance from image bottom-right corner
- **B13-B14:** Y1 Of type `uint16_t`. ROI top-left corner vertical distance from image top-left corner
- **B15-B16:** Y2 Of type `uint16_t`. ROI bottom-right corner vertical distance from image bottom-right corner
- **B17-B18:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.16. GET_ZOOM_CONFIG

[GET] Get the complete zoom configuration

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	X_FACTOR				Y_FACTOR		

B16	B17	B18	B19	B20	B21	B22	B23
Y_FACTOR	X_CENTER				Y_CENTER		

B24	B25	B26	B27
Y_CENTER	METHOD	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 17 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12316 (0x301C)
- **B9-B12:** X_FACTOR Of type `uint32_t`, between 65536 (min) and 524288 (max). X resize factor, see [SET_ZOOM_PARAMS]
- **B13-B16:** Y_FACTOR Of type `uint32_t`, between 65536 (min) and 524288 (max). Y resize factor, see SET_ZOOM_PARAMS
- **B17-B20:** X_CENTER Of type `uint32_t`. X zoom center, see [SET_ZOOM_PARAMS]
- **B21-B24:** Y_CENTER Of type `uint32_t`. Y zoom center, see [SET_ZOOM_PARAMS]
- **B25:** METHOD Of type `uint8_t`. Zoom method, see [SET_ZOOM_METHOD]
- **B26-B27:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.17. GET_CONTRAST_TYPE

[GET] Get IR sensor contrast algorithm used

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	TYPE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12320 (0x3020)
- **B9:** TYPE Of type `uint8_t`. Contrast algorithm used
  Allowed values are defined by the CONTRAST_TYPE enum:
  - 0: CONTRAST_CLHE Contrast algorithm used is CLHE (Contrast Limited Histogram Equalization)
  - 1: CONTRAST_CLAHE Contrast algorithm used is CLAHE (Contrast Limited Adaptative Histogram Equalization)
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.18. GET_FIRMWARE_ID

[GET] Get both FPGA and RISCV firmware ID

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	FPGA_VERSION		RISCV_VERSION		CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12321 (0x3021)
- **B9-B10:** FPGA_VERSION Of type `uint16_t`. FPGA build version
- **B11-B12:** RISCV_VERSION Of type `uint16_t`. RISCV build version
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.19. SET_FLIP_H

[SET] Enable/Disable horizontal flip

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12323 (0x3023)
- **B9:** ENABLE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.20. GET_FLIP_H

[GET] Get the horizontal flip status

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12322 (0x3022)
- **B9:** ENABLE Of type `uint8_t`. Horizontal flip status, see [SET_FLIP_H]
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.21. SET_FLIP_V

[SET] Enable/Disable vertical flip

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12325 (0x3025)
- **B9:** ENABLE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.22. GET_FLIP_V

[GET] Get the vertical flip status

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12324 (0x3024)
- **B9:** ENABLE Of type `uint8_t`. Vertical flip status, see [SET_FLIP_V]
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.23. SET_COLUMN_CORRECTION

[SET] Enable/Disable the IR sensor column correction

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	VALUE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12326 (0x3026)
- **B9:** VALUE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.24. GET_COLUMN_CORRECTION

[GET] Get IR sensor column correction value

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	VALUE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12327 (0x3027)
- **B9:** VALUE Of type `uint8_t`. Column correction status, see [SET_COLUMN_CORRECTION]
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.25. SET_VIGNETTING_CORRECTION

[SET] Enable/Disable IR sensor vignetting correction

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	VALUE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12328 (0x3028)
- **B9:** VALUE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.26. GET_VIGNETTING_CORRECTION

[GET] Get IR sensor vignetting correction value

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	VALUE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12329 (0x3029)
- **B9:** VALUE Of type `uint8_t`. Vignetting correction status, see [SET_VIGNETTING_CORRECTION]
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.27. GET_BIT

[GET] Get the Built In Test register. See documentation for details.

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	BIT				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12358 (0x3046)
- **B9-B12:** BIT Of type `uint32_t`. Built in test
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.28. GET_CAMERA_TEMPERATURE

[GET] Get the system temperatures. See documentation for details.

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	FPGA_TEMPERATURE				SENSOR_TEMPERATURE		

B16	B17	B18
SENSOR_TEMPERATURE	CRC16	

<u>NOTE:</u> Each MAVLink packet is coded in little endian format.

<u>Bytes:</u>

- **B0:** MAGIC — Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN — Data payload length in bytes, here equal to 8 byte(s)
- **B2-B3:** RESERVED — Reserved value, must be kept 0
- **B4:** SEQ — Sequence number, incremented at each command
- **B5-B6:** RESERVED — Reserved value, must be kept 0
- **B7-B8:** MID — Message ID, here equal to 12359 (0x3047)
- **B9-B12:** FPGA_TEMPERATURE — Of type `uint32_t`, in mK. FPGA temperature expressed in mK
- **B13-B16:** SENSOR_TEMPERATURE — Of type `uint32_t`, in mK. External sensor temperature expressed in mK
- **B17-B18:** CRC16 — 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.29. GET_TYPE

[GET] Get the camera and sensor type

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	TYPE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12288 (0x3000)
- **B9:** TYPE Of type `uint8_t`. Camera model and sensor type
  Allowed values are defined by the CAMERA_TYPE enum:
  - 0: TYPE_VISIBLE Visible sensor
  - 1: TYPE_INFRARED Infrared sensor
  - 2: CAMSIGHT_LS CamSight LS camera model
  - 3: CAMSIGHT_HD CamSight HD camera model
  - 4: CAMSIGHT_HDLP CamSight HD LP camera model
  - 5: CAMSIGHT_LP CamSight LP camera model
  - 6: FOR_IRGC FOR IR GC camera model
  - 7: FOR_IRPC FOR IR PC camera model
  - 8: FOR_VIS FOR Visible camera model
  - 9: SMARTSIGHT_IR SmartSight IR camera model
  - 10: SMARTSIGHT_VIS SmartSight Visible camera model

- ▷ 11: CAMSIGHT_METEO CamSight Meteo camera model
- ▷ 12: CAMSIGHT_IA CamSight IA camera model
- ▷ 13: CAMAXE CamAxe camera model
- ▷ 21: CAMSIGHT_FUSION_BLOCK CamSight Fusion Block model

▶ **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.30. GET_RESOLUTION

[GET] Get the camera image size

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	WIDTH				HEIGHT		

B16	B17	B18
HEIGHT	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 8 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12289 (0x3001)
- **B9-B12:** WIDTH Of type `uint32_t`. Sensor image width in pixels
- **B13-B16:** HEIGHT Of type `uint32_t`. Sensor image height in pixels
- **B17-B18:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.31. NUC_CONTROL

[SET] Set Non-Uniformity Correction mode

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	MODE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12295 (0x3007)
- **B9:** MODE Of type `uint8_t`. Enable or disable the NUC correction
  Allowed values are defined by the NUC_MODE enum:
  - 0: NUC_DISABLE Disable NUC correction on the video pipe.
  - 1: NUC_AUTO_TEMPERATURE NUC is automatically managed by camera, auto-triggers a NUC request when the temperature drifts past a certain threshold
  - 2: NUC_ENABLE Enable NUC correction on the video pipe.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.32. NUC_REQUEST

[SET] Trigger the acquisition of a new Non-Uniformity Correction

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	OPTION	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12296 (0x3008)
- **B9:** OPTION Of type `uint8_t`. NUC request type
  Allowed values are defined by the NUC_REQUEST_OPTION enum:
  - 0: NUC_REQUEST_OPTION_NONE No option. NUC is directly processed
  - 1: NUC_REQUEST_OPTION_WITH_SHUTTER Process the NUC with shutter closed, automatically reopens the camera shutter after NUC
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.33. ENABLE_GAIN

[SET] Enable/Disable gain correction

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12312 (0x3018)
- **B9:** ENABLE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.34. ENABLE_OFFSET

[SET] Enable/Disable offset correction

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12313 (0x3019)
- **B9:** ENABLE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.35. ENABLE_BPR

[SET] Enable/Disable bad pixel replacement

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	ENABLE	CRC16	

<u>NOTE:</u> Each MAVLink packet is coded in little endian format.

<u>Bytes:</u>

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12314 (0x301A)
- **B9:** ENABLE Of type `uint8_t`.
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.36. GET_SENSOR_CONFIG

[GET] Return complete IR sensor configuration

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	GSK				GFID		

B16	B17	B18	B19	B20	B21	B22	B23
GFID	GMS				TINT		

B24	B25	B26	B27	B28	B29
TINT	GAIN_ENABLED	OFFSET_ENABLED	BPR_ENABLED	CRC16	

<u>NOTE:</u> Each MAVLink packet is coded in little endian format.

<u>Bytes:</u>

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 19 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12317 (0x301D)
- **B9-B12:** GSK Of type `uint32_t`, in mV. GSK value
- **B13-B16:** GFID Of type `uint32_t`, in mV. GFID value
- **B17-B20:** GMS Of type `uint32_t`. GMS value
- **B21-B24:** TINT Of type `uint32_t`. TINT value
- **B25:** GAIN_ENABLED Of type `uint8_t`. Gain correction status

- **B26:** OFFSET_ENABLED Of type `uint8_t`. Offset correction status
- **B27:** BPR_ENABLED Of type `uint8_t`. BPR correction status
- **B28-B29:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.37. SET_TRIG_MODE

[SET] Set the trigger mode, internal or external, in the trigger IP

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11
MID	MODE	CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 1 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12364 (0x304C)
- **B9:** MODE Of type `uint8_t`. Trigger source
  Allowed values are defined by the MAV_TRIG_MODE_ENUM enum:
  - 0: MAV_TRIG_MODE_INTERNAL New frame is automatically triggered by the camera
  - 1: MAV_TRIG_MODE_EXTERNAL New frame is triggered externally using the VIDEO_TRIG signal
- **B10-B11:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.38. GET_TRIG_MODE

[GET] Get the trigger mode adn the status of the trigger IP

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14	B15
MID	MODE	STATUS				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 5 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12365 (0x304D)
- **B9:** MODE Of type `uint8_t`. Trigger source
  Allowed values are defined by the MAV_TRIG_MODE_ENUM enum:
  - 0: MAV_TRIG_MODE_INTERNAL New frame is automatically triggered by the camera
  - 1: MAV_TRIG_MODE_EXTERNAL New frame is triggered externally using the VIDEO_TRIG signal
- **B10-B13:** STATUS Of type `uint32_t`.
- **B14-B15:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.39. SET_SHARPENING

Set IR sensor sharpening value

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	VALUE				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12318 (0x301E)
- **B9-B12:** VALUE Of type `uint32_t`, between 0 (min) and 10240 (max). UQ8.8 fixed point value
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.1.5.1.40. GET_SHARPENING

Get IR sensor sharpening value, see [SET_SHARPENING]

B0	B1	B2	B3	B4	B5	B6	B7
MAGIC	LEN	RESERVED		SEQ	RESERVED		MID

B8	B9	B10	B11	B12	B13	B14
MID	VALUE				CRC16	

NOTE: Each MAVLink packet is coded in little endian format.

Bytes:

- **B0:** MAGIC Magic number for MAVLink V2, equal to 253 (0xFD)
- **B1:** LEN Data payload length in bytes, here equal to 4 byte(s)
- **B2-B3:** RESERVED Reserved value, must be kept 0
- **B4:** SEQ Sequence number, incremented at each command
- **B5-B6:** RESERVED Reserved value, must be kept 0
- **B7-B8:** MID Message ID, here equal to 12319 (0x301F)
- **B9-B12:** VALUE Of type `uint32_t`. UQ8.8 fixed point value
- **B13-B14:** CRC16 16-bit CRC (MCRF4XX) computed over header and data bytes

7.2. MIPI CSI I2C communication interface

7.2.1. MIPI CSI-2 video output

As described in chapter §5.1, the camera outputs its video stream over a MIPI CSI-2 interface.

This interface is composed of

- ✓ 1 clock lane
- ✓ 2 data lanes
- ✓ 1 I²C control interface (3.3V)

The clock and data lane signal levels are MIPI CSI-2 compliant. The clock frequency of the CSI-2 interface is 800MHz.
The Mono16 format requires doubling the MIPI frequency (800MHz) compared to the Mono8 format (400MHz).

When a Mono14 stream is selected, the MIPI interface board is encapsulating camera 16b parallel bus in a 16b MIPI stream:

- ▶ MIPI Data Type (DT) = 0x2E = RAW-16 image data
- ▶ 14b camera payload is on MIPI RAW-16-bit bus LSB

7.2.2. MIPI Control

"**POWER_EN**" signal allows to control camera power when resistor R16 is soldered (Figure).

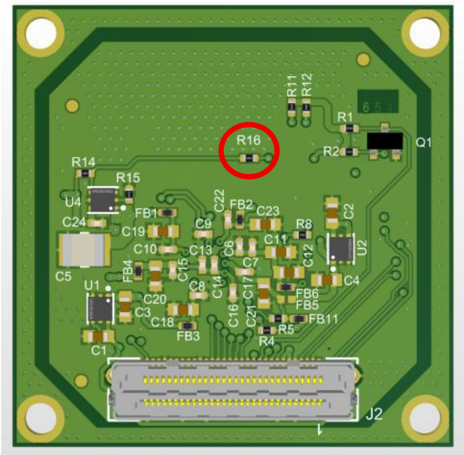

Figure 34: R16 location on MIPI CSI2 interface board

Check connector description (4) to locate connector position and required electrical levels:

- ▶ When driven high, camera power is enabled
- ▶ When driven low, camera power is turned off

This signal enables or disables entirely camera power, thus allowing to restart MIPI start sequence.

7.2.3. MIPI I2C configuration interface

The control part of the MIPI CSI-2 standard relies on a standard I²C interface. This interface allows an external processor to configure and control the camera module.

The maximum frequency for the SCK signal is 400KHz.
The address of the camera on this bus is: 0x30 (7bits).

7.2.3.1. I²C Registers summary

ADDRESS	REGISTER	R / W	DESCRIPTION
0x00	DEVICE_INFO	R	Device type and register map version.
0x01	FPGA_VERSION (LSB)	R	Version of the FPGA (LSB).
0x02	FPGA_VERSION (MSB)	R	Version of the FPGA (MSB).
0x03	RISCV_VERSION (LSB)	R	Version of the RISCV (LSB).
0x04	RISCV_VERSION (MSB)	R	Version of the RISCV (MSB).
0x05	CAMERA_SN_BYTE0	R	Serial number of the camera (Byte 0).
0x06	CAMERA_SN _BYTE1	R	Serial number of the camera (Byte 1).
0x0B	FPGA_TEMPERATURE (LSB)	R	FPGA Temperature.
0x0C	FPGA_TEMPERATURE (MSB)	R	FPGA Temperature.
0x0D	SENSOR_TEMPERATURE (LSB)	R	Sensor temperature
0x0E	SENSOR_TEMPERATURE (MSB)	R	Sensor temperature
0x0F	CONTRAST_CFG	RW	Contrast configuration register
0x10	GAMMA_FACTOR	RW	Configure the gamma correction factor
0x11	CONTRAST_VALUE (LSB)	RW	Set histogram limit for CLAHE algorithm. (LSB)
0x12	CONTRAST_VALUE (MSB)	RW	Set histogram limit for CLAHE algorithm. (MSB)
0x13–0x36	RESERVED	-	
0x37	OUTPUT_CFG_2	RW	Configure hardware output and correction type.
0x38-0x43	RESERVED	-	
0x44	FLIP_CFG	RW	Configures the image flip
0x45	EDGE_CFG	RW	Configures the edge sharpening processing
0x46	EDGE_GAIN (LSB)	RW	The value of the sharpening gain
0x47	EDGE_GAIN (MSB)	RW	The value of the sharpening gain
0x48-0x4D	RESERVED	-	
0x4E	CAMERA_SN _BYTE2	R	Serial number of the camera (Byte 2).
0x4F	CAMERA_SN _BYTE3	R	Serial number of the camera (Byte 3).
0x50	SHUTTER	RW	Open and Close shutter
0x51	NUC_1POINT_RQST	RW	Set NUC 1 point reference
0x52	TRIGGER_MODE	RW	Set camera trig mode (freerun or external)

NOTE : All other addresses are reserved and must not be accessed or written to.

All values accessible through multiple registers must be read starting from the lowest register address.

For example, to read the FPGA temperature, you must first read CAMERA_TEMP_LSB and then CAMERA_TEMP_MSB.

7.2.3.2. I²C Registers description

7.2.3.2.1. DEVICE_INFO

This register contains the type of the device and the version of this register map. This version can be used to check driver compatibility.

7	6	5	4	3	2	1	0
DEVICE_ID				MAP_VERSION			
R				R			

Bit	Field	Type	Reset	Description
7-4	DEVICE_ID	R	0x03	The device ID. 0x02 = Camsight LS 0x03 = Camsight HD
3-0	MAP_VERSION	R	1h	The version of this register map. The current version is 0x01. This register is used for interface compatibility issues. Use FPGA and RISC version registers for feature availability.

7.2.3.2.2. FPGA_VERSION

These registers contain the FPGA's version. The most significant bit (MSB) value is available at address 0x02, and the least significant bit (LSB) at address 0x01.

7	6	5	4	3	2	1	0
FPGA_VERSION_LSB							
R							

7	6	5	4	3	2	1	0
FPGA_VERSION_MSB							
R							

7.2.3.2.3. RISCV_VERSION

These registers contain the RISCV's version. The most significant bit (MSB) value is available at address 0x04, and the least significant bit (LSB) at address 0x03.

7	6	5	4	3	2	1	0
RISCV_VERSION_LSB							
R							

7	6	5	4	3	2	1	0
RISCV_VERSION_MSB							
R							

7.2.3.2.4. CAMERA_SN (address = 0x05 & 0x06 & 0x4E & 0x4F)

These registers contain the serial number of the camera. The most significant bit (MSB) value is available at address 0x4F, and the least significant bit (LSB) at address 0x05.

@0x05

7	6	5	4	3	2	1	0
CAMERA_SN_BYTE0							
R							

@0x06

7	6	5	4	3	2	1	0
CAMERA_SN_BYTE1							
R							

@0x4E

7	6	5	4	3	2	1	0
CAMERA_SN_BYTE2							
R							

@0x4F

7	6	5	4	3	2	1	0
CAMERA_SN_BYTE3							
R							

7.2.3.2.5. FPGA_TEMPERATURE

These registers contain the temperature of the FPGA. The MSB value is available at the address 0x0C and the LSB at address 0x0B.

The value unit is 10 °mC.

This value is a 16bits signed integer.

Example

- Registers value = 0x157C => 5500 * 10 = 55 000m° = 55°C
- Registers value = 0x2BC0 => 11200 * 10 = 112 000m° = 112°C
- Registers value = 0xF060 => -4000 * 10 = -40 000m° = -40°C

7	6	5	4	3	2	1	0
FPGA_TEMPERATURE_LSB							
R							

7	6	5	4	3	2	1	0
FPGA_TEMPERATURE_MSB							
R							

7.2.3.2.6. SENSOR_TEMPERATURE

These registers contain the temperature of the FPA sensor. The MSB value is available at the address 0x0E and the LSB at the address 0x0D.

The value unit is 10 m°C.

This value is a 16bits signed integer.

Example

- Registers value = 0x157C => 5500 * 10 = 55 000m°C = 55°C
- Registers value = 0x2BC0 => 11200 * 10 = 112 000m°C = 112°C
- Registers value = 0xF060 => -4000 * 10 = -40 000m°C = -40°C

7	6	5	4	3	2	1	0
SENSOR_TEMPERATURE_LSB							
R							

7	6	5	4	3	2	1	0
SENSOR_TEMPERATURE_MSB							
R							

7.2.3.2.7. CONTRAST_CFG

This register configures the contrast and several LUT settings.

7	6	5	4	3	2	1	0
LUT_COLOR				Reserved	Reserved	POLARITY	CONTRAT_MODE
R/W				-	-	R/W	R/W

Bit	Field	Type	Reset	Description
7-4	RESERVED	R	0h	RESERVED
3	Reserved	-	0h	Reserved
2	Reserved	-	0h	Reserved
1	POLARITY	R/W	0h	Invert polarity. 0 = Normal 1 = Inverted
0	CONTRAT_MODE	R/W	0h	Select the contrast mode. 0 = ROI mode (CLHE) 1 = Locale mode (CLAHE)

7.2.3.2.8. GAMMA_FACTOR

This register configures the gamma correction factor. With $gamma \in [0.5; 2.5]$. The corresponding GAMMA_FACTOR follows the relation: $GAMMA\_FACTOR \in [0;\ 20]$.

This register is an unsigned integer on 8 bits.

The equation linking the register and the real gamma value is:

gamma = (GAMMA_FACTOR / 10) + 0.5

7	6	5	4	3	2	1	0
GAMMA_FACTOR							
R/W							

Bit	Field	Type	Reset	Description
7-0	GAMMA_FACTOR	R/W	0h	0 corresponds to a gamma value of 0.5

CONTRAST_VALUE

These registers contain the histogram limit value used in the CLAHE algorithm. The most significant bit (MSB) value is available at address 0x12, and the least significant bit (LSB) at address 0x11. No conversion is required; we use the raw value in pixels directly.

Mandatory: Write first LSB and then the MSB (even if MSB is unchanged)

The value must be within the range [0:size_y * size_x] where size_x = 1280 and size_y = 1024.

7	6	5	4	3	2	1	0
CONTRAST_VAL_LSB							
R/W							

7	6	5	4	3	2	1	0
CONTRAST_VAL_MSB							
R/W							

7.2.3.2.9. OUTPUT_CFG_2

This register configures the selected hardware output and corrections to apply.

7	6	5	4	3	2	1	0
COLUMN	VIGNETTING	Reserved					
R/W	R/W	-					

Bit	Field	Type	Reset	Description
7	COLUNM	R/W	1h	Enable / Disable the column correction. 0 = disable 1 = enable
6	VIGNETTING	R/W	1h	Enable / Disable the vignetting correction. 0 = disable 1 = enable
5-0	Reserved	-	0h	Reserved

7.2.3.2.10. FLIP_CFG

This register configures the zoom method and the wished image flip.

7	6	5	4	3	2	1	0
Reserved						Y_FLIP	X_FLIP
-						R/W	R/W

Bit	Field	Type	Reset	Description
7-2	Reserved	-	0h	Reserved
1	Y_FLIP	R/W	0h	Enable / Disable image flip on Y axis. 0 = disable 1 = enable
0	X_FLIP	R/W	0h	Enable / Disable image flip on X axis. 0 = disable 1 = enable

7.2.3.2.11. EDGE_CFG

This register configures the edge correction type.

7	6	5	4	3	2	1	0
Reserved						EDGE_CORRECTION	
-						R/W	

Bit	Field	Type	Reset	Description
7-2	Reserved	-	0h	Reserved
1-0	EDGE_CORRECTION	R/W	0h	Select edge correction type. 0 = None 1 = Sharpening 2-3 = Reserved

7.2.3.2.12. EDGE_GAIN

These registers contain the Sharpening filter gain. No conversion is required; we use the raw value directly. The value is an unsigned fixed point with 8 bits used for the decimal part. The 8bits of the fractional part are in the register SHARP_GAIN_LSB whereas the integer part is in the register SHARP_GAIN_MSB.

The range for the sharpening gain is from 0 (minimum enhancement) to 40 (maximum enhancement). Therefore, in fixed point the range value is from 0x0 to 0x2800.

7	6	5	4	3	2	1	0
SHARP_GAIN_LSB							
R/W							

7	6	5	4	3	2	1	0
SHARP_GAIN_MSB							
R/W							

7.2.3.2.13. SHUTTER

This register controls shutter position.

7	6	5	4	3	2	1	0
Reserved							CONTROL_SHUTTER
-							R/W

Bit	Field	Type	Reset	Description
7 - 1	Reserved	R	0h	Reserved
0	CONTROL_SHUTTER	R/W	0h	Get and set the shutter position. 0 = Open 1 = Close

7.2.3.2.14. NUC_1POINT_RQST

This register controls the NUC 1 point function.

7	6	5	4	3	2	1	0
Reserved				REQUEST_AUTO	REQUEST	STATUS	
-				W	W	R	

Bit	Field	Type	Reset	Description
7-4	Reserved	R	0h	Reserved
3	REQUEST_AUTO	W	0h	Write 1 to request an automatic NUC (with an automatic management of the shutter)
2	REQUEST	W	0h	Write 1 to request a NUC
0-1	STATUS	R	0h	Control the shutter position. 0 = Ok 1 = NUC is processing 2 = An error occurred

7.2.3.2.15. TRIGGER_MODE

This register configures camera trigger mode.

7	6	5	4	3	2	1	0
Reserved						Trig mode	STATUS
-						R/W	R

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0h	Reserved
1	TRIG_MODE	R/W	0h	Set camera trigger mode: 1 = external trigger is enabled 0 = external trigger is disabled = camera is in freerun mode
0	STATUS	R	0h	External trigger detection status. 0 = No external trigger detected or invalid trigger 1 = Valid external trigger is detected

8. ADDITIONAL SOFTWARE

8.1. Compagnon Software

CamSight HD cameras are supplied with a companion software tool to set the camera parameters in factory.

See AD002 for full documentation.

8.2. Updater Software

CamSight HD cameras are supplied with an updater tool to download new firmware or update the existing camera firmware.

See AD001 for full documentation.