



FSM-IMX462 Datasheet

Sony IMX462LQR-C Sensor Module

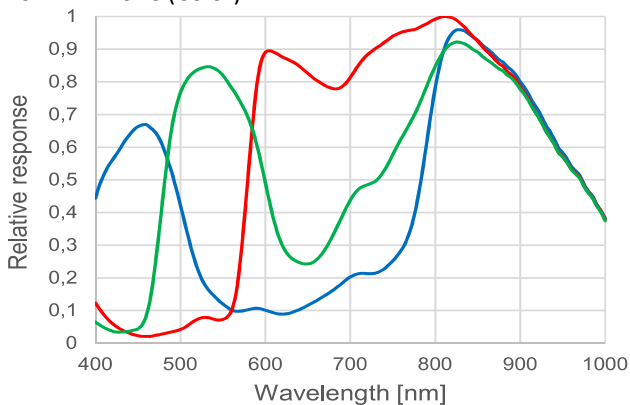
FRAMOS Sensor Module



Key Benefits & Features:

- 2.1 Mpx Sony CMOS Rolling Shutter sensor module, ready to embed!
- All FSMs are part of a rapid prototyping ecosystem, consisting of:
 - ✓ Adapters to various processing boards
 - ✓ Design sources for deep embedding
 - ✓ Various accessories and design in services

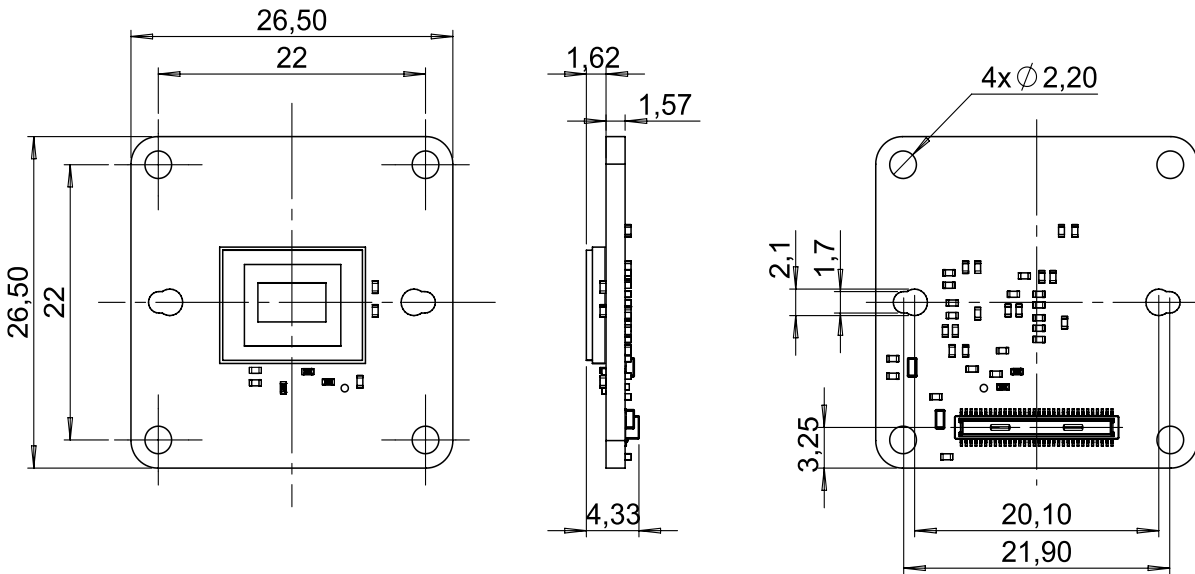
FSM-IMX462C (Color):



Specification	
Model Name	FSM-IMX462C (V1A)
Image Sensor	
Vendor / Name	Sony IMX462LQR-C
Shutter Type	CMOS Rolling Shutter
Chromaticity	Color
Optical Format	1/2.8"
Pixel Size	2.9 x 2.9 μm
Max. Resolution	2.1 Mpx / 1920 x 1080 px
Framerate (max.)	120 FPS (4-Lane) 60 FPS (2-Lane)
Bit Depth(s)	10 / 12 bit
Interface	
Data Interface	MIPI CSI-2 (2 / 4 Lane)
Communication Interface	I ² C
Drive Frequency(s)	37.125 / 74.25 MHz
Input Voltages	1.2V, 1.8V, 2.9V
Interface Connector	Hirose DF40C-60DP-0.4V(51)
EEPROM (Sensor ID)	No
Mechanical	
Dimensions (HxWxD)	26.5 mm x 26.5 mm
Environmental	
Operating Temperature	-30°C - 85°C
Storage Temperature	-40°C - 85°C
Ambient Humidity	20% to 95% RH, non condensing
Software Support	
Driver	V4L2 Based Device Driver
Supported Platform(s)	NVIDIA Jetson Family: AGX Xavier, Nano, TX2, Xavier NX
Software Version(s)	NVIDIA JP4.4 / L4T32.4.3
Suggested Accessories & Adapters	
Compatible FSA	FSA-FT3/A
Recommended Devkit(s)	FSM-IMX462C/TXA_Devkit (TX2, AGX) FSM-IMX462C/NVN_Devkit (Nano, NX)
Lens Mounts	M12 or C/CS-Mount options

A matrix with compatible *Sensor Adapters (FSA)* and *Processor Board Adapters (FPA)* for various setups can be found at the end of this document.

Mechanical Drawing



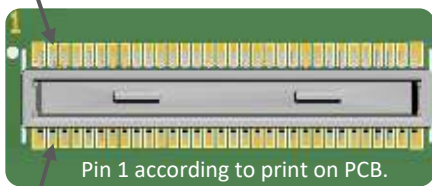
Sensor image optical center is in mechanical board center.

Connector Pinout

Type: Hirose DF40C-60DP-0.4V(51)

Mating Type: Hirose DF40HC(4.0)-60DS-0.4V(51)

Pin	Name
1	NC
3	NC
5	2V9
7	2V9
9	1V8
11	GND
13	GND
15	XCLR
17	NC
19	XMASTER
21	SCL
23	XCE
25	XVS
27	SDA
29	XHS
31	XTRIG
33	NC
35	NC
37	GND
39	INCK
41	NC
43	GND
45	NC
47	NC
49	GND
51	D_DATA_0_N
53	D_DATA_0_P
55	GND
57	CSI_D2_P
59	CSI_D2_N



Pin	Name
2	NC
4	NC
6	1V2
8	1V2
10	NC
12	GND
14	GND
16	NC
18	SDO
20	NC
22	NC
24	NC
26	NC
28	NC
30	TENABLE
32	TOUT
34	NC
36	NC
38	GND
40	NC
42	NC
44	GND
46	D_DATA_3_P
48	D_DATA_3_N
50	GND
52	D_DATA_1_N
54	D_DATA_1_P
56	GND
58	CSI_CLK_P
60	CSI_CLK_N

Signals are routed directly from image sensor to connector. Details on specific signals are described in the respective image sensor datasheet.



Table of Contents

- 1 FRAMOS Sensor Module Ecosystem 2
 - 1.1 Ecosystem Overview..... 3
 - 1.2 Materials and Services..... 4
- 2 Software Package and Drivers 5
 - 2.1 Reference Software: NVIDIA Jetson Family..... 6
 - 2.1.1 Platform and Sensor Device Drivers 6
 - 2.1.2 Image Pre-Processing Examples 7
- 3 Ecosystem Compatibility Matrix..... 10
 - 3.1 Hardware Support 10



1 FRAMOS Sensor Module Ecosystem

The FSM Ecosystem consists of FRAMOS Sensor Modules, Adapters, Software and Sources, and provides one coherent solution supporting the whole process of integrating image sensors into embedded vision products.

During the evaluation and proof-of-concept phase, off-the-shelf sensor modules with a versatile adapter framework allow the connection of latest image sensor technology to open processing platforms, like the NVIDIA Jetson Family or the 96boards.org standard. Reference drivers and sample applications deliver images immediately after installation, supporting V4L2 and an optional derivate API providing comfortable integration. Within the development phase, electrical design references and driver sources guide with a solid and proven baseline to quickly port into individual system designs and extend scope, while decreasing risk and efforts.

To simplify and relieve the whole supply chain, all FRAMOS Sensor Modules and adapters are optimized and ready for delivery in volume and customization with pre-configured lens holder, lens and further accessories.

Off-the-Shelf Hardware

- FRAMOS Sensor Modules (FSM) from stock, ready for evaluation and optimized for initial mass production.
- Versatile adapter framework, allowing flexible testing of different modules, on different processing boards:
 - FRAMOS Sensor Adapter (FSA): Everything the specific sensor needs for operation
 - FRAMOS Processor Adapter (FPA): Connect up to four FSM + FSA to a specific processor board
- From lenses, mechanics and cables, all needed imaging accessories from one hand

Ready to go Software Package

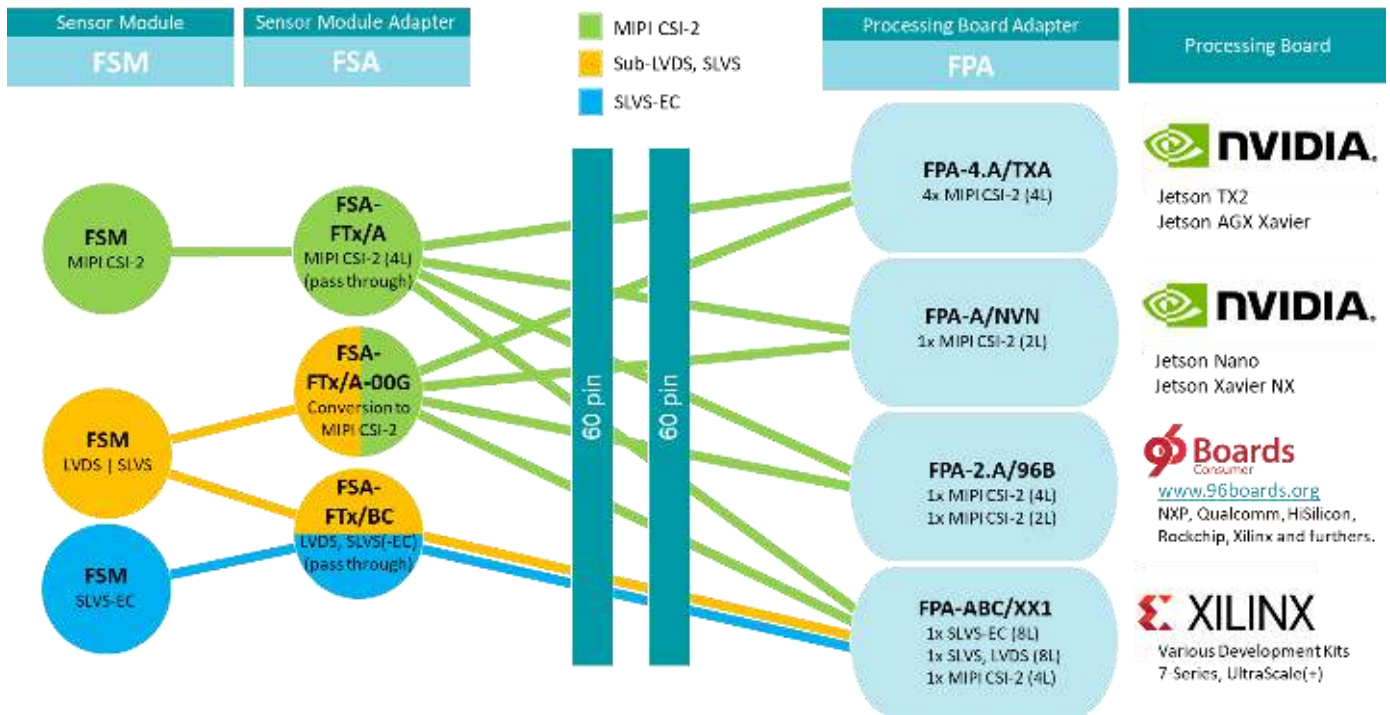
- Drivers with basic sensor integration:
 - Platform specific device drivers
 - V4L2 subdevice drivers for specific image sensors (low-level C API)
- Streamlined V4L2 library (LibSV) with comfortable and generic C/C++ API
- Example applications demonstrating initialization, main configuration and image stream processing

Further to the off-the-shelf hard- and software, the Ecosystem supports you on project basis with:

- Driver sources allowing the focus on application specific scope and sensor features
- Electrical references for FSA and FPA, supporting quick and optimized embedding of FSMs
- Engineering services via FRAMOS and its partners, allowing you to focus on your product's unique value

1.1 Ecosystem Overview

The figure below shows a map of compatibility with all components inside the Ecosystem. Every element (or hardware) and connection displayed in **Green** operates with native MIPI CSI-2 (D-PHY) data.



Every component and connection displayed in **Orange** or **Blue** operates with proprietary (Sub-LVDS, SLVS) or standardized (SLVS-EC) LVDS data, that requires further attention to the physical processing of the image data by either data conversion or specific FPGA IP. Users of MIPI CSI-2 based processing systems are supported by FSM specific data conversion located on dedicated FRAMOS Sensor Adapters (FSAs).

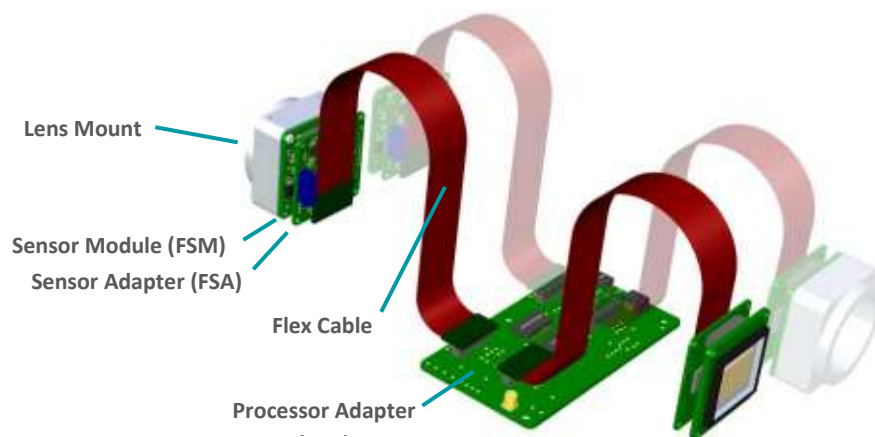


Figure 1: Assembly of a typical Sensor Module Development Kit

Specification and compatibility of all individual components are listed in the appropriate chapter of the full datasheet. Access to software and drivers is only granted with the purchase of the appropriate development kit. Electrical design sources, support and services are provided on individual basis, they are not part of the development kit or component purchase.



1.2 Materials and Services

Below you can find a list of materials and services as part of the FRAMOS Sensor Module Ecosystem.

Hardware

- FRAMOS Sensor Module Development Kits
- Individual Parts:
 - FRAMOS Sensor Modules
 - FRAMOS Sensor Adapters
 - FRAMOS Processor Adapters
 - FRAMOS Module Accessories (Cables, Mounts)

Software (part of the Development Kit)

- Software Package for NVIDIA Jetson AGX Xavier, Nano, TX2 and Xavier NX
- Software Package for DragonBoard 410c (96Boards)
- Xilinx FPGA reference implementation for SLVS-EC (Sony IMX421, IMX530)

Design Sources (on Project Basis)

- Software Driver Sources
- Electrical References for FSA, FPA (Schematics)

Design Services

- Off-the-shelf hardware customization including size, shape, connector and extended functionality
- Software customization and extension
 - Additional processor board support
 - Further sensor features and image (pre-)processing
- Integration of additional sensors
- Optimization for volume production
- Lens assembly and alignment
- System / solution development
- Production and integration

2 Software Package and Drivers

As FRAMOS we know that the getting started with a new technology is the biggest challenge. The idea behind the Software Package is to enable embedded software engineers to get quickly to a streaming system and provide at the same time all tools that are needed to extend and adapt it according the individual needs of the application.

What the software package and driver are:

- A reference for a custom sensor implementation
- Demonstrating how to use the required interfaces
- Demonstrating how to communicate with the image sensor
- Demonstrating how to generally initialize and configure the image sensor
- Provide initial image streaming output to the user space
- Demonstrating how to run basic image processing on pixel data

What it is not:

- A fully featured camera implementation - not all sensors features might be implemented
- Ready to be use in the field
- A benchmark for the capabilities of the image sensor
- Focused on image processing

Supported Processor Platforms

The table below shows which platforms are supported by the standard driver package, and how many FSMs can at maximum be operated in parallel.

Sensor Module	NVIDIA Jetson TX2	NVIDIA AGX Xavier	NVIDIA Jetson Nano, Xavier NX	DragonBoard 410c	96Boards Consumer Edition	Xilinx Development Boards	
FSM-AR0144		4	2		HW only, driver development on project basis.		
FSM-AR0521		4	2	2			
FSM-AR1335		4	2				
FSM-HDP230	4	4	2				
FSM-IMX264	2	4	-				
FSM-IMX283	2	4	-				
FSM-IMX290		4	2	2			
FSM-IMX296		4	2	2			
FSM-IMX297		4	2				
FSM-IMX304	2	4	-				
FSM-IMX327		4	2	2			
FSM-IMX334	2	4	-				
FSM-IMX335		4	2				
FSM-IMX412		4	2	2			
FSM-IMX415		4	2				
FSM-IMX462		4	2				
FSM-IMX464		4	2				
FSM-IMX477		4	2				
FSM-IMX485		4	2				
FSM-IMX577		4	2				
FSM-IMX530	2	4	-				1 ¹

Table 1: Ecosystem Software Package - Supported number of FSMs per processing board

¹ SLVS-EC based FPGA reference implementation as part of the SLVS-EC RX IP Core offering.



2.1 Reference Software: NVIDIA Jetson Family

The software package provided with the Development Kits of the FRAMOS Sensor Module Ecosystem provided for NVIDIA Jetson platforms provides a reference implementation of sensor and device drivers for MIPI CSI-2. It contains a minimum feature set demonstrating how to utilize the platform specific data interface and communication implementation, as well as the initialization of the image sensor and implementation of basic features.

Package Content:

- Platform and device drivers with Linux for Tegra Support
- V4L2 based subdevice drivers (low-level C API)
- Streamlined V4L2 library (LibSV) providing generic C/C++ API
- Image Pre-Processing Examples:
 - OpenCV (Software)
 - LibArgus (Hardware)

Supported Devices:

- Jetson Nano (B01)
- Jetson TX2
- Jetson Xavier NX
- Jetson AGX Xavier

2.1.1 Platform and Sensor Device Drivers

The driver divides into two main parts that are configured in separate ways – the Image Modes and the General Features of the image sensor.

Image Modes

These are major attributes that have impact to the image data stream formatting. They require a static pre-configuration within the device tree (DT):

- Image / streaming resolution
- Pixel format / bit depth
- Data rate / lane configuration

Each driver provides access to 3 – 5 pre-built configurations, reflecting the main operation modes of the imager. Beside the full resolution, that is always available, they allow to receive image streams in common video resolutions like VGA, Full HD and UHD as they are supported or make sense by the imagers, and utilize sensor features like ROI and binning.

They act as an example for implementation and usage and are available as source. Due to the size limitation of the device tree, it is not possible to integrate an extensive set of options.

General Features

These are attributes of the image sensor that do not manipulate the data stream formatting. The drivers provided with the Software Pack integrate the sensor features as shown in the table below.

Pre-Implemented Features per Model	Gain (Analog / Digital)	Frame Rate	Exposure Time	Flip / Mirror	IS Mode (Master / Slave)	Sensor Mode ID	Test Pattern Output	Black Level	HDR Output	Broadcast	Data Rate	Synchronizing Master
FSM-AR0144	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red	Red
FSM-AR0521	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red	Red
FSM-AR1335	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red	Red
FSM-HDP230	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red
FSM-IMX264	Green	Green	Green	Red	Green	Green	Green	Green	Red	Red	Red	Red
FSM-IMX283	Green	Green	Green	Red	Red	Green	Green	Green	Red	Red	Red	Red
FSM-IMX290	Green	Green	Green	Red	Green	Green	Green	Green	Red	Red	Green	Red
FSM-IMX296	Green	* Green	Green	Red	Green	Red	Green	Green	Red	Red	Red	Red
FSM-IMX297	Green	* Green	Green	Red	Green	Red	Green	Green	Red	Red	Red	Red
FSM-IMX304	Green	Green	Green	Red	Green	Green	Green	Green	Red	Red	Red	Red
FSM-IMX327	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Green	Red
FSM-IMX334	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Green	Green
FSM-IMX335	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Green	Green
FSM-IMX412	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Red	Red
FSM-IMX415	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Green	Green
FSM-IMX462	Green	Green	Green	Red	Green	Green	Green	Green	Red	Red	Green	Red
FSM-IMX464	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Green	Green
FSM-IMX477	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Red	Red
FSM-IMX485	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Green	Green
FSM-IMX530	Green	Green	Green	Red	Green	Green	Green	Green	Red	Red	Red	Red
FSM-IMX577	Green	Green	Green	Red	Green	Green	Green	Green	Red	Green	Red	Red

■ V4L (libsv) and libargus
■ V4L (libsv)
■ Not Implemented

Table 2: Supported sensor features on NVIDIA Jetson TX2 / AGX Xavier

*Only supported in all pixel mode

Further features, as been supported by the image sensor, can be integrated into the driver sources using the image sensor datasheet.

2.1.2 Image Pre-Processing Examples

The provided image processing examples show the general mechanisms of data handling, for an image processing using 3rd-party IP. Both, the OpenCV and the LibArgus examples do not output data that is tuned for best visual experience.

LibArgus Example:

- Closed source ISP implementation
- Using hard ISP in NVIDIA Jetson SOCs
- Most performant option
- Example Implementation: Full but not tuned image pipeline, displaying



Color tuning and lens correction needs to be calibrated for every image sensor separately and depends on sensor and lens attributes as well as illumination situation.

Image Pre-Processing Features per Model	Bad Pixel Correction	Noise Reduction	Black Level Comp.	Auto Exposure, Gain	Auto White Balance	Demosaic	Color Correction	Color Artifact Suppr.	Downscaling	Edge Enhancement
FSM-AR0144	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-AR0521	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-AR1335	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-HDP230	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX264	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX283	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX290	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX296	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX297	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX304	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX327	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX334	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX335	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX412	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX415	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX462	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX464	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX477	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX485	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX530	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow
FSM-IMX577	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow	Yellow	Yellow	Yellow

Implemented
 Using Default Config
 Not Implemented

Table 3: Implemented LibArgus features for NVIDIA Jetson TX2 / AGX Xavier

Default Config

Image streaming is performed through the LibArgus pipeline, using a common configuration. It demonstrates the usage of LibArgus but is not optimized for the certain sensor configuration and might not lead to good image representation.

As NVIDIA camera partner, FRAMOS provides appropriate ISP tuning services on project basis for the individual customer system.

OpenCV Example:

- Open software library
- Easy to use and large feature set
- Extremely performance hungry (CPU)
- Not recommended for pre-processing
- Example Implementation: Demosaicing, Displaying

Image Pre-Processing Features per Model	Bad Pixel Correction	Noise Reduction	Black Level Comp.	Auto Exposure, Gain	Auto White Balance	Demosaic	Color Correction	Color Artifact Suppr.	Downscaling	Edge Enhancement
FSM-AR0144	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-AR0521	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-AR1335	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-HDP230	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX264	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX283	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX290	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX296	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX297	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX304	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX327	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX334	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX335	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX412	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX415	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX462	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX464	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX477	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX485	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX530	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented
FSM-IMX577	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented	Implemented	Not Implemented	Not Implemented	Not Implemented	Not Implemented

■ Implemented
■ Not Implemented

Table 4: Implemented features in OpenCV example

Due to limited performance and extreme resource utilization, it is not planned to enhance the image processing support on software side.

3 Ecosystem Compatibility Matrix

3.1 Hardware Support

The following matrix shows the compatibility of FSMs, FSAs and FPAs to each other. The FSAs differentiate to each other by supplied voltages, power up sequence, generated clock (oscillator) and physical attributes.

Sensor Modules with MIPI CSI-2 (D-PHY) Output

Item	FSM-IMX412 FSM-IMX477 FSM-IMX577	FSM-IMX290 FSM-IMX327 FSM-IMX334 FSM-IMX335 FSM-IMX462 FSM-IMX464 FSM-IMX485	FSM-IMX296 FSM-IMX297	FSM-AR0521 FSM-AR1335	FSM-IMX415	FSM-IMX283	FSM-AR0144	FSM-HDP230
FSA-FT1/A	FPA-4.A/TXA FPA-A/NVN FPA-2.A/96B FPA-ABC/XX1 ²							
FSA-FT3/A		FPA-4.A/TXA FPA-A/NVN ³ FPA-2.A/96B FPA-ABC/XX1 ²						
FSA-FT6/A			FPA-4.A/TXA FPA-A/NVN FPA-2.A/96B FPA-ABC/XX1 ²					
FSA-FT7/A				FPA-4.A/TXA FPA-A/NVN FPA-2.A/96B FPA-ABC/XX1 ²				
FSA-FT11/A					FPA-4.A/TXA FPA-A/NVN FPA-2.A/96B FPA-ABC/XX1 ²			
FSA-FT12/A						FPA-4.A/TXA FPA-2.A/96B FPA-ABC/XX1 ²		
FSA-FT13/A							FPA-4.A/TXA FPA-A/NVN FPA-2.A/96B FPA-ABC/XX1 ²	
FSA-FT19/A								FPA-4.A/TXA FPA-A/NVN FPA-2.A/96B FPA-ABC/XX1 ²

Table 5: Ecosystem Compatibility Matrix – Native CSI-2 (D-PHY) FSMs

² Not verified, Xilinx Development Board with hard MIPI CSI-2 / D-PHY interface.

³ FSM-IMX334 is not supported due to the sensor requiring 4-lanes MIPI.

**Sensor Modules with (Sub-) LVDS and SLVS Output**

Item	Data Output (FSA)	FSM-IMX264	FSM-IMX304	FSM-IMX421	FSM-IMX530
FSA-FT14/A-00G	MIPI CSI-2	FPA-4.A/TXA FPA-2.A/96B FPA-ABC/XX1 ²			
FSA-FT15/A-00G	MIPI CSI-2		FPA-4.A/TXA FPA-2.A/96B FPA-ABC/XX1 ²		
FSA-FT18/A-00G	MIPI CSI-2				FPA-4.A/TXA FPA-2.A/96B FPA-ABC/XX1 ²
FSA-FT18/BC	SLVS, SLVS-EC				FPA-ABC/XX1
FSA-FT20/BC	SLVS, SLVS-EC			FPA-ABC/XX1	

Table 6: Ecosystem Compatibility Matrix – Sub-LVDS, SLVS and SLVS-EC FSMs