EAN-MIPI Cameras

2021-10-21

Exports: Export Summary Sheet

EULA: End User License Agreement

Web: sightlineapplications.com

Sales: sales@sightlineapplications.com

Support: support@sightlineapplications.com

Phone: +1 (541) 716-5137

CAUTION: Alerts to a potential hazard that may result in personal injury, or an unsafe practice that causes damage to the equipment if not avoided

IMPORTANT: Identifies crucial information that is important to setup and configuration procedures.

Used to emphasize points or reminds the user of something. Supplementary information that aids in the use or understanding of the equipment or subject that is not critical to system use.

1 Overview

This document describes how to configure the SightLine 4000-OEM video processing board to receive video from MIPI camera modules.

1.1 Additional Support Documentation

Additional Engineering Application Notes (EANs) can be found on the Documentation page of the SightLine Applications website.

The Panel Plus User Guide provides a complete overview of settings and dialog windows located in the Help menu of the Panel Plus application.

The Interface Command and Control (IDD) describes the native communications protocol used by the SightLine Applications product line. The IDD is also available as a PDF download on the Documentation page under Software Support Documentation.

1.2 SightLine Software Requirements

4000-OEM: Version 3.00.xx and higher.

ⓘ **IMPORTANT:** The Panel Plus software version should match the firmware version running on the board. Firmware and Panel Plus software versions are available on the Software Download page.

2 Optional Interface Boards and Adapters

Additional system interface boards provide options for network interfacing, serial ports, and GPIO. Camera interface and adapter boards provide an interface from the camera to OEM. See the ICD-3000-4000 Adapter Boards for complete specifications and pinouts.

ⓘ **IMPORTANT:** All boards should be connected and secured with the included cables and hardware fasteners first before applying power.

3 Hardware Connections

3.1 4000-OEM MIPI Camera Bench Setup

Parts listed in Table 1 are included in the 4000-OEM MIPI camera kit. If your application calls for other options and interface boards, please contact Sales.

Table 1: SLA-KIT-4000-MIPI-CAM

Part Number	Qty	Description
SLA-4000-PI22	1	Ribbon interface board. Converts 30-pin OEM interface to a 22-pin camera interface.
SLA-CAB-MIPI-02	1	Cable, FFC MIPI 28p 2-inches
SLHW-0001	1	Spacer
SLHW-0016	1	Stainless Steel Socket Head Screw, M2 x 0.4 mm thread, 8 mm Long

Interface and adapter boards:

- SLA-4000-PI22: Ribbon interface board. Converts 30-pin OEM interface to a 22-pin camera interface.

Camera to SLA-4000-PI22 board ribbon cable:

- The ribbon cable that connects the VC-MIPI-IMX412 camera to the SLA-4000-PI22 ribbon interface board must be purchased with the camera from Vision Components.
- The 15 to 22-pin ribbon cable (UC-376 or similar) that connects the Raspberry PI cameras to the SLA-4000-PI22 ribbon interface board must be purchased separately.

Common MIPI Cable connections:

- SLA-CAB-MIPI-02: Connects to J9 on 4000-OEM board and J1 on the SLA-4000-PI22 ribbon interface board. Provides serial communication and digital video to the camera.

SLA-CAB-MIPI-02 is an FFC cable and must be oriented and connected correctly for the camera to operate. See FFC cable instructions and precautions before connecting the cable.

- SLA-CAB-0403: Connects to J4 on the 4000-OEM board. Provides an RJ45 Ethernet connection.
- SLA-CAB-1504 / SLA-PWR-B12V-36W (110-250VAC input / 12VDC output): Connects to J50 on the 4000-OEM board.

Power and network connectivity LEDs:

A green light (D1) on the 4000-OEM board indicates that all boards are powered on. An amber light (D5) verifies network connection.

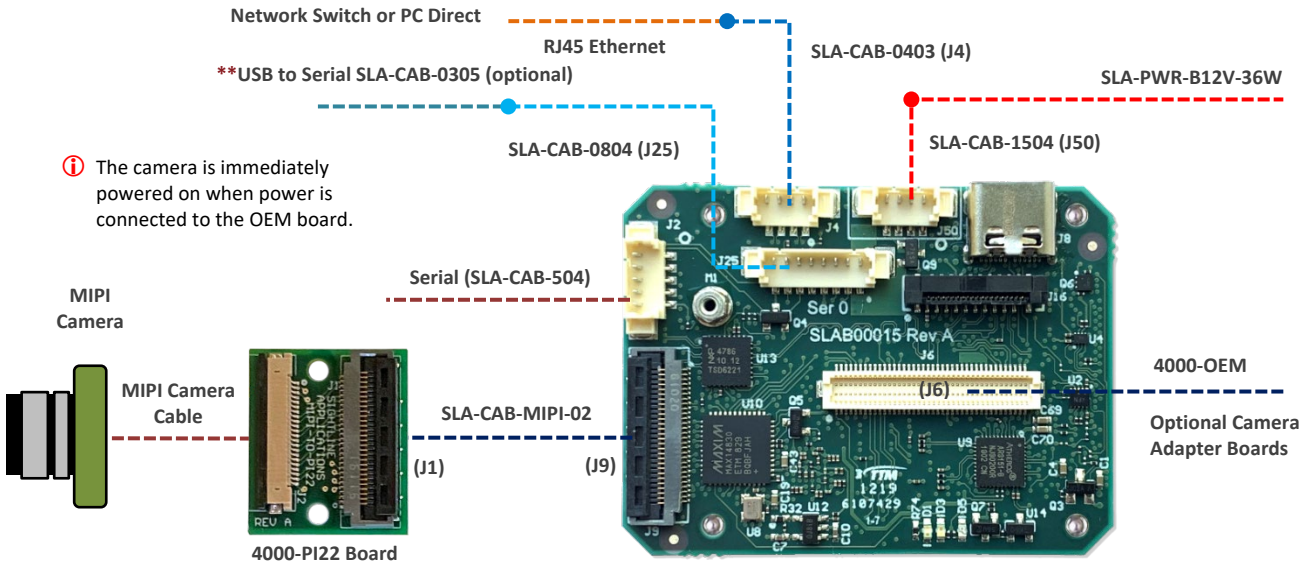


Figure 1: 4000-OEM MIPI Camera Bench Setup

**SLA-CAB-MIPI-02 FFC cable must be connected correctly. See FFC cable instructions before connecting the SLA-4000-MIPI board.*

***SLA-CAB-0305 can connect to SLA-CAB-0804 to facilitate a PC/USB connection to serial port 0 on the 4000-OEM.*

4 Configuration Settings

This section covers the basic camera configuration settings in Panel Plus for the SightLine 4000-OEM video processing board.

Before connecting with the Panel Plus software, the OEM board should be powered up and connected through:

- a network switch or directly to the host PC (preferred) or,
- Direct serial connection (for troubleshooting or if a network connection cannot be established).

See the EAN-Startup Guide 4000-OEM for connection and video streaming instructions.

IMPORTANT: This procedure assumes that the customer has read the OEM startup guide and has a basic understanding of the following fundamentals:

- Completed a functional connection between the SightLine video processing board and Panel Plus application.
- Familiar with Panel Plus controls.
- Successfully streamed video in Panel Plus.

If you do not have a strong basic system setup and familiarity, we recommend reviewing the OEM startup guide and work with our support team to establish basic connection and streaming fundamentals.

4.1 Acquisition Settings

From the main menu in Panel Plus go to *Configure* » *Acquisition Settings*. If available, use the *Auto Fill* drop-down menu in the *Acquisition Settings* dialog to automatically populate the relevant fields with the correct settings.

The settings can also be manually entered as shown in the camera configuration tables.

For information about Acquisition fields in Panel Plus see EAN-Digital Video Configuration.

IMPORTANT: Save parameters and reset the board when changing parameters. Cycle system power when changing resolution.

If video does not display, check the hardware and camera connections, and then try saving and activating the settings again. Check the encoding settings on the Compress tab and review the network addresses for the destination video.

5 Exposure, Gain and White Balance

5.1 Manual Controls

The 4000-OEM uses values from user input for exposure (integration time), gain and white balance (red, green, and blue gains). These values can be adjusted from the *Capture* tab in the *Image Control section* of Panel Plus (Figure 2).

In the IDD, exposure (integration time), luma (analog gain), and red/green/blue (digital color gains) can be adjusted using the **Set ADC Parameter (0x18)** command.

5.1.1 Exposure

Sets the integration time of the sensor. A higher setting is better for noise, lower is better for fast motion.

If this setting is too high, it may result in motion blur with a moving camera.

5.1.2 Luma (Auto Gain)

This is a gain applied on the sensor before the ADC. Increase it to get a brighter image. 0 to 255 maps to the minimum to maximum gain available on the sensor.

5.1.3 Red / Green / Blue

These are digital gains applied in the sensor after the ADC. Make small adjustments of individual gains to achieve white balanced color. Increase them together to get a brighter image.

For lowest noise, keep *Red*/*Green*/*Blue* as small as possible and increase *Exposure* and *Luma* to get a brighter image. 0 to 255 maps to the minimum to maximum gain available on the sensor.

Specific image controls are only available for select cameras. Controls are also enabled or disabled depending on the Auto Gain control setting (see Automatic Exposure and Gain).

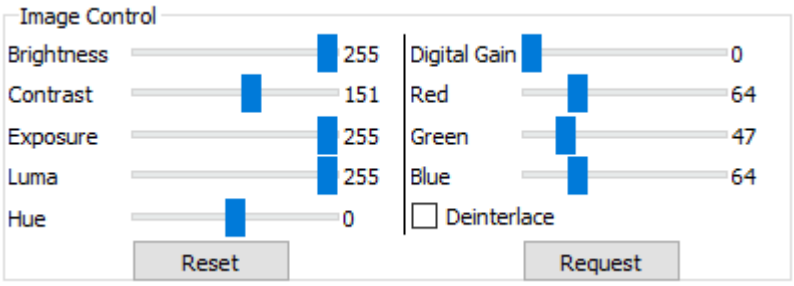


Figure 2: Image Control - Panel Plus Capture Tab

5.2 Automatic Exposure and Gain

New in 3.2.x software. Automatic exposure and gain controls (*Auto Gain*) are on the *Capture* tab in Panel Plus. Auto Gain is only available for high bit depth mono EO or IR cameras and for some MIPI color and mono cameras.

The system calculates the mean value over a specified ROI and then adjusts exposure and gains to move the mean towards the target value, which is specified in the *Brightness* control (Figure 2).

The Brightness control sets the target value for how bright the scene should be.

Color Sensor Auto Gain: Select this option in the drop-down menu to enable this feature. Use the settings in the *Image Control* section for manual exposure and gain control.

Auto Gain Region of Interest: Sets the region that auto gain is calculated.

Auto Min Range: The system will stop changing if the mean is within this value of target. A small non-zero value can reduce the auto gain solution from bouncing around. This value should be 10 or less.

Auto Rate: This is the auto gain step size. Higher values result in faster response. If the value is too high, it can make the system flash or bounce between solutions. The default is 32.

Min/Max Exposure: Limits the exposure time. High exposure values may result in image blur. Low exposure values may give a dark or high-noise image.

New in 3.4.x software:

Auto Reject Dark Tail: Reject or ignore a percentage of the left/dark side of the histogram when calculating autogain. 0 = reject none, 10 = reject 1.0%, 255 = reject 25.5%. This is useful when there are a small number of very dark pixels in a scene.

Auto Reject Bright Tail: Reject or ignore a percentage of the right/bright side of the histogram when calculating autogain. 0 = reject none, 10 = reject 1.0%, 255 = reject 25.5%. This is useful when there are a small number of very bright pixels in a scene.

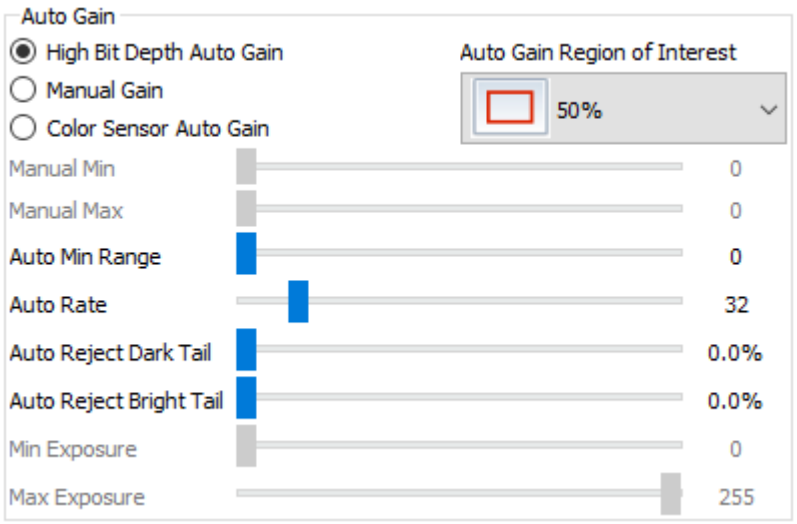


Figure 3: Auto Gain Controls - Panel Plus Capture Tab

Auto gain algorithm notes:

- For best image quality at a given brightness, controls are changed in a specific order to attempt to maximize exposure (subject to limits) and minimize digital gain.
- To make a scene brighter first increase the exposure. If the exposure is at maximum increase *Luma* (analog gain). If *Luma* is at maximum, increase the digital red, green, and blue gains. Auto gain moves the RGB gains together, increasing or decreasing them by equal amounts.
- To make a scene darker, first reduce *Digital Gain*, then reduce *Luma*, and then reduce *Exposure*.

6 Vision Components VC-MIPI-IMX412 Camera

See the Hardware Connections section for camera connection information.

6.1 Camera Configuration

The IMX412 camera is a 4056 wide by 3040 high 10-bit Bayer imager that can use two or four MIPI data lanes (Figure 4).

The Auto Fill option is only for two MIPI data lanes in Software 3.00.xx.

Supported display resolutions are full 4K (3840x2160), 1080p, and 720p. Big snapshot supported resolutions are 4056x3040, 4K, 1080p, and 720p.

Software 3.01.xx and above have a higher frame rate than software version 3.00.xx.

Software 3.2.x and above uses four MIPI data lanes for the default *Auto Fill* option. To set two MIPI data lanes use the *mipi=imx412_2lane* option in *Acquisition Settings* dialog shown in Figure 4.

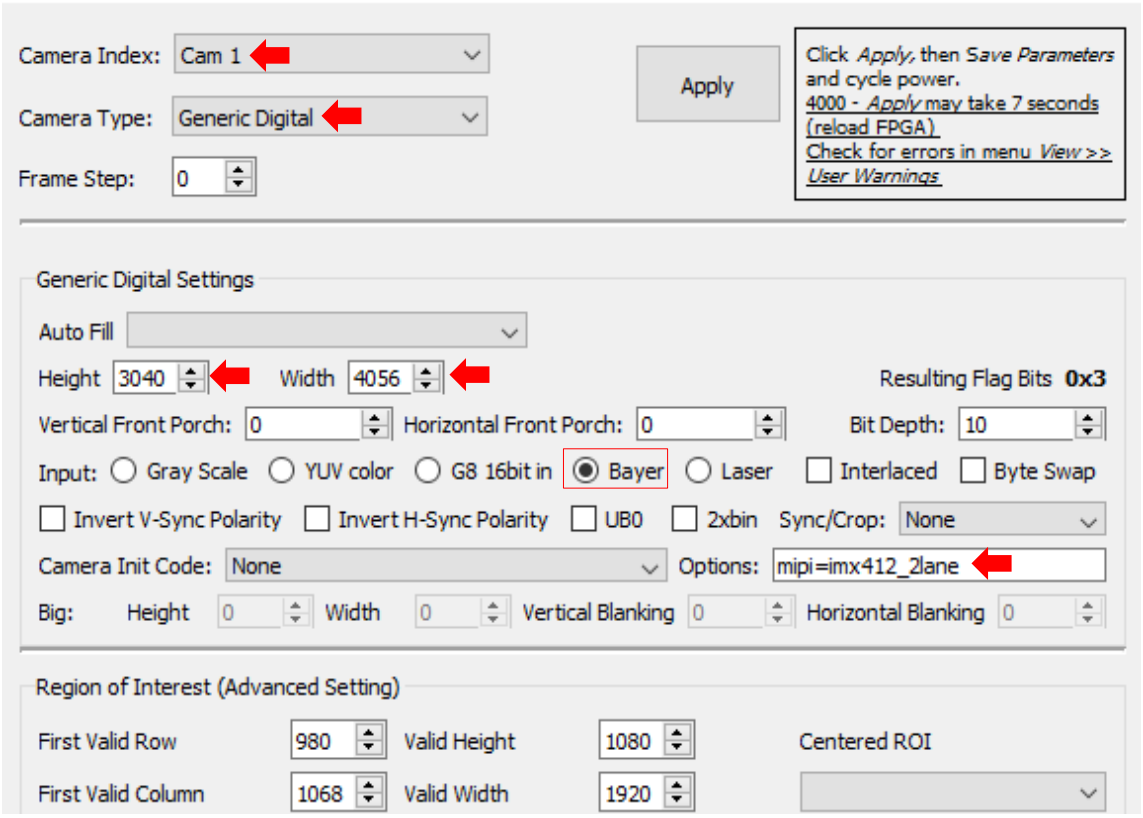
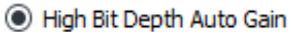


Figure 4: Acquisitions Settings - 3.2.x Software - Two MIPI Data Lanes - 4056x3040 Resolution

6.1.1 High Bit Depth Auto Gain

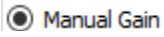


Manually adjusting the controls allows for better low light imaging. It also uses a 10 to 8-bit scaling and auto gain solution.

Exposure: Value 0 maps to 8 lines of exposure or 0.045 ms. 255 maps to 4088 lines of exposure or 22.0 ms.

Luma (analog gain): Value 0 maps to 0 gain, 255 maps to 978.

6.1.2 Manual Gain



Manually adjust *Exposure*, *Luma* (analog gain) and *Digital Gain* controls. When switching from *Color Sensor Auto Gain* to *Manual Gain* mode, the image should look the same.

6.1.3 Color Sensor Auto Gain

Color Sensor Auto Gain

The 4000-OEM supports automatic exposure and gain control for the Vision Component IMX412 camera by automatically controlling the exposure and gains (digital and analog). The automatic exposure and gain are described in the Exposure, Gain and White Balance section. For the IMX412, the camera has separate digital gains for red, green, and blue.

6.2 2x Binning Support

This example shows a 4K image that is 2x binned vertically and horizontally to 1080p.

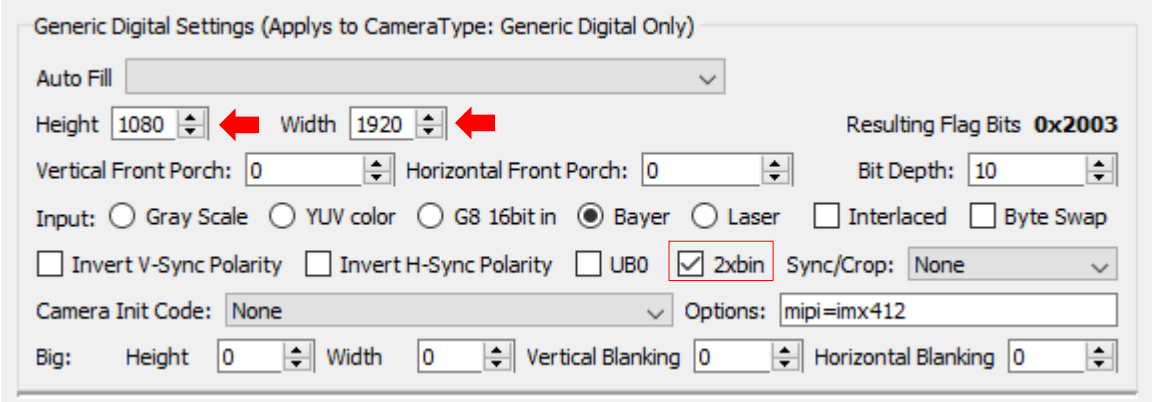


Figure 5: Acquisition Settings - 2x Binning Support

The image represented by the red box shown in Figure 6 is taken out of the full image. It is 1920*2 (width) and 1080*2 (height) and binned (downsampled) by 2 to produce a 1920x1080 output image.

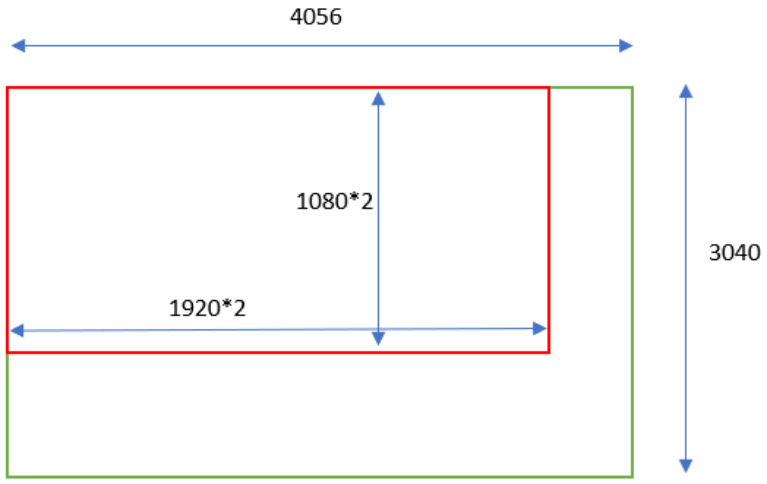


Figure 6: Downsampled 1920 x 1080 Image

6.2.1 Hardware Crop - 2x Vertical / Horizontal Binning

Images can also be cropped from the active image area using the *Hardware Crop* function. In Acquisitions Settings dialog set the *Horizontal Front Porch* and *Vertical Front Porch* values, and then select *Hardware Crop* in the *Sync/Crop* drop-down menu.

Horizontal Front Porch (HFP) and Vertical Front Porch (VFP) are applied to the full image as a starting point to the start of the image to be binned.

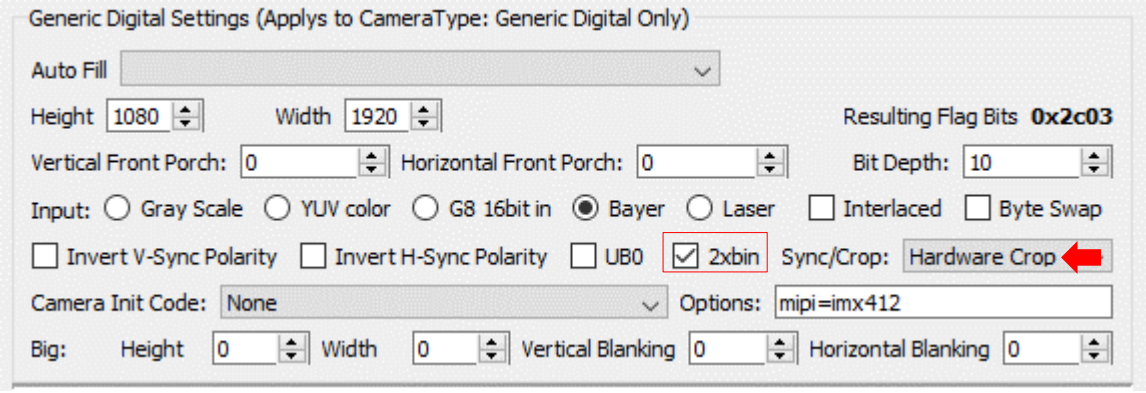


Figure 7: Acquisition Settings - Hardware Crop - 2x Vertical / Horizontal Binning

The image represented by the red box shown in Figure 8 is the image that will be binned (downsampled) by 2 in the vertical and horizontal to get a 1920x1080 image.

The maximum HFP in the above image is 4056 - (1920*2) = 216.

The maximum VFP in the above image is 3040 - (1080*2) = 880.

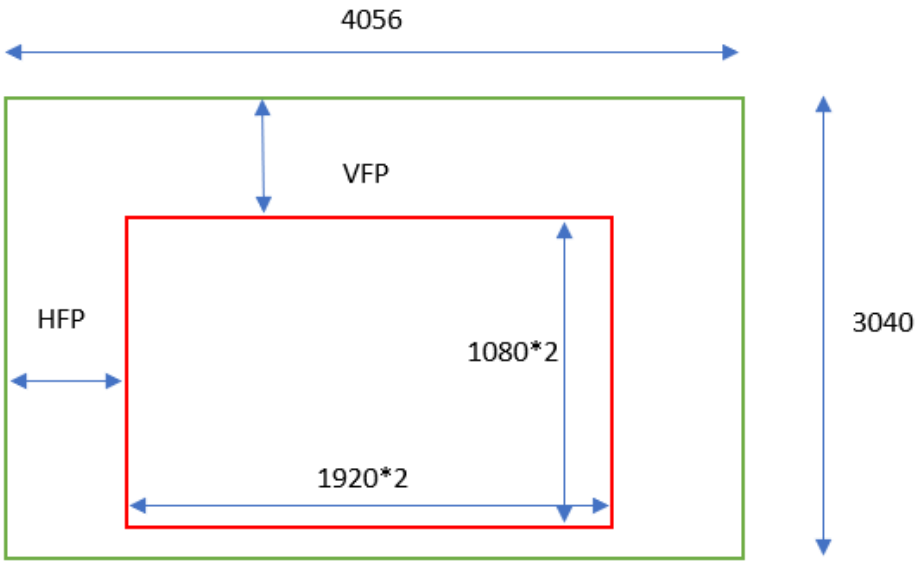


Figure 8: Downsampled 1920 x 1080 Image

6.3 Hardware Crop - 4056 x 3040 Image

The example in Figure 10 shows a 1080p image cropped from the center of the 4056x3040 image using the following Panel Plus *Acquisition Settings* shown in Figure 9.

Horizontal Front Porch: (4056-1920) /2 = *1068*

Vertical Front Porch: (3040-1080)/2 = *980*

Syn/Crop: Hardware Crop

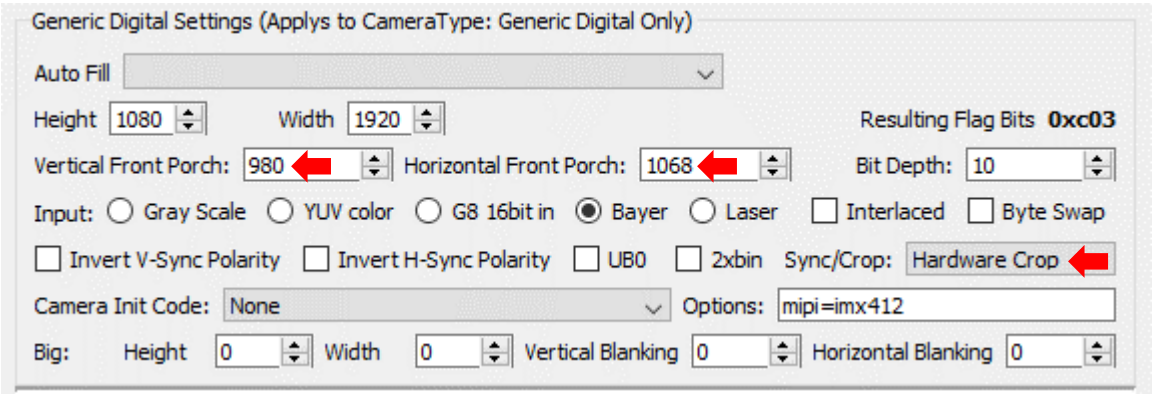


Figure 9: Acquisition Settings - Hardware Crop - 4056 x 3040 Image

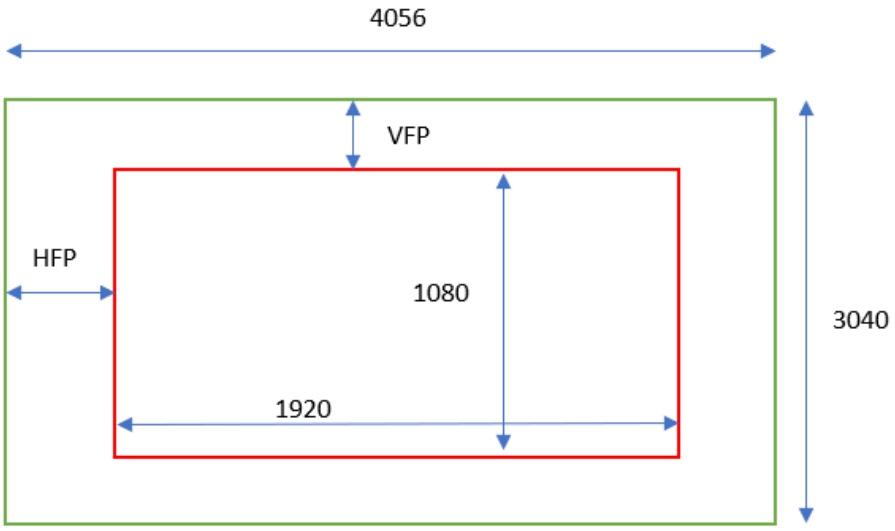


Figure 10: 1080p Image Cropped from 4056 x 3040 Image

6.4 I²C Communication

MIPI cameras use the **I2C Command (0x94)** to communicate with the attached sensor. Use the *Send Raw Message* dialog window to set a register in the sensor. Panel Plus main menu » *File* » *Send Command*.

For example, to set a register 0x0204 to a value of 0x01, send this command as shown in Figure 11.

The address of the IMX412 is 0x1a. Contact Sony to get a list of register descriptions and settings.

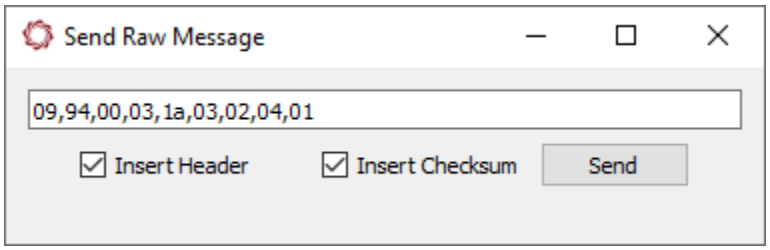


Figure 11: I²C Sensor Communication - Gain

It is possible to use Linux commands to read a register on the sensor from the command line, but this is not a reliable process.

7 Raspberry Pi V2.0 IMX219 Camera

See the Hardware Connections section for camera connection information.

7.1 Camera Configuration

The Raspberry Pi V2.0 uses a 10- to 8-bit scaling and auto gain solution.

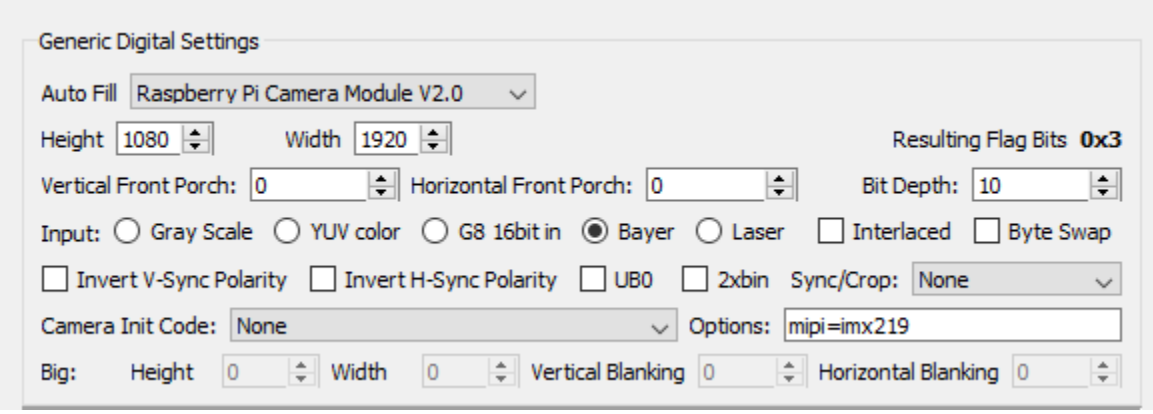


Figure 12: Acquisitions Settings - Pi V2.0 IMX219 Camera

7.1.1 High Bit Depth Auto Gain

From the *Image Control* dialog adjust *Exposure*, *Brightness*, Contrast, Luma, *Digital Gain, Red*, *Green*, and *Blue*. Adjust *Red*, *Green*, and *Blue* first before switching to *Automatic Exposure and Gain*.

7.1.2 Manual Gain

From the *Image Control* dialog adjust *Exposure*, *Luma*, *Digital Gain, Red*, *Green*, and *Blue*.

Adjust *Red*, *Green*, and *Blue* first before switching to *Automatic Exposure and Gain.*

Switching from High Bit Depth Auto Gain or Color Sensor Auto Gain initially has the same image.

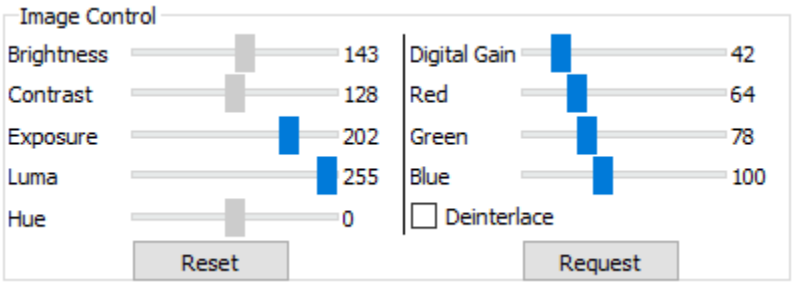


Figure 13: Image Control

Exposure: Value 0 maps to 8 lines of exposure, 255 maps to 1665 lines of exposure.

Luma (analog gain): Value 0 maps to 0 gain, 255 maps to 232.

7.1.3 Color Sensor Auto Gain

The 4000-OEM supports automatic exposure and gain control for the Raspberry Pi V2.0 camera by automatically controlling the exposure and gains (digital and analog). See the Exposure, Gain and White Balance section for more information.

The Raspberry Pi V2.0 automatic exposure and gain does a 10-bit to 8-bit conversion. The algorithm does an initial adjustment based on the *Red*, *Green* and *Blue* values in *Image Control*. It then adjusts the exposure, and gains (analog and digital).

The Raspberry Pi V2.0 has a global digital gain.

7.2 Full Size Acquisition Settings

The Raspberry Pi V2.0 is capable of a 3280x2464 acquisition size. The frame rate is 15 fps at this size.

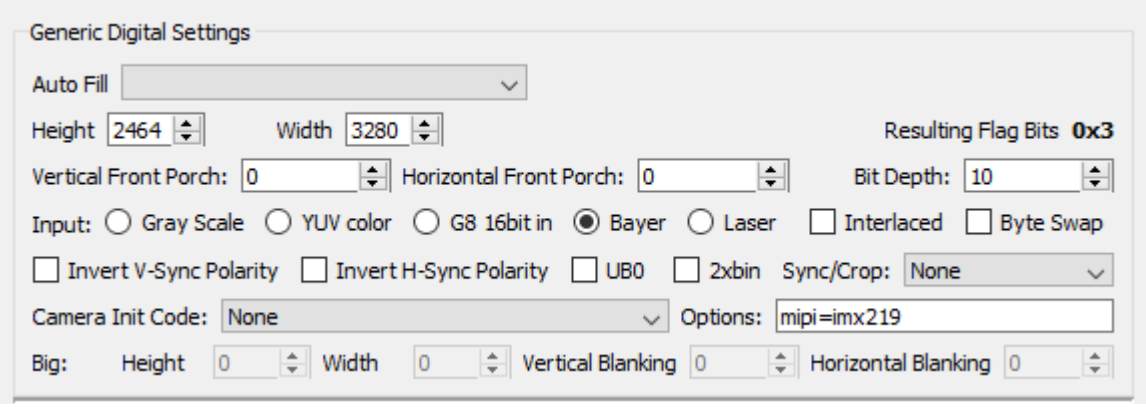


Figure 14: Acquisitions Settings - Pi V2.0 IMX219 Camera

8 Raspberry Pi V1.0 OV5647 Camera

See the Hardware Connections section for camera connection information.

8.1 Camera Configuration

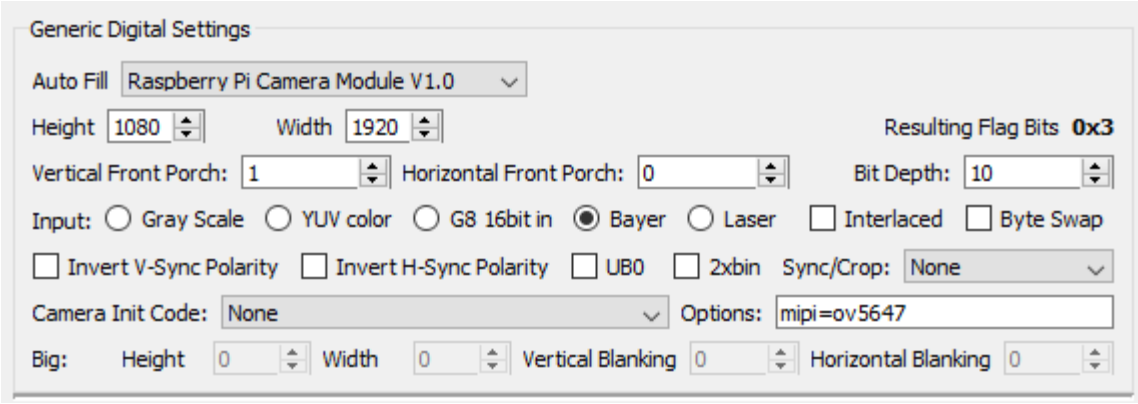


Figure 15: Acquisitions Settings - Pi V1.0 OV5647 Camera

8.1.1 Manual Gain/ High Bit Depth Auto Gain

From the *Image Control* dialog adjust *Exposure* and *Luma* (analog gain).

Exposure: Value 0 maps to 8 lines of exposure, 255 maps to 12248 integration time.

Luma (analog gain): Value 0 maps to 0 gain, 255 maps to 1023.

8.1.2 Color Sensor Auto Gain

The 4000-OEM supports automatic exposure and gain control for the Raspberry Pi V1.0 camera by automatically controlling the exposure and luma (analog gain). See previous section on Exposure, Gain and White Balance for more information.

8.2 Full Size Acquisition Settings

The Raspberry Pi V1.0 is capable of a 2592x1944 acquisition size. The frame rate is 15 fps at this size.

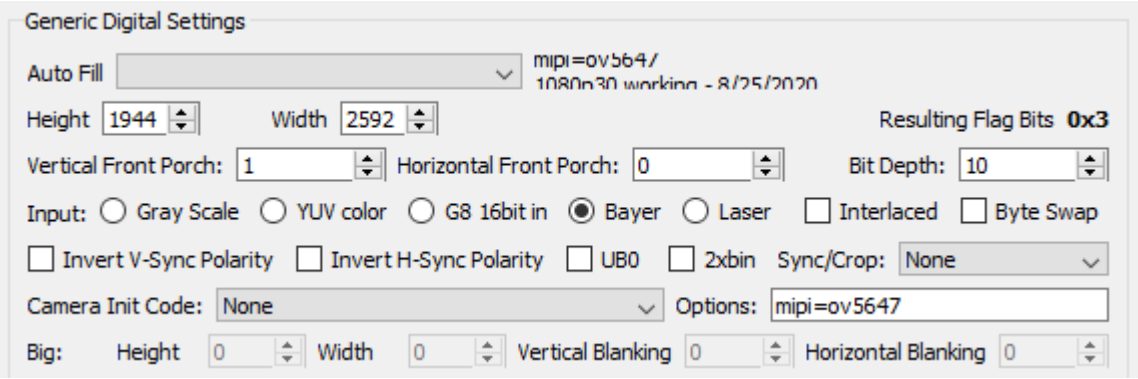


Figure 16: Acquisitions Settings - Pi V1.0 OV5647 Camera

9 Vision Components VC-MIPI-IMX296 Mono Camera

See the Hardware Connections section for camera connection information.

9.1 Camera Configuration

The IMX296 camera is a 1440 wide by 1080 high 10-bit Monochrome imager.

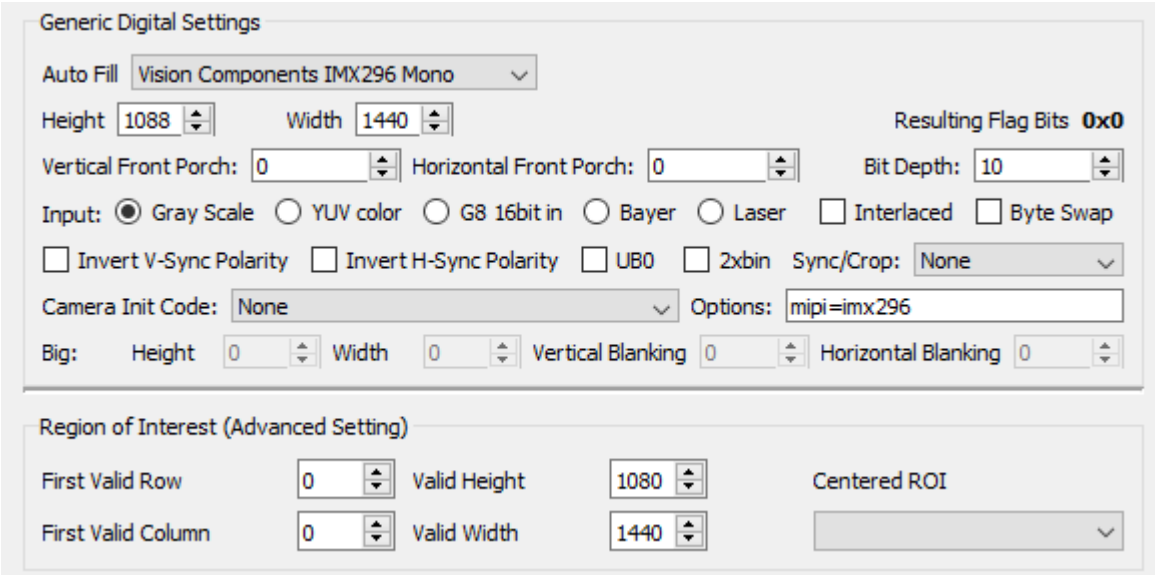


Figure 17: Acquisitions Settings - Vision Components IMX296 Camera

10 Troubleshooting

10.1 Determining Settle-cnt

In the 4000-OEM FPGA, a *Ths-Prepare* and *THS-Zero* are set. Each MIPI camera has a *Ths-Settle*. The *Ths-Settle* (settle-cnt) is calculated in the 4000-OEM using the camera driver setting for *V4L2_CID_PIXEL_RATE*.

If captured video looks incorrect or not capturing at all, adjust the *V4L2_CID_PIXEL_RATE* in the camera driver so that a different settle-cnt is calculated.

10.2 Questions and Additional Support

For questions and additional support, please contact Support. Additional support documentation and Engineering Application Notes (EANs) can be found on the Documentation page of the SightLine Applications website.

Appendix A - MIPI Camera Requirements for Connecting Commercial Cameras

Every commercial MIPI camera requires a Linux driver to work with any type of acquisition system. Camera registers are vendor and model specific and must be configured over the MIPI I^2C bus.

Each camera has different operating characteristics and requirements for proper operation. Currently a standard interface for MIPI camera register setup and operations does not exist.

For new MIPI cameras working through configuration issues takes time. Reusing a driver code from another camera in most cases does not work.

It is essential to address the Linux driver availability for new MIPI camera early in the process.

A1 Snapdragon 820 MIPI CSI-2 Version Support

MIPI support for the Snapdragon processor is important since the selected camera must output packet types that are supported in the hardware.

Supported Version: MIPI Alliance Specification for CSI-2 v1.3

Reference: *Qualcomm® Snapdragon™ 820E Processor (APQ8096SGE) Device Specification LM80-P2751-1 Rev. E February 9, 2018*

A2 Vendor Provided Driver / Interface

The vendor provided driver/interface does not use Snapdragon 820 ISP. A Linux driver must be provided.

- Linux driver with I^2C register settings. Helpful when setting up the camera and turning on streaming.
- Linux driver settings must be for raw capture. The 4000-OEM only supports raw capture.
- Linux driver to include *V4L2_CID_LINK_FREQ* and *V4L2_CID_PIXEL_RATE* settings.
- SightLine recommends the device tree source file (overlay) for the DragonBoard 820c. This shows data lanes, clock lanes, clock settings, and the I^2C address of the sensor.
- Four data lanes are available at the J9 connector on the 4000-OEM.

- If available, a datasheet and register explanation for the camera.
- Maximum of 750MHz pixel clock frequency (1.5Gbps / lane).
- SightLine recommends the sensor have an ISP (Sony IMX sensors do not).

A raw capture using a sensor that does not have ISP will appear substandard. This would require additional color 10-bit to 8-bit conversion using auto gain, brightness, and white balance.

- Format of the image - width, height, media bus format.

Linux and board compatibility notes:

- DragonBoard 820c: 96Boards compliant board based on the Snapdragon 820 processor.
- Smart Wireless Computing SOM: Used on the 4000-OEM (IFC6601 SOM).
- 96Boards: Linaro kernel / images (DB410C / DB820C Debian, OE)

A3 Raw Capture Only

Raw capture does not use Snapdragon 820 ISP.

The following are supported media bus formats for the Snapdragon:

YUYV/UYVY/YVYU/VYUY (packed YUV 4:2:2):

- V4L2_PIX_FMT_YUYV
- V4L2_PIX_FMT_UYVY (supported)
- V4L2_PIX_FMT_YVYU
- V4L2_PIX_FMT_VYUY

MIPI RAW8 - 8-bit Bayer (raw):

- V4L2_PIX_FMT_SRGGB8
- V4L2_PIX_FMT_SGRBG8
- V4L2_PIX_FMT_SGBRG8
- V4L2_PIX_FMT_SBGGR8

MIPI RAW10 - 10-bit packed Bayer (raw):

- V4L2_PIX_FMT_SBGGR10P
- V4L2_PIX_FMT_SGBRG10P
- V4L2_PIX_FMT_SGRBG10P
- V4L2_PIX_FMT_SRGGB10P (supported)

MIPI RAW12 - 12-bit packed Bayer (raw):

- V4L2_PIX_FMT_SRGGB12P
- V4L2_PIX_FMT_SGBRG12P
- V4L2_PIX_FMT_SGRBG12P
- V4L2_PIX_FMT_SRGGB12P

The 4000-SOM currently has support for V4L2_PIX_FMT_UYVY and V4L2_PIX_FMT_SRGGB10P media bus formats.

A4 Connector Requirements

- Four data lanes.
- One clock lane.
- Reset and standby for camera control in Linux driver.
- I²C lines for setting register in the camera, used by the Linux driver.

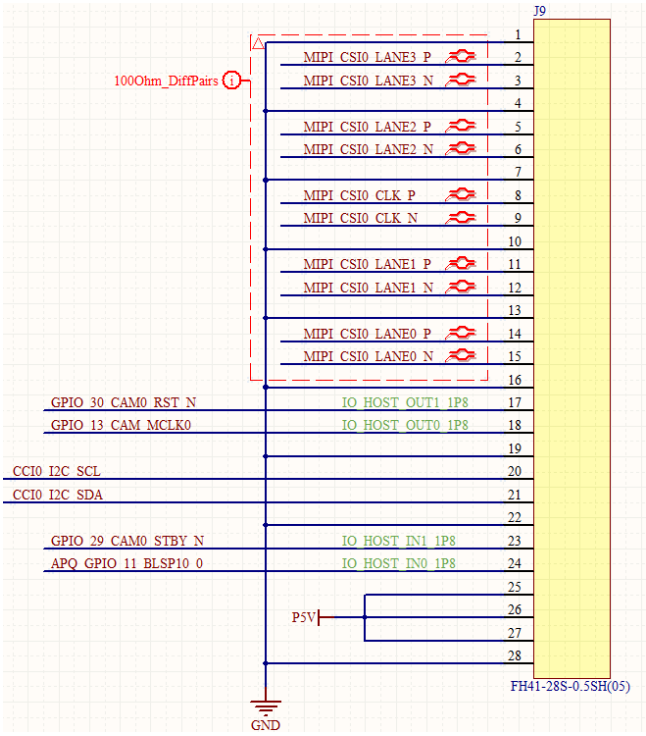


Figure A1: MIPI Connector Pinout

Appendix B - MIPI Camera Capture Requirements - Custom MIPI Inputs

This section covers requirements for customers that externally convert camera data into MIPI signals input to the 4000-OEM. The system consists of a camera, acquisition system, and an FPGA that generates MIPI signals.

See the Customer Designed 4000-OEM Boards and Camera Interface Options section in the *ICD-4000-OEM for more information on implementing a design with external FPGAs to interface with the 4000-OEM.*

The MIPI camera interface consists of a clock signal and 1, 2 or 4 data lanes. The number of data lanes depends on the camera frame rate and resolution. The MIPI standard also provides for an I²C bus for camera configuration that may not be used in customer FPGA implementations.

IMPORTANT :

- MIPI packet support: The MIPI support for the Snapdragon process is important since the FPGA selected must output packet types that are supported in SightLine hardware.
- Supported Version: MIPI Alliance Specification for CSI-2 v1.3 Reference: Qualcomm® Snapdragon™ 820E Processor (APQ8096SGE) Device Specification LM80-P2751-1 Rev. E February 9, 2018.

ⓘ **IMPORTANT:** When designing a custom MIPI (FPGA) camera board, it is important to include a method to externally reset the FPGA and camera state machine. The commercial SightLine supported MIPI cameras utilize the I^2C bus over MIPI and a Linux driver that allows for a camera reset at the appropriate times. Design examples may include an external connection to a 4000-OEM GPIO pin or an I^2C to GPIO/serial bridge chip similar to the design on the 4000-MIPI adapter board.

MIPI was designed to support cell phone cameras, which are color cameras in 8, 10, and 12 bits. There is no support in the packet types for 14-bit IR grayscale cameras or 16-bit YCbCr cameras.

The 4000-OEM uses the MIPI as a transport layer. The data packets are received by the processor and treated as bytes. These bytes are then assumed to be in the format set in the acquisition parameters persisted on the OEM. The next describe the MIPI packet format used to transport video data from different supported cameras.

All MIPI data elements are 8-bits. Since multiple pixel values are grouped together to create data that is divisible by 8, 10-bit color video will have a group containing the upper 8-bits of 4 pixels, followed by a single 8-bit value that contains the lower 2-bits of the 4-pixel group. This results in 5x8-bit values to represent 4x10 bit pixels. SightLine camera support uses a full 16-bits (into 2x8) for camera pixel values that are 14-bits. This makes the encoding simpler and there is enough bandwidth in the MIPI bus.

B1 MIPI Packet Format

Table B1: MIPI Packet Formats

Camera Format	MIPI Type	ID,Lanes,Gear	Pkt Width	Capture Setting	Note
14/16-bit Grayscale	YUV422_8	0x1E, 2, 8	x2	Gray, bits=16	IR Camera
8-bit Grayscale	YUV422_8	0x1E, 2, 8	x1	Gray, bits=8	IR Camera
16-bit YCbCr (HD)	YUV422_8	0x1E, 2, 8	x2	YUV, bits=16	BT.1120
8-bit YCbCr (SD)	YUV422_8	0x1E, 2, 8	x2	YUV, bits=8	BT.656, NTSC/PAL
8-bit Bayer	YUV422_8	0x1E, 2, 8	x1	Bayer, bits=8	AGS720P Color

The following describes how the current Sightline CrossLink FPGA supports various camera formats:

- YUV422_8 packet. The packet width is optionally doubled to support 2x8-bit (16-bit data).
  - ⓘ Important – this requires doubling the pixel clock rate to output 2x8 bit packets for each 16 bit pixel value
- 16-bit data order is low byte, then high byte (little endian). We recommend providing for byte-swap in custom FPGAs.
- Number of MIPI Tx Lanes = 1,2, or 4 – the minimum MIPI bit clock rate is 40 MHz and this will drive the number of lanes required for a specific camera.
- The data packets are received by the OEM processor and treated as bytes. They are then assumed to be in the format set in the acquisition parameters persisted on the OEM (G16, YCbCr16,G8).
- The pixel width entered in the acquisition parameters setting is not doubled but is inferred by the other capture parameters (bits=16/8).

- The number of rows is set by the number of packets sent between FS (Frame Start) and FE (Frame End). Each packet is a row of data.
- Normally there is no blanking in the MIPI data, only active pixels are included. OEM system processing provides ROI (Region of Interest) within a larger frame if you want to include blanking in your packets. However, this adds capture overhead and reduces processing rates on the system.
- For a diagram of CSI-2 packets see page 17 in the Camera Serial Interface CSI-2 and CSI-3 document from MIPI Alliance.

B2 MIPI Packet Timing

The following parameters will depend on the pixel clock rate of the camera:

ⓘ **IMPORTANT:** The high speed (HS) clock mode may require a new release of 4000-OEM firmware. Contact Support for details.

- Low power states are used between short and long packets. See the CSI-2 packets in the previous section. This means that timing during the transition from low power to high-speed mode needs to be specified. This is dependent on the pixel clock of the camera and therefore the MIPI Tx clock.
- This timing is covered by *T-HS-PREPARE and T-HS-ZERO*:
  - The timing diagram can be referenced in page 3 in the Tektronix App note on MIPI timing.
  - The timing parameters should be specified by the documentation in the MIPI IP package used to generate the MIPI packets. This will be dependent on the camera pixel clock.
  - Figure B1 shows how a scope trace can be used to measure and compare values as the FPGA code is being developed. The trace in this example was taken off the N phase of Tx Lane 0 (the P phase looks different).

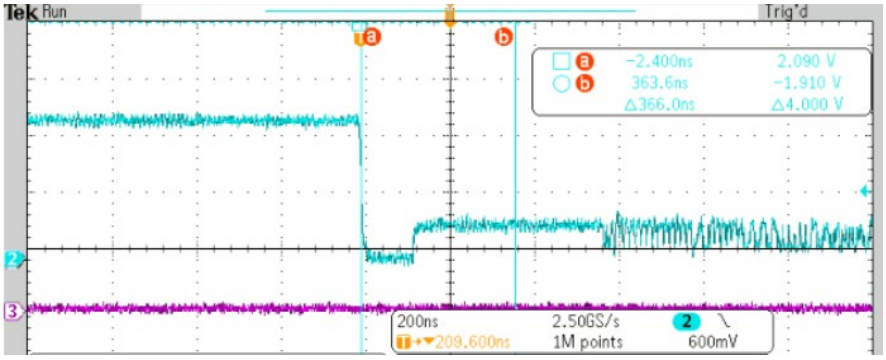


Figure B1: MIPI Packet Low Power to High-Speed Transition

B3 Acquisition Settings for MIPI Configuration - Tx Lanes and MIPI Clock Rate

The 4000-OEM needs MIPI transmit information to be able to detect and capture the MIPI packets.

The following new settings shown in Table B2 are available in the 3.2.x software release. Settings are set as comma separated values (CSV) in the *Options* field in the *Acquisition Settings* dialog.

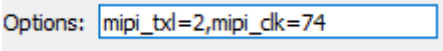


Figure B2: Options Field - MIPI Acquisition Settings

Table B2: MIPI Acquisition Settings

MIPI Settings	Setting Notes
mipi_txl=1	• The number of Tx lanes. • Valid values are 1, 2, and 4. • Requires a parameter save and board reset. • *Changing the number of Tx Lanes will update the TSettle count automatically.*
mipi_clk=74	• MIPI clock rate in MHz. This will vary based on camera input pixel clock. • This affects both the MIPI CSI frequency and the TSettle count. • The CSI frequency is comparable to a sampling rate, and the TSettle count the number of samples where it looks for the packet. The CSI frequency changes in discrete intervals. • The CSI frequency will be 100 MHz, 200 MHz, or 266 MHz. • Used when the 4000-MIPI cannot acquire packets from a camera source due to packet. timing incompatibility with default driver settings. • Requires a parameter save and board reset.

The parameter values shown in Table B3 and their effects can be seen in the debug port (Serial 0) output of the 4000-OEM. Accessing the debug serial port requires the use of the 4000-DEBUG board. See the ICD-4000-OEM for details.

The debug serial port output defaults to 115,200 baud.

The *CSI Frequency* is shown in the debug output line:

```
Setting csi frequency=100000000 pixel_clock=74000000
```

The *pixel_clock* value will be the clock value set in *mipi_clk*.

The *TSettle Count* and number of *Tx Lanes* are shown in the debug output line:

```
pixel_clock=74000000 settle_cnt = 4 lane=0x83
```

1 Lane = 0x81, 2 Lane = 0x83, 4 Lane = 0x8F

Table B3: MIPI CSI Frequency and TSettle Values

Pixel Clock	Tx Lanes	CSI Frequency	TSettle Count
148.5	2	200	12
74.25	2	100	4
27	2	100	8
20	2	100	10

B4 FPGA I2C Control of Camera Acquisition (Optional)

The section describes how to control camera acquisition through I^2C registers for custom designed FPGAs. Although not required, implementing an I^2C register interface may make it a more flexible design.

The 4000-OEM FPGA supports programmable camera timing parameters. This allows using the same FPGA code to support multiple camera types and resolutions. The camera timing parameters are updated from the 4000-OEM acquisition settings into the FPGA I^2C registers. This data is automatically updated through the MIPI I^2C bus to the FPGA.

Additionally, this requires implementing I²C interface code on your FPGA and connecting the appropriate pin for I²C to the MIPI connector. Generally, this is a data pin *sda* that supports open drain (and may be identified as an I²C port) and an I²C clock *scl*.

The I2C data line *sda* must be defined as a bidirectional port and pulled up in the FPGA pin configuration. The open drain operation can be implemented in Verilog as

```
assign sda = (sdaOut == 1'b0) ? 1'b0 : 1'bz;
```

The *scl* I²C clock line can also be defined as bidirectional open drain, but this is not necessary since the SightLine FPGA device does not do clock stretching.

Successful results have been achieved using Open Cores I²C implementation (see the OpenCores website). This implementation requires a clock internal to the FPGA (the SightLine FPGA provides an internal 48 Mhz reference clock) for sampling (and debouncing) the I²C clock and data lines.

The 4000-OEM can also reset the FPGA state machines (synchronized to frame sync). It does this through an I²C register. This may not be necessary in your design.

There are two available MIPI ports on the 4000-OEM. Each port has a different I²C address. The I²C bus speed is limited to 400 kb due to incompatibility with the SLA-3000-HDSDI-IN input board.

The FPGA interface is assumed to reside at I²C address 0x40 (7-bit I²C address). This means that the upper 7-bits are 0x40 and the lower bit is 0 or 1 depending on the Read/Write attribute. Since the slave address is 0x40, the 8-bit write address is 0x80 and the 8-bit read address is 0x81.

CSI2 supports I²C Bus 1 (Cam 0). CSI0 supports I²C Bus 3 (Cam 1) On the SOM Port J9.

All register values are 16-bit, little endian (low byte first).

Table B4: I²C Register Map

I²C Address	Description	Example
0x00 0x01	Version0 - read only. Intended for firmware version or version of register map as more registers are added or change order. Lower 3 nibbles are version. Upper nibble is register layout (0x0 = current).	0x001 = version 0x0 = register layout0 (used if this map of register values or size is changed)
0x02 0x03	Version1 - read only.	SVN revision. Used to track revisions.
0x04 0x05	FPGA Control and status. B0 - Frame sync reset. Set B0 to high to assert reset on FPGA logic (state machines), and then set B0 low to deassert reset. This will be deasserted on the next falling edge of VSync (before the start of the next frame) frame sync reset of FPGA logic.	Read / Modify / Write
0x06 0x07	Dig Cam Params 16-bit exact copy of digCamParams flags, e.g., Embedded sync, Invert HSync, Invert VSync, Byte Swap.	Hex Value
0x08	Vertical Front Porch - blanking lines at start of frame	0x0015 = 21 lines
0x09		
0x0 0x0B	Active Vertical Lines (height)	0x02D0 = 720
0x0C 0x0D	Horizontal Front Porch - blanking pixels at start of line	0x00E9 = 233
0x0E 0x0F	Active Horizontal Length - width in active pixels	0x00500 = 1280
0x011 0x010	MIPI clock estimate Mhz (B9->B2) and TxLanes (B1->B0) 00=def,01=1,10=2,11=4	0x0029 = 1 TxLane, 10 MHz

Example - Resetting the FPGA (for I²C bus 1 - J6 port):

Resetting the FPGA should be done programmatically with a read-modify-write to allow for other control bits to be added to this register in the future.

This only resets the internal FPGA registers and state machines (global reset). This does not reset the actual FPGA hardware and will not fix an FPGA program that will not load.

Assert reset: (video stops)

I2cset -f -y 1 0x44 1

Deassert reset: (Video starts synchronized with next camera Vertical Sync)

I2cset -f -y 1 0x44 0

Example - Register Read (for I²C bus 1 - J6 Port):

echo Version 0

i2cget -f -y 1 0x40 1

i2cget -f -y 1 0x40 0

echo Version 1

i2cget -f -y 1 0x40 3

i2cget -f -y 1 0x40 2

echo VFP

i2cget -f -y 1 0x40 0x9

i2cget -f -y 1 0x40 0x8

echo height

i2cget -f -y 1 0x40 0xB

i2cget -f -y 1 0x40 0xA

echo HFP

i2cget -f -y 1 0x40 0xD

i2cget -f -y 1 0x40 0xC

echo width

i2cget -f -y 1 0x40 0xF

i2cget -f -y 1 0x40 0xE