PRODUCT BRIEF

E SERIES

SV4E-DPRXG

MIPI D-PHY Frame Grabber

Any-Rate, Any-Resolution MIPI CSI-2 Sensor Calibration and Characterization Solution

The SV4E-DPRXG MIPI D-PHY Frame Grabber is a flexible solution for capturing and analyzing MIPI® Alliance CSI-2® sensor data. It can be attached to any CSI-2 camera output or radar output, and it will automatically extract image data and support automated application development, calibration, and regression testing. The SV4E-DPRXG's unique analog front-end technology for the MIPI Alliance D-PHY℠ physical layer means that users can achieve high-confidence sensor validation without worrying about physical attachment issues.

KEY FEATURES:

- **D-PHY Physical Layer:** Monolithic receiver with integrated LP/HS signaling and support for data rates up to 3.5 Gbps per lane.
- **CSI-2 Controller:** Support for all CSI-2 data types and pixel formats, including RAW16 and RAW20.
- **Virtual Channels:** Automatic extraction of all virtual channels supported by the CSI-2 standard.
- **I2C and I3C Master:** Dual-mode I2C/I3C master for controlling sensors and providing true host-emulation capability.
- **Diagnostics:** Built-in frame-rate and CRC monitors.

KEY BENEFITS:

- **Future-Proof:** High-performance receiver that is upgradable – within the same hardware – to include packet and protocol analysis.
- **Self-Contained:** An all-in-one system reduces bench space and helps create very compact regression farms.
- **Flexible:** Live streaming mode helps with manual sensor setup, and bulk capture mode helps with automation.
- **Automated:** Leverages the full power of Python and the award-winning Pinetree software.

Typical Application: CSI-2 D-PHY Sensor Calibration and Characterization

Simple Connection with DUT Directly Mounted

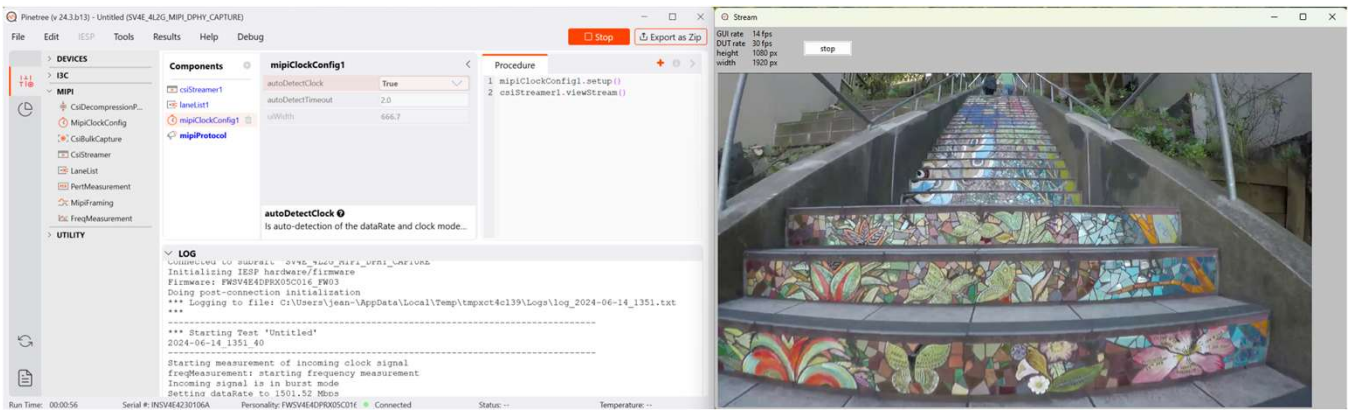
```
i2cWrite(slaveAddr, slaveReg=0x5412, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x01])

# SET DEFAULT EXPOSURE
i2cWrite(slaveAddr, slaveReg=0x4400, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
i2cWrite(slaveAddr, slaveReg=0x4401, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
i2cWrite(slaveAddr, slaveReg=0x4402, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])

# START STREAMER
mipiCphyCsiStreamer1.start()

# CAPTURE TEST PATTERN
print("TEST PATTERN ENABLE")
i2cWrite(slaveAddr, slaveReg=0x6E00, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x08])
sleepMillis(50)
testPatternFrame1 = mipiCphyCsiStreamer1.getFrame()
i2cWrite(slaveAddr, slaveReg=0x6E00, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x00])
sleepMillis(50)

# ENABLE VERTICAL AND HORIZONTAL FLIP
i2cWrite(slaveAddr, slaveReg=0x4820, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x10])
print("VERTICAL FLIP ON"); sleepMillis(100)
verticalFrameFlipOn = mipiCphyCsiStreamer1.getFrame()
i2cWrite(slaveAddr, slaveReg=0x4820, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x00])
print("VERTICAL FLIP OFF"); sleepMillis(100)
verticalFrameFlipOff = mipiCphyCsiStreamer1.getFrame()
i2cWrite(slaveAddr, slaveReg=0x4821, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
print("HORIZONTAL FLIP ON"); sleepMillis(100)
horizontalFrameFlipOn = mipiCphyCsiStreamer1.getFrame()
```

Highly Flexible Automation Environment

Key Performance Parameters

PARAMETER	VALUE	NOTES
Number of Lanes	4 Data + 1 Clock	Adjustable lane configuration; supports D-PHY v2.5 signaling
Symbol Rates	80 Msps – 3.5 Gbps	Supports a wide array of sensors
Minimum $\|V_{OD}\|$	140 mV	Measured at module connector
Maximum $\|V_{OD}\|$	300 mV	Measured at module connector
Minimum T_{LPX}	50 ns	
Minimum $T_{HS\text{-}PREPARE}+T_{HS\text{-}ZERO}$	145 ns + 10 UI	
Total Memory Space	1 GByte	Entire memory space is available for captured image storage
Number of Programmable Power Supplies	6	Independently controlled through Python scripting
Programmable Power Supply Parameters	1 V – 5 V in steps of 1 mV	Each power supply provides 3 A

Use the SV4E-DPRXG Frame Grabber with Pinetree to Stream CSI-2 Sensor Data



Dynamically Visualizing Sensor Response and Sensitivity During Live Streaming