

E SERIES

SV4E-CPRXG

MIPI C-PHY Frame Grabber



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

The SV4E-CPRXG MIPI C-PHY Frame Grabber is a flexible solution for capturing and analyzing MIPI® Alliance CSI-2SM sensor data. It can be attached to any CSI-2 camera output or radar output, and it will automatically extract image data and provide for automated application development, calibration, and regression testing.

The SV4E-CPRXG's unique analog front-end technology for the MIPI Alliance C-PHYSM physical layer means that users can achieve high-confidence sensor validation without worrying about physical attachment issues.

KEY FEATURES:

- **C-PHY Physical Layer:** monolithic receiver with integrated LP/HS signaling and support for symbol rates up to 3.5 Gbps
- **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 Introspect ESP Software

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



Simple Connection Scheme

```

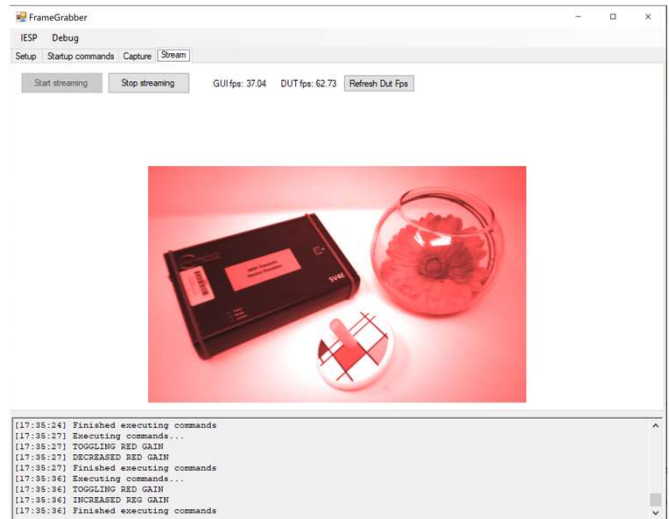
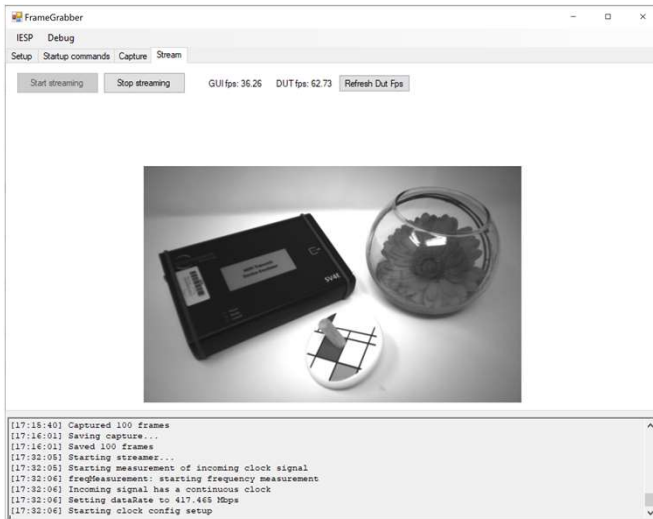
17 i2cWrite(slaveAddr, slaveReg=0x5412, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x01])
18
19 # SET DEFAULT EXPOSURE
20 i2cWrite(slaveAddr, slaveReg=0x4400, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
21 i2cWrite(slaveAddr, slaveReg=0x4401, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
22 i2cWrite(slaveAddr, slaveReg=0x4402, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
23
24 # START STREAMER
25 mipiCphyCsiStreamer1.start()
26
27 # CAPTURE TEST PATTERN
28 print("TEST PATTERN ENABLE")
29 i2cWrite(slaveAddr, slaveReg=0x6E00, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x08])
30 sleepMillis(50)
31 testPatternFrame1 = mipiCphyCsiStreamer1.getFrame()
32 i2cWrite(slaveAddr, slaveReg=0x6E00, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x00])
33 sleepMillis(50)
34
35 # ENABLE VERTICAL AND HORIZONTAL FLIP
36 i2cWrite(slaveAddr, slaveReg=0x4820, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x10])
37 print("VERTICAL FLIP ON"); sleepMillis(100)
38 verticalFrameFlipOn = mipiCphyCsiStreamer1.getFrame()
39 i2cWrite(slaveAddr, slaveReg=0x4820, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x00])
40 print("VERTICAL FLIP OFF"); sleepMillis(100)
41 verticalFrameFlipOff = mipiCphyCsiStreamer1.getFrame()
42 i2cWrite(slaveAddr, slaveReg=0x4821, regAddrLen=2, regAddrEndianness='big', msgBytes=[0x04])
43 print("HORIZONTAL FLIP ON"); sleepMillis(100)
44 horizontalFrameFlipOn = mipiCphyCsiStreamer1.getFrame()

```

Highly Flexible Automation Environment

Key Performance Parameters

PARAMETER	VALUE	NOTES
Number of Trios	3	Each trio consists of 3 wires and supports C-PHY v1.2 signaling
Symbol Rates	80 Msps – 3.5 Gsps	Supports a wide array of sensors
Minimum $ V_{OD} $ Weak	90 mV	Measured at module connector
Maximum $ V_{OD} $ Strong	300 mV	Measured at module connector
Minimum t_{LPX}	50 ns	
Minimum $t_{3-PREBEGIN}$	28 symbols	
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



Dynamically Visualizing Sensor Response and Sensitivity During Live Streaming