Getting Started with EZ-USB SX3

About this document

Scope and purpose

EZ-USB™ SX3 (hereafter abbreviated as SX3) is a configurable USB 3.2 Gen 1 SuperSpeed peripheral controller, providing integrated and flexible features. SX3 has a fully configurable, parallel interface called General Configurable Interface, which can connect to any ASIC, image signal processor (ISP), image sensor, or FPGA that supports Slave FIFO or Video interface. SX3 enables developers to easily add USB 3.0 functionality to their systems using the SX3 Configuration Utility.

This application note helps you get started with SX3. It highlights the key uses, applications, and features of SX3, and explains how to generate customized configurations for UVC applications and data applications using the SX3 configuration Utility. The application note also discusses the hardware design guidelines and explains different application examples using the SX3 explorer kit and USB3-HDMI capture card.

Intended audience

This application note is intended for customers using EZ-USB SX3.

Table of contents

1 SX3 variants and features

SX3 has integrated the USB 3.1 Gen 1 and USB 2.0 physical layers (PHYs) along with a 32-bit ARM926EJ-S microprocessor for powerful data processing and for building custom applications. It implements an architecture that enables up to 375 MBps data transfer from General Configurable Interface to the USB interface.

SX3 has three variants.

1.1 SX3 Data – 16 bit (CYUSB3015)

- Supports up to two endpoints with vendor class IN, OUT, or IN + OUT configurations up to 190 MBps
- Interfaces with FPGA, high speed ADCs for data acquisition applications
- Supports general configurable interface with slave FIFO interface with configurable bus width (8 or 16 bits) up to 100 MHz
- Includes configurable I2C interface to communicate with peripherals
- Supports SPI flash for firmware and configuration storage
- Supports up to seven configurable GPIOs

1.2 SX3 Data – 32 bit (CYUSB3016)

- Supports up to two endpoints with vendor class IN, OUT, or IN + OUT configurations up to 375 MBps
- Interfaces with FPGA, high speed ADCs for data acquisition applications
- Supports general configurable interface with slave FIFO interface with configurable bus width (8, 16, 24, or 32 bits) up to 100 MHz
- Configurable I2C interface to communicate with peripherals
- Supports SPI flash for firmware and configuration storage
- Supports up to seven configurable GPIOs

1.3 SX3 UVC (CYUSB3017)

- Supports up to two endpoints with USB Video Class (UVC), USB Audio Class (UAC), UVC + UAC configurations.
- Interfaces with image sensors, ISP, FPGA, HDMI Receiver, and so on for audio/video streaming applications
- Supports general configurable interface with slave FIFO or parallel camera interface with configurable bus width (8, 16, 24, 32 bits) up to 100 MHz
- Configurable I2C interface to communicate with peripherals
- Supports SPI flash for firmware and configuration storage
- Supports up to seven configurable GPIOs

2 Hardware design guidelines

2.1 Differences between SX3 and FX3

Table 1 lists the major differences between SX3 and FX3.

Table 1 Differences between SX3 and FX3

SX3	FX3
Configure using the SX3 Configuration Utility	Firmware development using FX3 SDK and EZ-USB Suite
Supports boot from SPI flash	Supports boot from USB, SPI, or I2C
Up to seven user configurable GPIOs	Up to 60 user configurable GPIOs
Supports up to two USB endpoints	Supports up to 32 USB endpoints
General configurable interface (Slave FIFO or camera parallel interface)	GPIF-II interface that is programmable.
121-BGA package	121-BGA package
Supports 512 KB RAM	Supports up to 512 KB RAM
Configurable I2C interface	Programmable I2C interface
Configurable SPI interface (for SPI flash storage and FPGA configuration)	Programmable SPI interface
CDC interface available for debugging. JTAG debug is not supported.	JTAG, UART, and CDC interfaces are available

2.2 Fixed function I/Os

There are four fixed GPIOs for SX3:

- PROGRAM#: This I/O is the output of SX3 used during FPGA configuration. When PROGRAM# is asserted LOW, FPGA enters a device configuration mode.
- INIT# / RESET: This I/O is bidirectional. This signal is used by SX3 to detect whether FPGA has entered configuration mode.

  The same I/O will be used by SX3 to reset FIFO master (FPGA). The reset pin will be asserted if there is an error condition during video streaming. The FIFO master should use this signal to reset the internal logic and restart the streaming.
- FIFOM_SS: This I/O will be used for chip select for SPI interface of FIFO master.
- SUSPEND_OUT: This I/O is output of SX3, used by FIFO master to enter low power mode.

2.3 Configurable GPIOs

There are seven configurable GPIOs (GPIO_0 to GPIO_6) in SX3. The functionality of these GPIOs can be assigned using the SX3 Configuration Utility.

Following are the options of configurable GPIOs:

- Configuration Done: This GPIO is an input to SX3. This signal is used by the FIFO master to indicate that the configuration update is complete.
- USER_GPIO_0 to USER_GPIO_4: User GPIOs are inputs to SX3, and falling edge on a User GPIO results in a I2C write to a specific register in the FIFO Master. Table 2 lists the I2C register address for each User GPIO.

Table 2 User GPIO register address

GPIO	I2C register address
USER_GPIO_0	0x10
USER_GPIO_1	0x11
USER_GPIO_2	0x12
USER_GPIO_3	0x13
USER_GPIO_4	0x14

- Streaming Indication LED (Active LOW): This GPIO will be asserted when any of the SX3 data endpoints are active. The GPIO will also toggle while the FIFO master configuration is in progress.
- Error LED (Active LOW): This GPIO is asserted when the device detects an internal system error condition.
- Still Capture Button (Active HIGH): This GPIO can be used as a hardware trigger signal for still capture (only for SX3 UVC configuration).
- Sensor Reset (Active LOW). This GPIO is toggled before updating video source configuration. This signal can be used as image sensor reset.

2.4 Interfacing FIFO master (FPGA), image sensor with SX3

SX3 supports slave FIFO interface to communicate with FIFO master. FIFO master interface can be implemented on FPGA, ISP, or any other processor.

The data streaming interface can be either slave FIFO or parallel camera interface with data bus width of 8, 16, 24, 32 bits. For details on the signals of these interfaces, see the Sx3 datasheet [9].

2.4.1 Interfacing FIFO master (FPGA) with SX3

The slave FIFO interface uses control signals (SLCS, SLWR, SLOE, SLRD, PKTEND#, PCLK) and DMA flags (DMA_READY and DMA_PARTIAL). For more details on interface timing, see the application note [1].

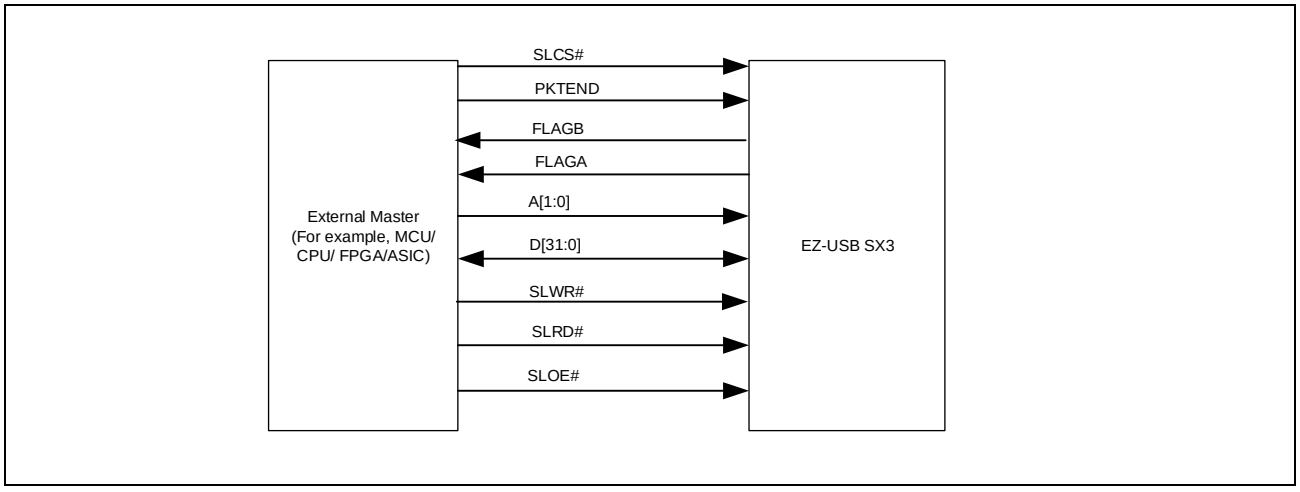

Figure 1 Slave FIFO interface

The slave FIFO interface uses watermark value to detect the full buffer availability for read/write operation. **Table 3** lists the watermark values used for different bus widths.

Table 3 Watermark values

FIFO bus width	Watermark IN/OUT
8	8
16	8
24	12
32	8

SX3 uses SPI and I2C interfaces to download FPGA configuration. SX3 uses I2C interface to communicate with FPGA after configuration.

2.4.2 Parallel camera interface

The parallel camera interface uses frame valid (FV), line valid (LV), clock (PCLK), sensor reset (INT# / RESET) signals. This interface can be used to directly connect image sensors/ISP with parallel interface to SX3.

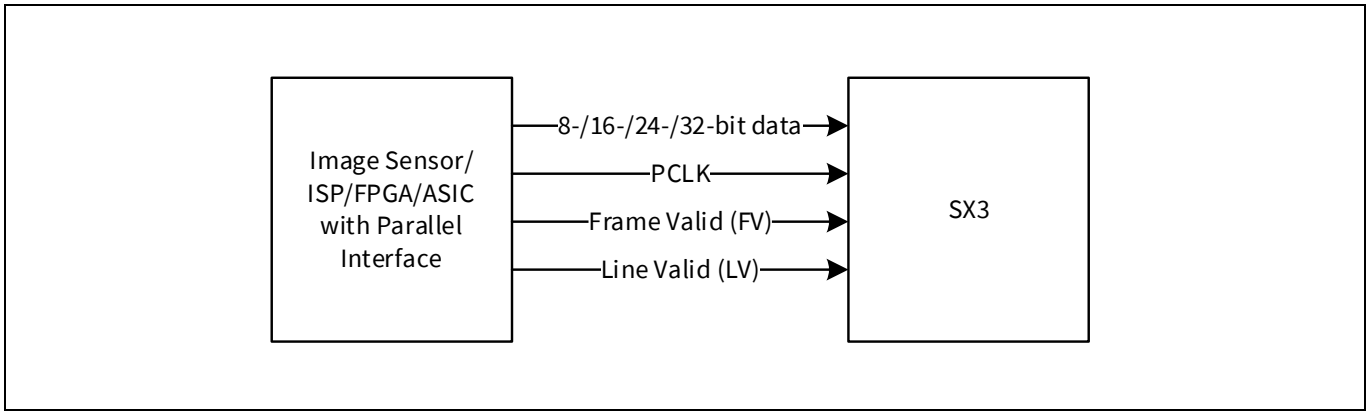

Figure 2 Parallel camera interface

Image sensors, HDMI receivers with MIPI-CSI, LVDS, ITU BT-656 interface, and so on can be connected to SX3 using FPGA or ISP.

2.4.3 Configuring FPGAs from SX3

SX3 supports the following configuration modes:

- Lattice Slave SPI (SSPI) Mode
- Lattice I2C configuration Mode
- Xilinx Slave Serial Mode
- Intel® Passive Serial Mode

These configuration modes are tested in the following FPGA families:

- Lattice Slave SPI Mode (SSPI) - Lattice ECP5
- Lattice I2C configuration mode - Lattice CrossLink
- Xilinx Slave Serial Mode - Xilinx Artix®-7
- Intel Passive Serial Mode - Intel Cyclone® 10 LP

The FPGA configuration file (*.bit*, *.bin*, *.rbf*, and so on) for the FPGA can be provided through the EZ-USB SX3 Configuration Utility. The provided configuration file will be part of the generated SX3 configuration stored in SPI flash. SX3 will read this file and configure FPGA on bootup. For configuration details of each supported FPGA family, see **Configuring Lattice ECP5 FPGA**.

2.4.3.1 Configuring Lattice ECP5 FPGA

SX3 uses SSPI Mode to configure Lattice ECP5. SX3 reads the *.bit* file from SPI flash and sends it to the FPGA using a 30 MHz clock.

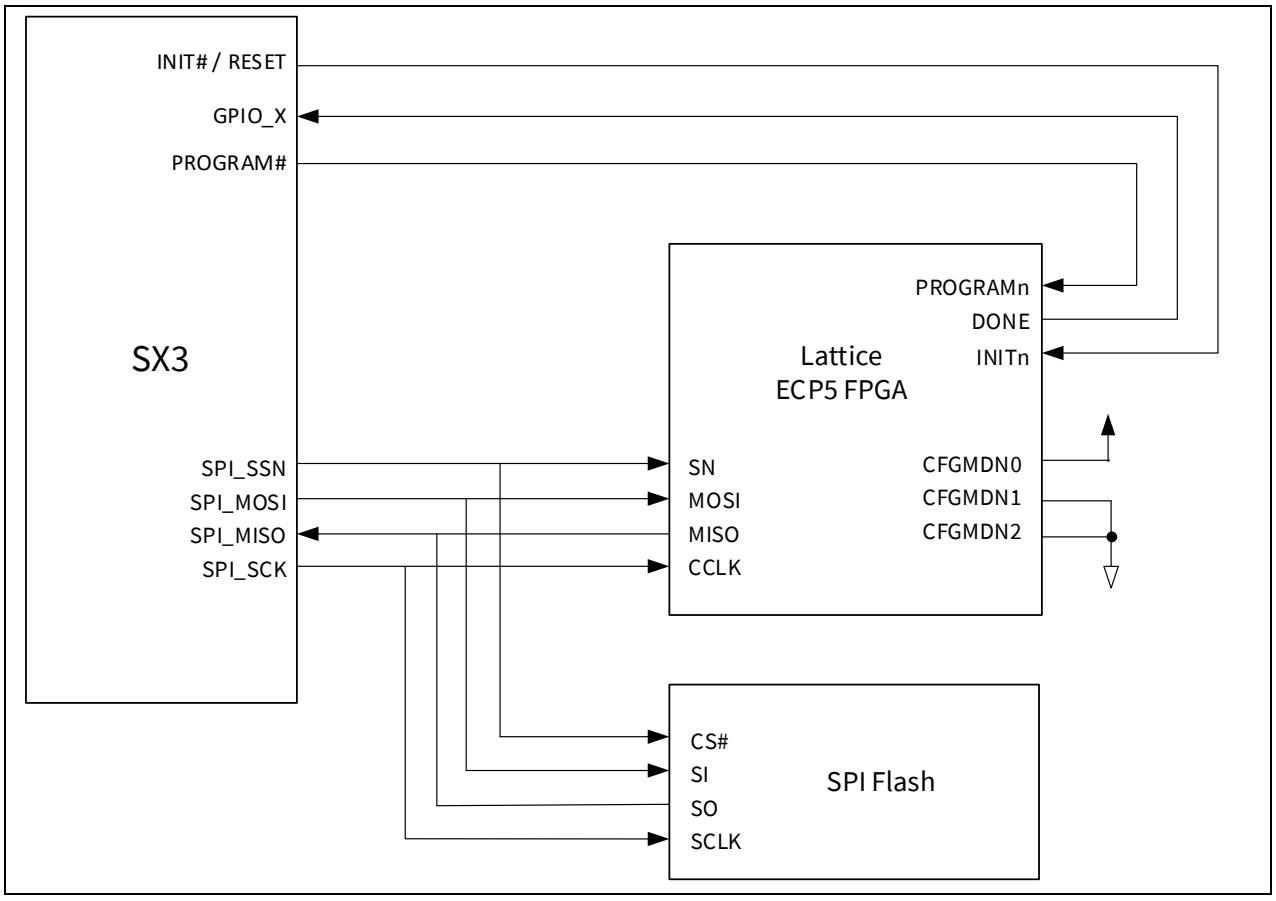

Figure 3 ECP5 Programming with slave SPI port and SX3

For more details, see the Lattice Usage Guide [5].

2.4.3.2 Configuring Lattice CrossLink FPGA

SX3 uses the Lattice I2C configuration mode to configure CrossLink. The *.bit* file is read from SPI flash and sent to CrossLink over I2C using 1 MHz clock.

For more details, see the Lattice Technical Note [6].

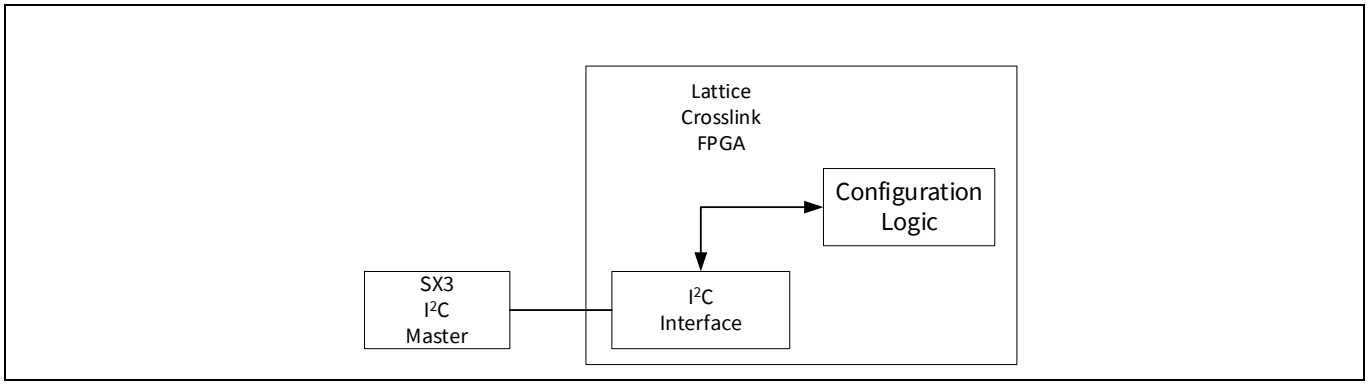

Figure 4 I2C configuration logic

2.4.3.3 Configuring Xilinx Artix-7 FPGA

SX3 uses Xilinx Slave Serial Configuration Mode to configure Artix-7. The *.bin* file is read from SPI flash and is received at the Data In (DI) pin of FPGA.

For more details, see the Xilinx User Guide [7].

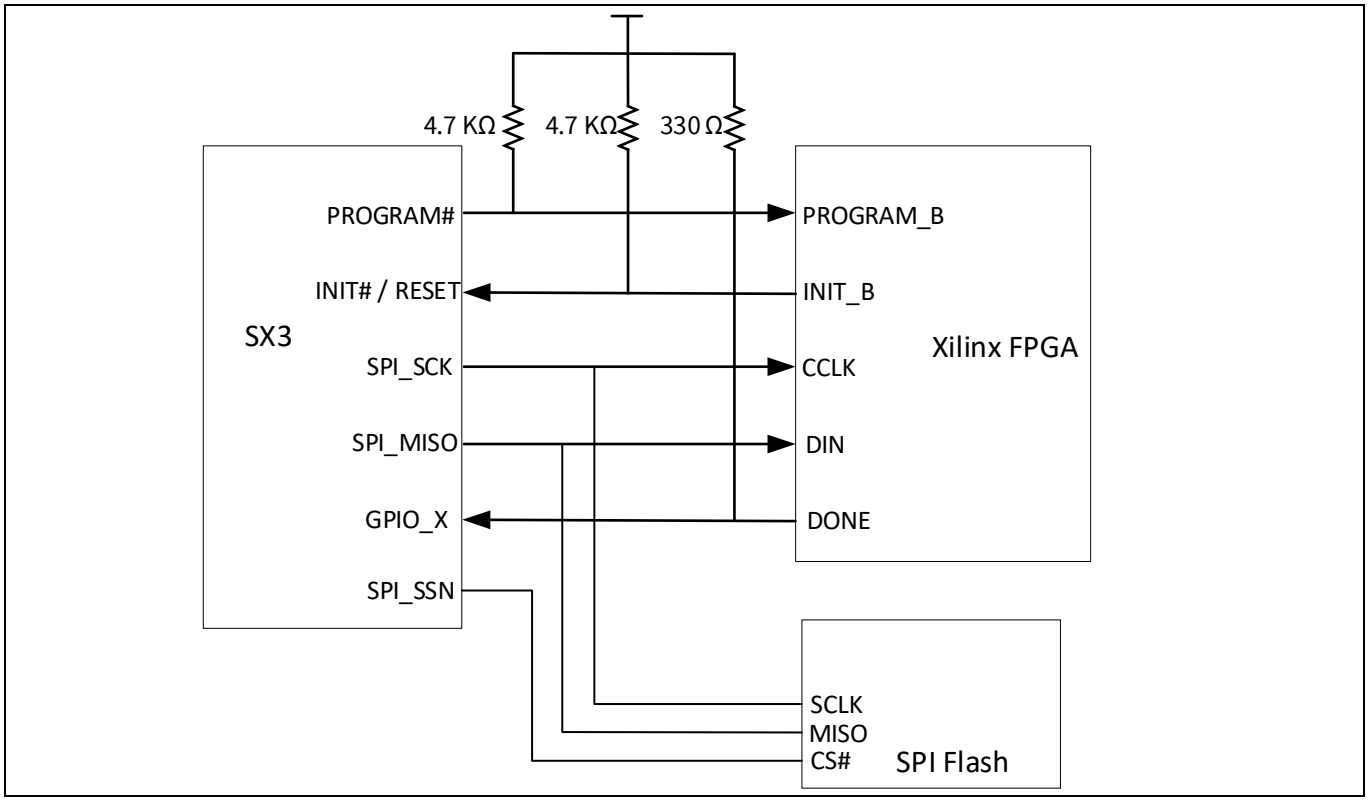

Figure 5 Slave serial mode

2.4.3.4 Configuring Intel Cyclone 10 FPGA

SX3 uses Intel Passive Serial Configuration Mode to configure Cyclone-10. The *.rbf* file is read from SPI flash and is received at the DI pin of FPGA. Note that the *.rbf* file should be in Least Significant Bit (LSb) first format. A python script to convert a generated *.rbf* file to LSb first format is attached with the application note.

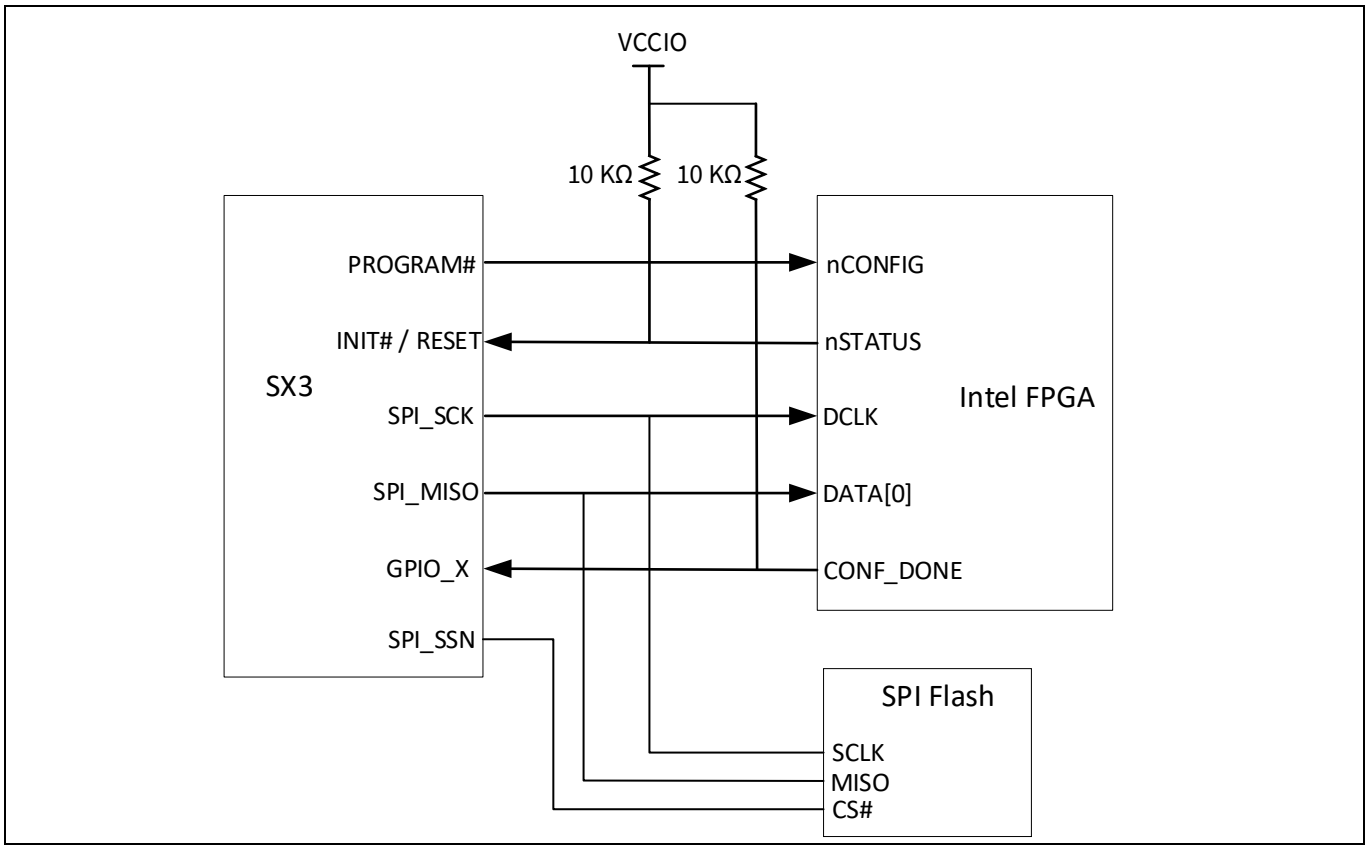

Figure 6 Passive serial mode

For more details, see the handbook **[8]**.

Table 4 shows the interconnection of configuration signals between various FPGA families and SX3.

Table 4 Interconnection of configuration signals for various FPGA families and SX3

SX3 Signal Name	Lattice ECP5	Xilinx Artix-7	Intel Cyclone 10 LP	Lattice CrossLink
FIFOM_SS	CSSPIN	-	-	-
PROGRAM#	PROGRAM_N	PROGRAM_B	N_CONFIG	-
INIT# / RESET	INIT_N	INIT_B	N_STATUS	CRESETB
GPIO_0 to GPIO_6	DONE	DONE	CONF_DONE	CDONE
SPI_SCK	MCLK/CCLK	CCLK	DCLK	-
SPI_MOSI	MOSI	DIN	-	-
SPI_MISO	MISO	DOUT	DATA[0]	-
I2C_SCL	-	-	-	SCL
I2C_SDA	-	-	-	SDA

2.5 SPI Flash for configuration storage

SX3 supports USB boot mode only for downloading the SX3 configuration. Once the SX3 configuration is downloaded, SPI boot mode should be selected to run the application. The SX3 configuration can be downloaded to SPI flash using the SX3 Configuration Utility. The SX3 Configuration Utility will merge the FIFO master (FPGA) configuration file with the SX3 configuration, and downloaded it to SPI flash. The SPI interface will not be available once the device is in application mode. For more details on SPI flash interface to SX3, see the SX3 development kit schematic and the application note [2].

2.5.1 Image sensor interface

The initialization of I2C register configuration can be written to the image sensor over the I2C interface.

SX3 supports writing I2C register configuration corresponding to the video resolution each time the resolution is selected in the UVC host application.

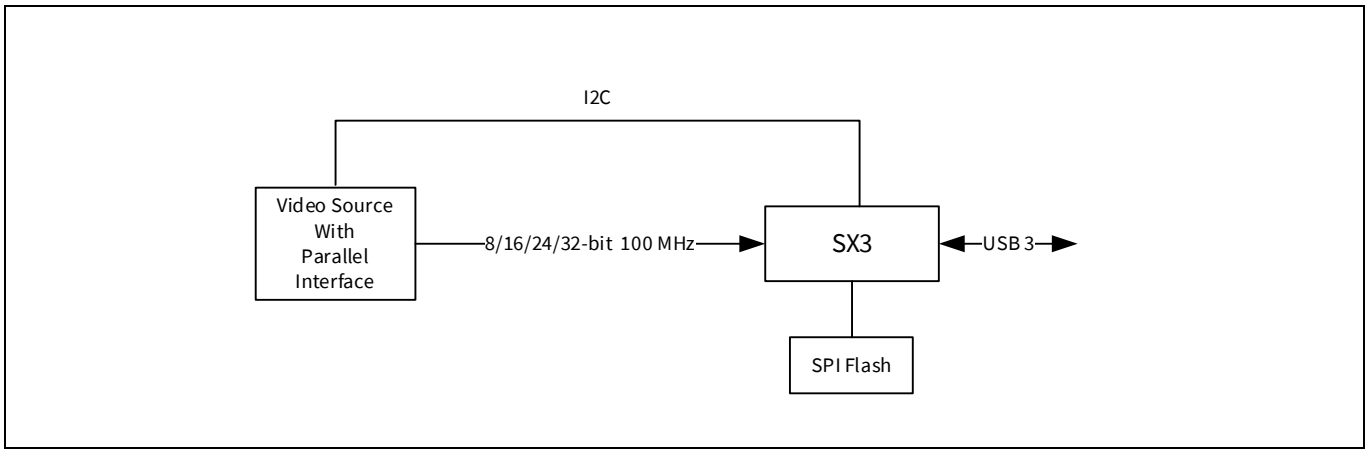

Figure 7 Direct image sensor interface with SX3

3 Applications

3.1 SX3 - UVC application

SX3 UVC (CYUSB3017) supports up to two endpoints that can be configured as UVC, UAC, or UVC + UAC. SX3 supports UVC version 1.1/1.5 and UAC 1.0. The UVC streaming endpoint can be Bulk or Isochronous. In the USB HighSpeed mode, only Bulk configuration is supported. The following sections describe typical application examples for SX3 using these configurations.

3.1.1 HDMI USB capture card application

The HDMI USB capture card application uses the UVC + UAC configuration to support simultaneous video and audio streaming. **Figure 8** shows the block diagram of this application.

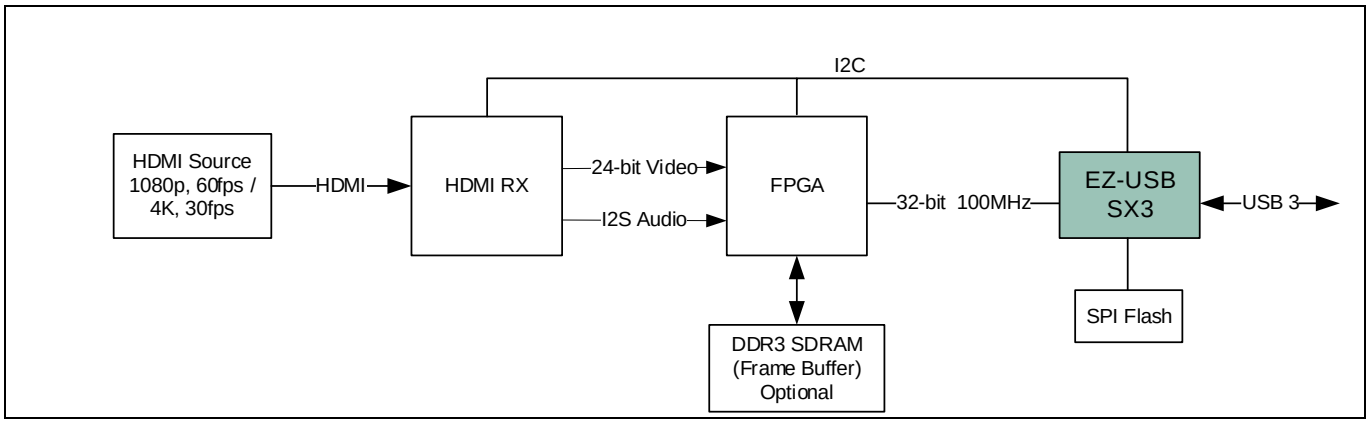

Figure 8 4K HDMI USB3 capture card

The HDMI receiver converts the HDMI video and audio stream to parallel video interface and I2S audio interface. A FIFO master (FPGA) is required to convert the data and send it to SX3 over the slave FIFO interface. FIFO master supports frame buffer, which is required for supporting 4K video stream. SX3 uses I2C interface to communicate with HDMI RX and FIFO master. SX3 configuration tool has two types of HDMI receivers:

- HDMI RX - IT6801: In this mode, all HDMI events specific to IT6801 HDMI receiver will be handled by custom configuration.
- HDMI RX - Generic: In this mode, SX3 can be configured to detect any HDMI events with I2C-based register polling

Table 5 Differences between HDMI RX IT6801 and HDMI RX – generic configurations

HDMI RX - IT6801	HDMI RX - Generic
Applicable for IT6801 HDMI receiver	Applicable for all other HDMI receivers and also custom configurations of IT6801.
HDMI event handling is not configurable.	You can configure SX3 to handle HDMI events using SX3 Configuration Utility.
HDMI initialization and EDID configuration can be done using I2C interface.	HDMI initialization and EDID configuration can be done using I2C interface.
Auto-detection of HDMI resolution change and re-enumeration of UVC interface.	Must manually switch resolution in the UVC player host application to match the HDMI source resolution.

Applications

HDMI RX - IT6801	HDMI RX - Generic
Auto-detection of interlaced and progressive resolutions and configure FPGA to convert interlaced to progressive.	Interlaced to progressive conversion not supported.

3.1.1.1 Video and audio support

SX3 can support up to 4K 30fps uncompressed video along with audio of up to 32-bit, 12 channels. FPGA implements I2S receiver and sends out the data over the slave FIFO interface.

3.1.1.2 HDMI event handling in generic HDMI configuration

SX3 supports HDMI-generic configuration to support interface a generic HDMI receiver with the USB Device Controller. Add the HDMI events to be handled by USB Device Controller in the Event Handling Table of SX3 Configuration Utility . SX3 will detect the events and perform I2C writes to handle the specific event. Add the HDMI interrupt event register, event mask, and I2C structure to be written in the specific format mentioned in the **Help content** tab of the SX3 configuration utility and SX3 configuration utility user guide.

3.1.1.3 FPGA project for interfacing HDMI RX to SX3

An example FPGA project for interfacing ITE HDMI receiver IT6801 is available with the SX3 HDMI capture card kit. This FPGA project is developed for Lattice ECP5 FPGA. You can port the project to other FPGA families of your choice.

FPGA uses internal block memory to buffer the incoming video and stream it using FIFO interface to SX3.

FPGA uses ECP5 Memory Controller Block (MCB) to store the incoming video frames on an external DDR3 memory as frame buffer for UHD (4K) 30 fps video.

FPGA also converts the UHD video format from YUV422 to YUV420 to fit 4K 30 fps video in USB 3 bandwidth. The interlaced to progressive conversion is also supported in FPGA for 1920 x 1080i, 60 Hz HDMI resolution. SX3 detects interlaced video and communicates with FPGA. The odd and even lines are read twice from the RAM to convert the interlaced video to the progressive format.

Audio data is received in the I2S format with 2-channel, 16-bit sample width, and 48 KHz. The I2S receiver de-serializes the data and stores it in the audio buffers. Audio data is sent over the slave FIFO interface during the video frame blanking period.

As per the HDMI standard 1.4, the following resolutions are supported:

1. 640 x 480p, 60 Hz
2. 720 x 480p, 60 Hz
3. 720 x 576p, 50 Hz
4. 1280 x 720p, 60 Hz
5. 1920 x 1080i, 60 Hz
6. 1920 x 1080p, 60 Hz
7. 3840 x 2160p, 30 Hz

3.1.1.4 PCLK reduction, packing data to 32 bit, frame buffer support

The HDMI RX sends data to FPGA over a 16-bit interface, where the clock frequency can go up to 150 MHz. FPGA will use the frame buffer to pack the data to 32-bit wide format and send to the SX3 slave FIFO interface with 100 MHz clock.

3.1.1.5 FIFO master interface (multi-socket support, current thread DMA flag)

SX3 supports two sockets for each endpoint (UVC/UAC). For details on slave FIFO signals used and watermark value definition, see the application note **[1]**. The HDMI RX FPGA project supports 32-bit data bus, and the watermark value used is 8.

3.1.1.6 I2C slave interface support on FPGA - register details

SX3 uses the fixed I2C registers, listed in **Table 6**, which are to be implemented in FPGA. FPGA should implement I2C slave interface with the address width as two bytes and data width as one byte.

The SX3 uses the register map, listed in **Table 6**, to communicate with FPGA for different events. The register address is fixed for some functionalities. The provided FPGA project gives a reference implementation for each of the registers listed in **Table 6**.

Table 6 Register table

FPGA register address	Register name	Register address fixed in SX3	Default (Hex)	Description/Comments
0x0000	DMA Channel Reset	Yes	0x00	0: No active DMA Reset event
				1: UVC DMA reset occurred in SX3
				2: UAC DMA reset occurred in SX3
0x0009	Endpoint 1 Stream control	Yes	0x01	'1'- Enable stream
				'0'- Disable stream
0x000A	Endpoint 2 Stream control	Yes	0x01	'1'- Enable stream
				'0'- Disable stream
0x000E	Still capture Signaling	No	0x00	'1'- Start of still capture
				'0' – Stop still capture
0x0010	GPIO0 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO0
0x0011	GPIO1 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO1
0x0012	GPIO2 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO2
0x0013	GPIO3 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO3
0x0014	GPIO4 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO4
0x0015	GPIO5 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO5
0x0016	GPIO6 register	Yes	0x00	Value 0x01 will be written on negative edge of GPIO6

Applications

FPGA register address	Register name	Register address fixed in SX3	Default (Hex)	Description/Comments
0x0021	Image Height (MSB)	No	0x04	Number of lines in a frame (MSB)
0x0022	Image Height (LSB)	No	0x38	Number of lines in a frame (LSB)
0x0023	Image Width (MSB)	No	0x07	Number of pixels in a line (MSB)
0x0024	Image Width (LSB)	No	0x80	Number of pixels in a line (LSB)
0x0025	YUV422 - YUV420 conversion enable	No	0x00	0x01- Enables YUV422 - YUV420 conversion
				0x00- Disables YUV422 - YUV420 conversion
0x0026	Interlaced input enable	Yes	0x00	0x01- Indicates Interlaced format input
				0x00- Indicates Progressive format input

Note:

1. *The I2C register addresses that are marked as "fixed" are mandatory and should be implemented in the FIFO master.*
2. *SX3 can also write to custom I2C register addresses that are passed through the SX3 Configuration Utility. For more details, see **Configuration Utility**.*
3. *User GPIO Handling: SX3 signals falling edge on any user GPIO to the FIFO master by writing a '1' to the corresponding Register. FIFO Master is expected to clear the value to '0' after reading the set value.*

3.1.1.7 Development kit: e-CON Systems SX3 FPGA HDMI RX kit (PICTOR)

SX3 HDMI capture card kit includes two boards: SX3 FPGA base board and an add-on board HDMI RX. You can make your own add-on board using different HDMI RX, or other imaging source (such as SDI receiver, DP receiver, image sensor/ISP), and interface with the SX3 FPGA base board. For more details on this kit, visit the **SX3 product webpage**.

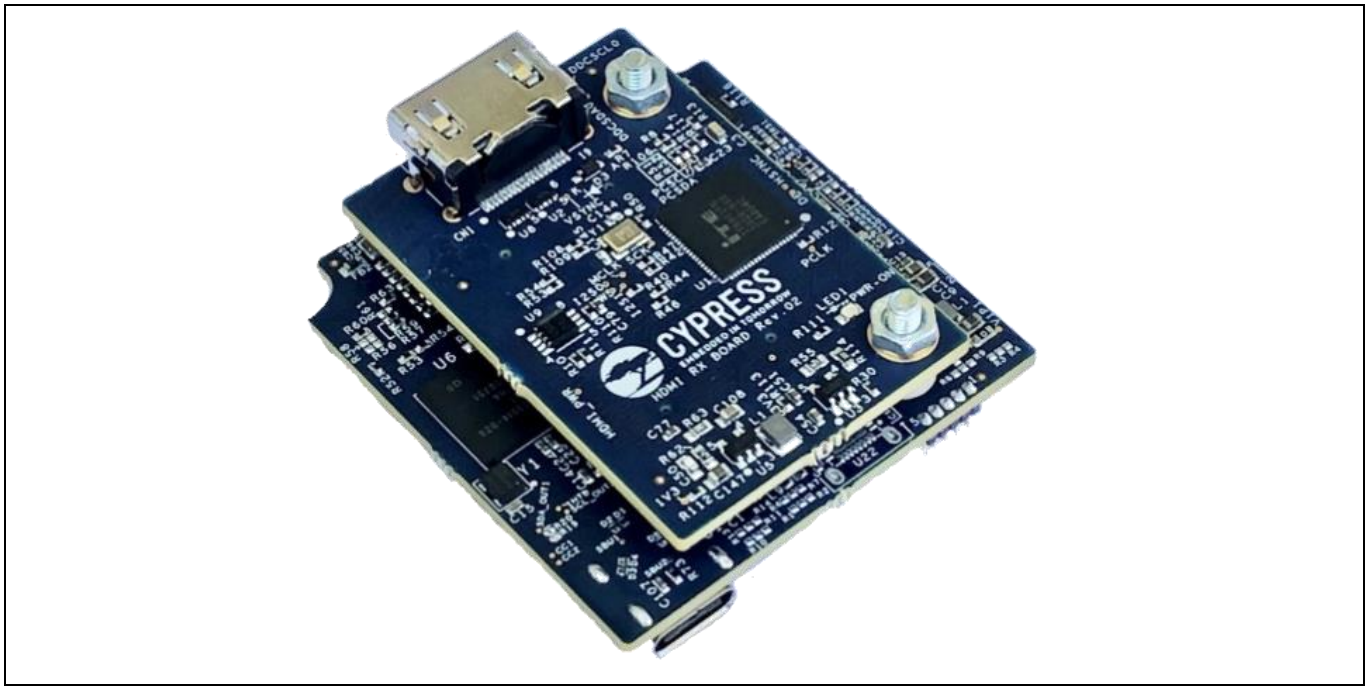

Figure 9 SX3 HDMI USB3 capture card kit (PICTOR)

The following configurations can be evaluated using the **SX3 FPGA HDMI RX kit**:

- SX3_UVC_UAC_HDMI_ITE_1080p
- SX3_UVC_UAC_HDMI_GENERIC_1080p
- SX3_UVC_UAC_HDMI_ITE_4K
- SX3_UVC_UAC_HDMI_GENERIC_4K
- SX3_UVC_FV_LV_BULK_32
- SX3_UVC_UAC_COLORBAR_BULK_32
- SX3_UVC_UAC_COLORBAR_ISOC_32
- SX3_UVC_24_BIT_ISOC

3.1.2 UVC camera application using image sensor and FPGA

The SX3 UVC variant can be used to design UVC applications where the image sensor is connected to an ISP /FPGA. **Figure 10** is an example where an image sensor is interfaced with SX3 using FPGA.

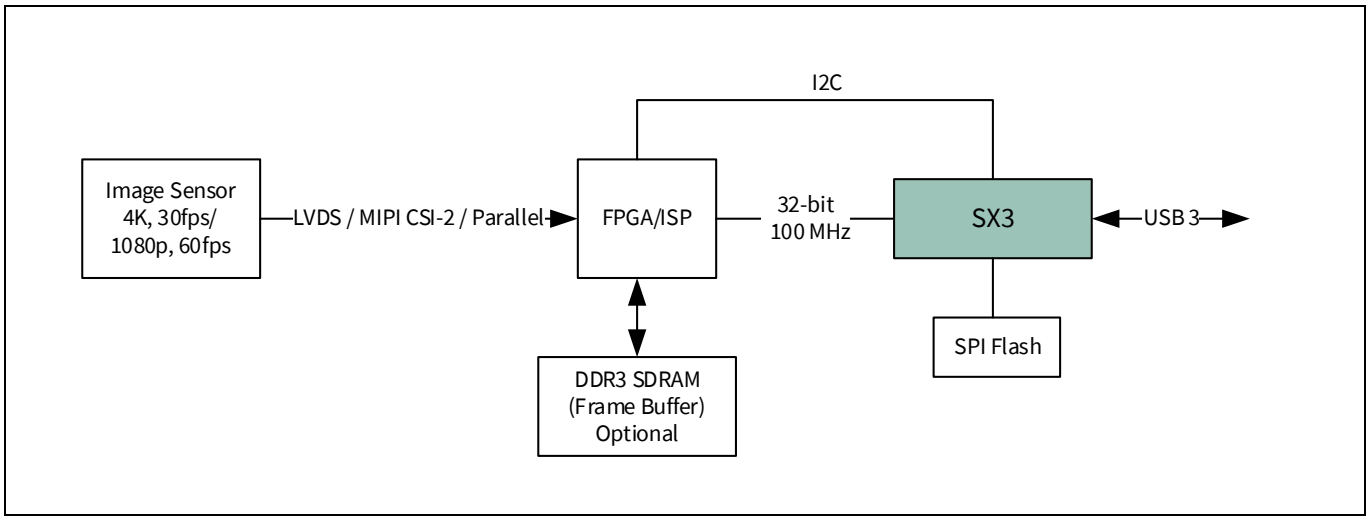

Figure 10 USB3 camera

3.1.2.1 Using crosslink FPGA and OV5640 sensor module with SX3

SX3 can be interfaced with OV5640 image sensor using Lattice crosslink FPGA. The OV5640 image sensor with built-in ISP provides YUV data format through MIPI CSI-2 interface. Lattice crosslink FPGA implements MIPI CSI - 2 to parallel bridge and provides data to SX3 over parallel camera interface.

Contact **Infineon Technical Support** for more details about hardware setup to be used for this configuration.

The following configurations can be evaluated using this kit:

- SX3_UVC_CROSSLINK_BULK_720
- SX3_UVC_CROSSLINK_BULK_1080
- SX3_UVC_CROSSLINK_BULK_VGA

3.1.2.2 Configuring FPGA

SX3 will configure the Lattice crosslink FPGA using I2C interface on bootup.

3.1.2.3 Configuring image sensor

SX3 will configure the OV5640 sensor using I2C interface.

3.1.3 Direct interface of image sensor

An example FPGA project, provided with this application note, simulates image sensor with 32-bit parallel interface with FV, LV signals. You can evaluate this using the SX3 FPGA base board, which is part of SX3 FPGA HDMI RX kit.

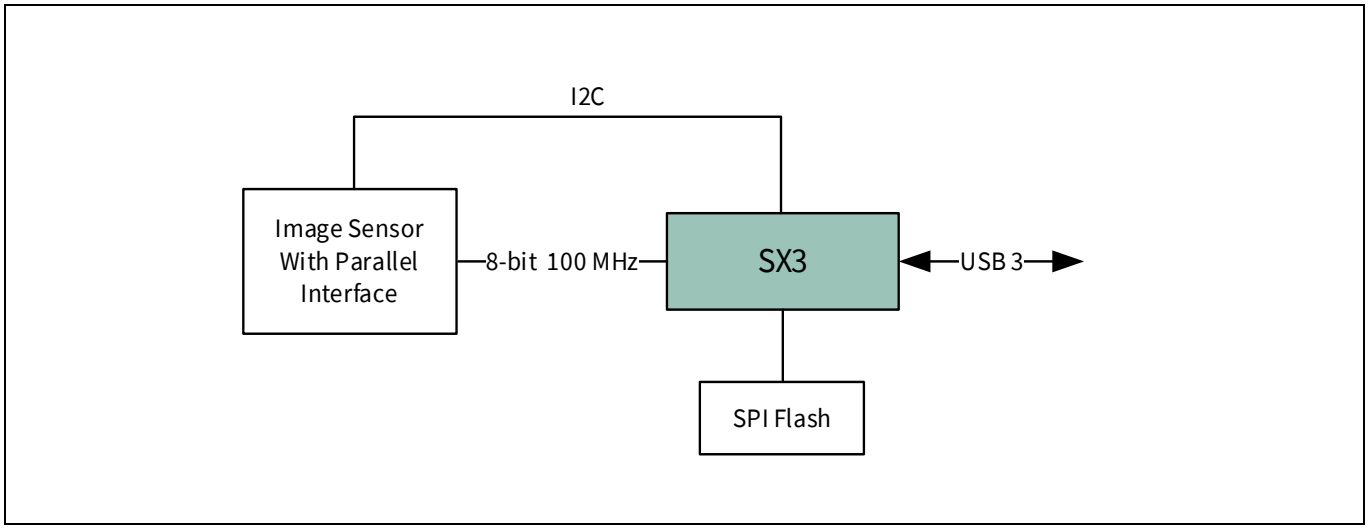

Figure 11 Direct image sensor interface to SX3

The following configuration can be evaluated SX3 FPGA HDMI RX kit.

- SX3_UVC_FV_LV_BULK_32

3.1.4 Host applications

Various host applications allow you to display and capture video from the SX3 UVC device. **Table 7** lists the popular host applications that can be used on different operating systems.

Table 7 Host applications for various operating systems

Operating systems	Host applications
Windows	Microsoft Windows Camera, MPC-HC, e-CAMView, VLC Player
Linux	QtCAM, VLC Player, Guvcview
macOS	Webcamoid, Photo Booth, QuickTime Player

UVC Still Image capture is supported only on e-CAMView.

3.1.5 Driver requirement and multi-OS support

SX3 UVC is supported by the in-built default drivers included in the OS.

Table 8 Driver requirement and multi-OS support

Operating systems	Drivers
Windows	Windows UVC driver
Linux	uvcvideo driver
macOS	In-built driver

3.2 Data application

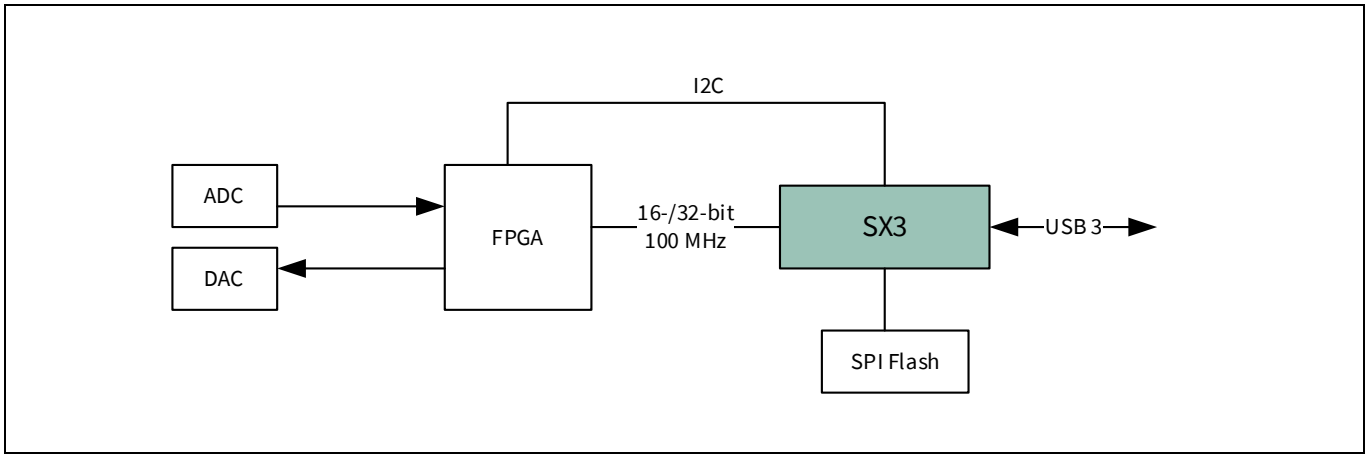

Figure 12 SX3 data interfaced to FPGA

SX3 Data – 16bit (CYUSB3015), SX3 Data - 32bit (CYUSB3016) parts support up to two data streaming endpoints that can be configured as IN only, OUT only or IN + OUT configuration. These SX3 variants can be used in generic data acquisition, logic analyzer, and USB oscilloscope applications. SX3 uses USB Bulk endpoint, Slave FIFO interface with clock speed up to 100 MHz. The following sections explain the typical application examples for SX3 Data variants.

3.2.1 Data streaming application example using FPGA

The FPGA example project provided with the application note can be used to transfer data continuously in the IN or OUT direction.

Table 9 lists the configurations supported by the FPGA project and the throughput measured in SuperSpeed and HighSpeed modes. However, the maximum achievable throughput also depends on critical factors such as host PC controller type, operating system, and USB design (transfer type and buffer sizes).

Table 9 Throughput measurement for SX3 data variants

Variant	Endpoint configuration	Throughput		Host application
		SuperSpeed	HighSpeed	
SX3 Data -32bit	IN	390 MBps	46 MBps	Streamer
SX3 Data -32bit	OUT	389 MBps	42 MBps	Streamer
SX3 Data -32bit	IN + OUT	NA	NA	Bulkloop
SX3 Data -16bit	IN	195 MBps	46 MBps	Streamer
SX3 Data -16bit	OUT	192 MBps	42 MBps	Streamer
SX3 Data -16bit	IN + OUT	NA	NA	Bulkloop

Note: Throughput is not measured for IN + OUT configuration as the Bulkloop host application will have additional overhead due to the data comparison and does not provide actual throughput.

Applications

The following SX3 template projects are part of SX3 configuration tool and can be evaluated using SX3 FPGA base board, which is part of SX3 HDMI RX kit:

- SX3_DATA_IN_32
- SX3_DATA_OUT_32
- SX3_DATA_IN_OUT_32
- SX3_DATA_IN_16
- SX3_DATA_OUT_16
- SX3_DATA_IN_OUT_16

The SX3 Data – 16-bit (CYUSB3015) template projects will also work with SX3 Data – 32-bit (CYUSB3016) and SX3 UVC (CYUSB3017) part numbers.

The SX3 Data – 32-bit (CYUSB3016) template projects will also work with SX3 UVC (CYUSB3017) part number.

3.2.2 Host applications – SX3 data

The SX3 Data variants can be tested with different host applications like streamer, control center, bulkloop, and so on. The host applications are available in the *tools* folder in installation path of the SX3 Configuration Utility. For more details on creating custom host applications, see the application note **[3]**.

Table 10 lists the host applications that can be used for each operating system.

Table 10 Host applications for different operating systems

Operating system	Applications	Functionalities
Windows	Streamer	Independent IN or OUT transfer
	Control Center	Single IN/OUT transfer
	Bulkloop	Loopback of IN + OUT
Linux, macOS	cybulkwrite_performance	Independent IN transfer
	cybulkread_performance	Independent OUT transfer

3.2.3 Driver requirement and multi-OS support - SX3 data

The template projects use Cypress VID and PID and binds to the *cyusb3.sys* driver, which is a vendor class driver on Windows. You can bind to another driver by changing the VID/PID using the SX3 Configuration Utility.

The template projects use the *libusb* driver in macOS and Linux. These projects are tested on Linux Ubuntu 20.04 and macOS High Sierra.

4 Configuration Utility

The EZ-USB SX3 Configuration Utility is a software application that can be used to configure SX3 device based on your application or system requirements. The utility allows you to intuitively select and configure the parameters for your application and thus saves time on firmware development. This utility also allows programming of the connected SX3 device with the generated configuration.

4.1 Features

- Provides support on Windows, Linux, and macOS
- Supports configuration of SX3-UVC (CYUSB3017) and SX3-Data (CYUSB3015 and CYUSB3016) variants
- Supports generation of new configuration and import of existing configurations
- Creates a single merged file with the SX3 device configuration, FIFO Master (FPGA/ISP) configuration, and Video Source (Image Sensor/ HDMI Receiver) configuration
- Supports programming of SX3 device with the generated configuration file
- Supports import and export of generated device configurations
- Provides integrated help content for each configuration parameter in the **Help** tab
- Allows you to view and save application logs in the **Log** tab

4.2 Installing Configuration Utility

You can download and install this tool from **SX3 product page**.

After installing, go to the Windows **Start** > **All Programs** > **Cypress > EZ-USB SX3 Configuration Utility** > **EZ-USB SX3 Configuration Utility**.

The workflow for configuring any SX3 device includes three stages:

1. **Creating or Importing existing configuration:** Create a new configuration from the **File** menu of the utility or import an existing configuration and modify.
2. **Editing parameters:** Edit the application-specific parameters.
3. **Configuring Device:** Program the device using the **Program Configuration** option.

For more details on configuration and firmware update of the device, follow the menu path **Help** > **User Guide** to access the EZ-USB SX3 Configuration Utility user guide.

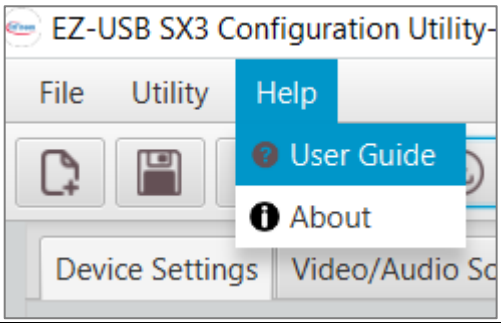

Figure 13 Help menu

Configuration Utility

The SX3 Configuration Utility is supported on Windows, Linux, and macOS. The Configuration Utility was tested using the following versions:

- Windows 10
- Linux Ubuntu 20.04
- macOS High Sierra

The SX3 Configuration Utility supports device programming feature using the drivers listed in **Table 11**.

Table 11 Supported drivers

Operating system	Driver
Windows	*Cyusb3.sys* provided by Infineon
Linux	Libusb
macOS	Libusb

4.2.1 Installing Windows driver

During the installation of the SX3 Configuration Utility, the *cyusb3.sys* driver is automatically copied, and driver files are in *<installation path> \Config tool\SX3ConfigurationUtility\app\drivers\Win10*. For more details on driver installation, see the section Windows Driver Installation of the EZ-USB SX3 Configuration Utility user guide.

4.2.2 Installing macOS driver

For details on installing *libusb* for macOS, see the Appendix related to macOS Driver Installation in the EZ-USB SX3 Configuration Utility user guide.

4.2.3 Installing Linux (Ubuntu) driver

For details on installing *libusb* for Linux (Ubuntu), see the Appendix related to Linux Driver Installation in the EZ-USB SX3 Configuration Utility user guide.

4.3 File storage

For the installation directory structure, see the Installation Folder section of the EZ-USB SX3 Configuration Utility user guide.

4.4 Working of SX3 Configuration Utility

The SX3 Configuration Utility generates following files:

- Configuration json file: The configuration you entered is saved in the user-readable json format. The json file can also be used to import any saved configuration.
- Hex file: Configuration table generated by the utility based on data in the json file.
- Merged Configuration file (*.img*): Generated by combining the firmware, configuration parameters, and the FPGA bit file (if you included). This file will be downloaded to the SPI flash during programming.

  See **Figure 14** for more details.

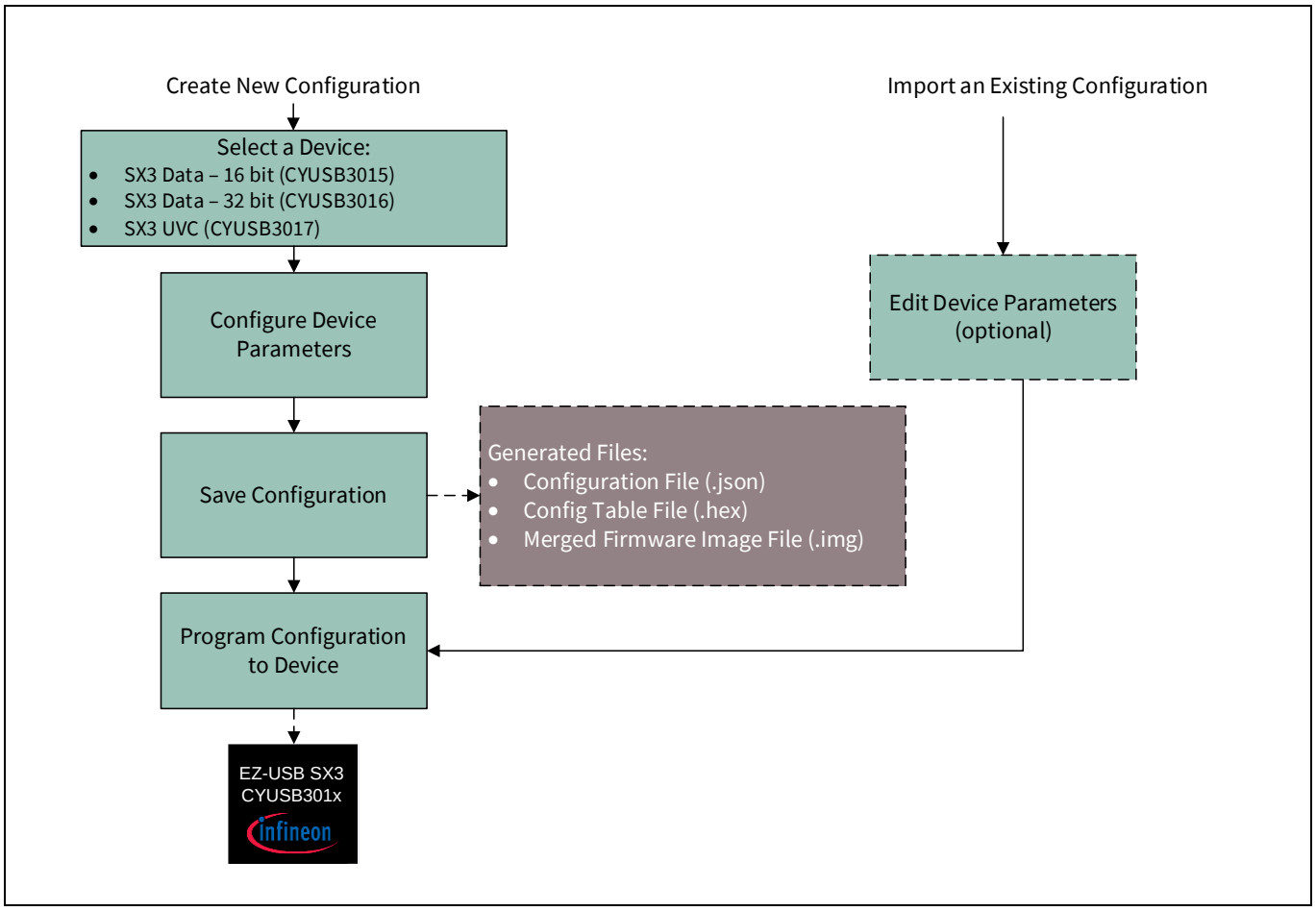

Figure 14 Working of SX3 Configuration Utility

4.4.1 Merging of files

The SX3 Configuration Utility will merge the FPGA bit file with the SX3 configuration file present in the *firmware* folder. The SX3 configuration will be saved from the first address location. The FPGA bit file will start from address location after the SX3 configuration.

4.4.2 Programming

The final configuration file can be programmed to the external SPI flash using USB boot mode or SPI boot mode. For details on the available options, see **Firmware update**.

5 Interfacing custom devices with SX3

Contact **Infineon Technical support** for interfacing custom Video sources, FPGAs, ISPs, and so on.

6 Debugging SX3-based designs

6.1 CDC interface

The SX3 device supports a CDC interface (virtual COM port) that can be enabled using the SX3 Configuration Utility. The debug logs can be collected using a standard terminal application after USB enumeration. The debug log contains information about error conditions, DMA producer, consumer events, frame rate, I2C writes, and so on.

You can enable or disable the CDC interface from the **Device Settings** tab of the SX3 Configuration Utility. The logs available for enabled based on debug levels.

6.1.1 Debugging levels

The SX3 CDC interface supports five debug levels. **Table 12** lists the debug logs available for each level. .

Table 12 SX3 debug Levels

Debug levels	Available logs
Level 0	Error messages only
Level 1	Error messages, start/stop messages, suspend/resume status
Level 2	Error messages, start/stop messages, suspend/resume status, UVC and UAC commands, , all I2C writes
Level 3	Error messages, start/stop messages, suspend/resume status, UVC commands, all I2C writes, HDMI event log
Level 4	Error messages, start/stop messages, suspend/resume status UVC commands, I2C writes, HDMI event log, DMA statistics (producer count, consumer count, fps, and so on)

6.1.2 Debug terminal applications

You can use any standard debug terminal application to collect the log from the SX3 device. **Table 13** lists the sample host applications that can be used on different operating systems.

Table 13 Debug terminal applications

Operating system	Application
Windows	Tera Term, PuTTY
Linux	Cutecom, PuTTy
macOS	CoolTerm

Note: No specific flowcontrol, baudrate setting is required. The COM port number can be identified from the Device Manager in Windows, using System Report in macOS and the `dmesg` command in Linux.

6.2 Help tab

Go to the **Help** tab for information on configurable parameters. Select a control and the related information is displayed in the **Help** tab.

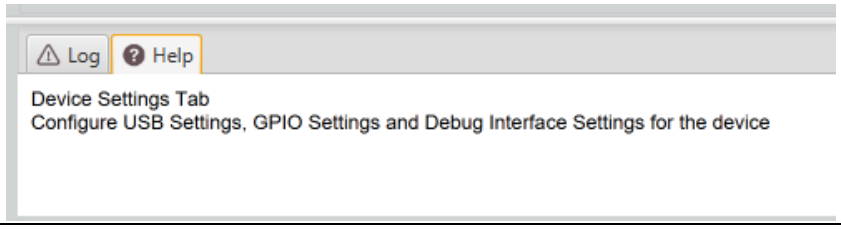

Figure 15 Displaying the Help tab

6.3 SX3 Configuration Utility user guide

You can view the SX3 Configuration Utility user guide from the **Help** menu.

7 Associated project files

Table 14 lists the links to the source code for the FPGA projects and sample applications available in GitHub repository.

Table 14 Project files

File/Folder name	Description
sx3_hdmi_4k_framebuffer	FPGA project source code to interface HDMI Receiver with SX3 with DDR based frame buffer implementation to support resolutions of up to 4K 30fps.
sx3_hdmi_1080p	FPGA project source code to interface HDMI Receiver with SX3 without frame buffer implementation to support video resolutions of up to 1080p 30fps.
sx3_data_slavefifo_example	FPGA source code that can be used to test SX3 data parts (CYUSB3015/CYUSB3016). Three modes are supported: • IN only • OUT only • IN + OUT in loopback mode I2C slave implementation is not added in this FPGA project. Compile time switches are provided to change between each mode and change the bus width to 16 bits or 32-bits.
sx3_testpattern	FPGA source code that generates single-tone sine wave audio data and color bar for video resolution with height and width configurable through I2C interface. This test project can be used to validate FPGA to SX3 interface without dependency on HDMI Receiver interface. This FPGA project can be used to test UVC only, UAC only, UVC+UAC configurations. The interface between SX3 and FPGA can be either of the following: • Slave FIFO mode – Supports audio and video pattern with data bus width of 8/16/24/32-bits. • Camera parallel interface mode – Supports video only with bus width 8/16/24/32-bits. The interface mode and bus width can be selected with a compile time switch in the FPGA project.
sx3_hid_sample_app	A sample HID command line application for Windows OS that can be used for firmware update over HID interface. The application takes VID, PID, and firmware image as inputs and updates the SPI flash connected to SX3. The HID application also supports the command to erase SPI flash and fallback to USB boot loader.
sx3_uvc_xu_sample_app	A sample UVC extension unit application for Windows OS that can be used for reading SX3 firmware version and for resetting the device. SX3 does not support additional extension unit commands.

The FPGA bit files that are part of the SX3 configuration templates can be generated from the FPGA projects mentioned in Table 14. Table 15 lists the FPGA bit files used along with example configuration templates and how it can be generated from the FPGA projects listed in Table 14.

Associated project files

Table 15 FPGA bit files and example configuration templates

S.No	SX3 configuration template	FPGA bit file name	Remarks
1.	sx3_data_in_16	slfifo_interface_impl1_16_ streamin.bit	Generated from sx3_data_slavefifo_example Enable STREAM_IN_ONLY, GPIF_WDT_16 in params.v
2.	sx3_data_in_32	slfifo_interface_impl1_32_ streamin.bit	Generated from sx3_data_slavefifo_example Enable STREAM_IN_ONLY, GPIF_WDT_32 in params.v
3.	sx3_data_in_out_16	slfifo_interface_impl1_16_ loopback.bit	Generated from sx3_data_slavefifo_example Enable LOOPBACK, GPIF_WDT_32 in params.v
4.	sx3_data_in_out_32	slfifo_interface_impl1_32_ loopback.bit	Generated from sx3_data_slavefifo_example Enable LOOPBACK, GPIF_WDT_32 in params.v
5.	sx3_data_in_out_intel	altera_input_flipped.rbf	Source code not available. Only for config update testing.
6.	sx3_data_in_out_xilinx	xilinx_artix7_abni_test.bit	Source code not available. Only for config update testing.
7.	sx3_data_out_16	slfifo_interface_impl1_16_ streamout.bit	Generated from sx3_data_slavefifo_example Enable STREAM_OUT_ONLY, GPIF_WDT_16 in params.v
8.	sx3_data_out_32	slfifo_interface_impl1_32_ streamout.bit	Generated from sx3_data_slavefifo_example Enable STREAM_OUT_ONLY, GPIF_WDT_32 in params.v
9.	sx3_uac	testptrn_proj_slfifo_audonly.bit	Generated from sx3_testpattern Enable SLFIFO_INTERFACE, SLFIFO_INTERFACE_AUD in parameters.v
10.	sx3_uvc_24_bit_isoc	sx3_uvc_24bit_isoc.bit	Generated from sx3_testpattern. Enable SLFIFO_INTERFACE in parameters.v, Change INIT_GPIF_WDT value to 3 in i2c_slave.v
11.	sx3_uvc_crosslink_bulk_720p	Crosslink_720p.bit	Source code not released.
12.	sx3_uvc_crosslink_bulk_1080p	Crosslnk_1080p.bit	Source code not released.
13.	sx3_uvc_crosslink_bulk_VGA	Crosslink_VGA.bit	Source code not released.
14.	sx3_uvc_fv_lv_bulk_32	testptrn_proj_camera_ interface.bit	Generated from sx3_testpattern.

Associated project files

S.No	SX3 configuration template	FPGA bit file name	Remarks
			Enable CAMERA_INTERFACE in parameters.v
15.	sx3_uvc_uac_colorbar_bulk_32	Colorbar_Audio_SlaveFIFO.bit	Generated from sx3_testpattern. Enable SLFIFO_INTERFACE in parameters.v
16.	sx3_uvc_uac_colorbar_isoc_32	Colorbar_Audio_SlaveFIFO.bit	Generated from sx3_testpattern. Enable SLFIFO_INTERFACE in parameters.v
17.	sx3_uvc_uac_generic_4k	hdmi_4k_project_audio_ video.bit	Generated from sx3_hdmi_4k_framebuffer
18.	sx3_uvc_uac_generic_1080p	hdmi_fullhd_project_audio_ video.bit	Generated from sx3_hdmi_1080p
19.	sx3_uvc_uac_hdmi_ite_4K	hdmi_4k_project_audio_ video.bit	Generated from sx3_hdmi_4k_framebuffer
20.	sx3_uvc_uac_hdmi_ite_1080p	hdmi_fullhd_project_audio_ video.bit	Generated from sx3_hdmi_1080p

8 Troubleshooting

- **Device not booting**
  - The SX3 device supports USB boot mode and SPI boot mode. For more details on these boot options, see the application note **[4]**.
  - The device logs are not available through the CDC interface before USB enumeration. Thus, the CDC interface cannot be used for debugging during bootup.
- **Device always enumerates in bootloader mode**
  - Check PMODE lines to make sure it is set to boot from SPI. See the SX3 datasheet **[9]** for the boot mode selection.
  - Make sure the configuration is created for the correct SX3 Variant. SX3 UVC configurations are not supported on SX3 Data variants.
  - See the sections SPI and UART and Selection of SPI Flash of the application note **[2]** to make sure that SPI flash is suitable for SX3.
  - If SPI flash is corrupted, the SX3 device will fall back to USB bootloader. SPI flash needs to be reprogrammed with the template configuration to confirm that there are no hardware-related issues.
  - SX3 does not support second stage bootloader. The boot option should be set to USB boot for device recovery.
- **Unable to download firmware**
  - Check if the PMODE setting is done for USB boot.
  - Make sure that the bootloader enumeration is successful.
  - Make sure that no errors are reported in the SX3 Configuration tool.
  - See **Firmware update** for the tools available for downloading firmware.
- **I2C errors**
  - The FIFO Master should implement the mandatory registers of the SX3 device. SX3 reports I2C errors when it does not receive an ACK from the I2C Slave. For more details on the I2C interface and register, see **I2C slave interface support on FPGA - register details**.
  - Verify if the I2C Slave address entered for the FIFO Master and the Video Source is correct.
  - The required I2C voltage levels should match the voltage levels of the VIO5 power domain. See the section I2C Interface of the application note **[2]**.
  - SX3 should not be used in multi-master I2C configuration. See the Errata section in the Sx3 datasheet **[9]**.
- **Unable to configure FPGA**
  - Verify whether the hardware connections as per Configuring FPGAs from SX3.
  - Verify if the correct bit file is provided through the SX3 Configuration Utility.
  - For Altera Cyclone 10 FPGA, make sure that the *.rbf* file is in the LSb first format.

- **Video not streaming**
  - Check whether device enumeration is successful, and the video format and frame resolution matches with the utility parameters, using the *usbview.exe* application.
  - Make sure that the UVC endpoint DMA buffer size is not a multiple of video frame size.
  - Make sure that the video resolution line size (H Resolution * bits per pixel) is a multiple of the FIFO bus width.
  - Make sure that the sensor I2C writes associated with the video resolution are correct. You can verify this using the CDC interface.

Troubleshooting

- Verify the DMA producer count, consumer count, frame rate with CDC interface Level 4. The producer count/consumer count displayed will be the number of buffers received/committed every second.
- Make sure that the frame rate calculated is based on the formula:

  ```
  Frame rate = (producer count x DMA buffer size (in bytes)) / (Hresolution x Vresolution x bytes per pixel)
  ```

- Disable the DMA watchdog option and check the video streaming to avoid periodic DMA reset sequence.

- **Audio not streaming**
  - Set the audio endpoint DMA buffer size as:

    ```
    sample width x number of channels x sampling nfrequency
    ```

  - Use the required ISOC service interval.
  - Make sure to select the correct audio source from the Windows Control Panel or Audio host application.

9 Firmware update

SX3 firmware can be updated using download option in EZ-USB SX3 Configuration Utility.

Table 16 lists the tools and drivers available for downloading SX3 firmware for various operating systems.

Table 16 Firmware update options

Operating System	Driver	Tool	Remarks
Windows	cyusb3.sys	Control center	Part of FX3 SDK
	cyusb3.sys	SX3 Configuration Utility	Generate image and program.
	Cyusb3.sys	fwdownload_fx3	Command line tool, part of FX3 SDK
	hidusb.sys	HID_Sample_App.exe	Command line tool, only available once a valid FW is loaded with HID interface enabled
Linux	libusb	SX3 Configuration Utility	Generate Image and program
		cyusb_linux	GUI-based tool, Part of FX3 SDK
		download_fx3	Command line tool, part of FX3 SDK
macOS	libusb	SX3 Configuration Utility	Generate image and program
		cyusb_linux	GUI-based tool, part of FX3 SDK
		download_fx3	Command line tool, part of FX3 SDK

References

[1] **AN65974 - Designing with the EZ-USB FX3 Slave FIFO Interface**

[2] **AN70707 - EZ-USB® FX3™/FX3S™ Hardware Design Guidelines and Schematic Checklist**

[3] **AN70983 - Designing a Bulk Transfer Host Application for EZ-USB® FX2LP™/FX3™**

[4] **AN76405 - EZ-USB® FX3™/FX3S™ Boot Options**

[5] **Technical Note TN1260 - ECP5 and ECP5-5G sysCONFIG Usage Guide**

[6] **FPGA-TN-02014-1.2 - CrossLink Programming and Configuration Usage Guide**

[7] **UG470 - 7 Series FPGAs Configuration User Guide**

[8] **Intel C10LP51003 - Intel® Cyclone® 10 LP Core Fabric and General Purpose I/Os Handbook**

[9] **EZ-USB SX3 Datasheet**

[10] **Universal Serial Bus Specification for Video Devices Version 1.5**

[11] **Universal Serial Bus Device Class Definition for Audio Devices**

Revision history

Document version	Date of release	Description of changes
**	2021-03-31	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2021-03-31
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to: **www.cypress.com/support**

Document reference
002-31295 Rev.**

IMPORTANT NOTICE
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.