# 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 (SS) peripheral controller, providing integrated and flexible features. SX3 has a fully configurable, parallel interface known as 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

Table of contents

# 1 SX3 variants and features

SX3 has integrated the USB 3.2 Gen 1 and USB 2.0 physical layers (PHYs) along with a 32-bit ARM926EJ-S microprocessor for powerful data processing and building custom applications. It implements an architecture that enables up to 375 MBps data transfer from the 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 $I^2C$ 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 $I^2C$ 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), and 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 $I^2C$ interface to communicate with peripherals
- Supports SPI Flash for firmware and configuration storage
- Supports up to seven configurable GPIOs

## 2 More information

Infineon provides a wealth of data at **www.infineon.com** to help you to select the right device for your design, and to help you to integrate the device into your design quickly and effectively.

- Overview: **USB portfolio**
- USB 3.0 product selectors: **EZ-USB™ SX3**, **EZ-USB™ FX3**, **EZ-USB™ FX3S**, **EZ-USB™ CX3**, **EZ-USB™ HX3**
- Application notes: Infineon offers a large number of USB application notes covering a broad range of topics, from basic to advanced level. Recommended application notes for getting started with EZ-USB™ SX3 are:
  - **AN70707** – EZ-USB™ FX3/FX3S/SX3 hardware design guidelines and schematic checklist
  - **AN65974** – Designing with the EZ-USB™ FX3 slave FIFO interface
  - **AN75779** – How to implement an image sensor interface with EZ-USB™ FX3 in a USB Video Class (UVC) framework
- Knowledge base articles (KBA)
  - EZ-USB™ SX3 HDMI 4K Capture Card Solution Demo Kit - **KBA233573**
- Development kits:
  - **CYUSB3KIT-004, EZ-USB™ SX3 explorer kit**

# 3 Hardware design guidelines

## 3.1 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 I$^2$C |
| 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 I$^2$C interface | Programmable I$^2$C 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 |

## 3.2 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 the 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 of the FIFO master.
- SUSPEND_OUT: This I/O is output of SX3, used by the FIFO master to enter low-power mode.

## 3.3 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 an I$^2$C write to a specific register in the FIFO master. **Table 2** lists the I$^2$C 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.

## 3.4 Interfacing FIFO master (FPGA) image sensor with SX3

SX3 supports slave FIFO interface to communicate with the 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 or 32 bits. For details on the signals of these interfaces, see the Sx3 datasheet [9].

### 3.4.1 Interfacing FIFO master (FPGA) with SX3

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

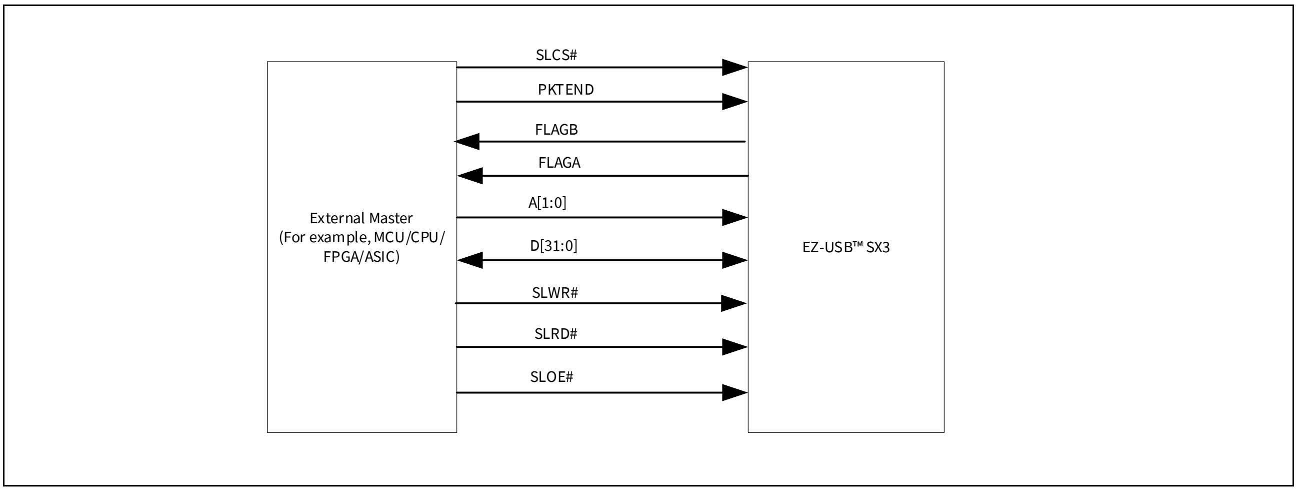


Figure 1 Slave FIFO interface

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 I$^2$C interfaces to download FPGA configuration. SX3 uses the I$^2$C interface to communicate with FPGA after configuration.

### 3.4.2 Parallel camera interface

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

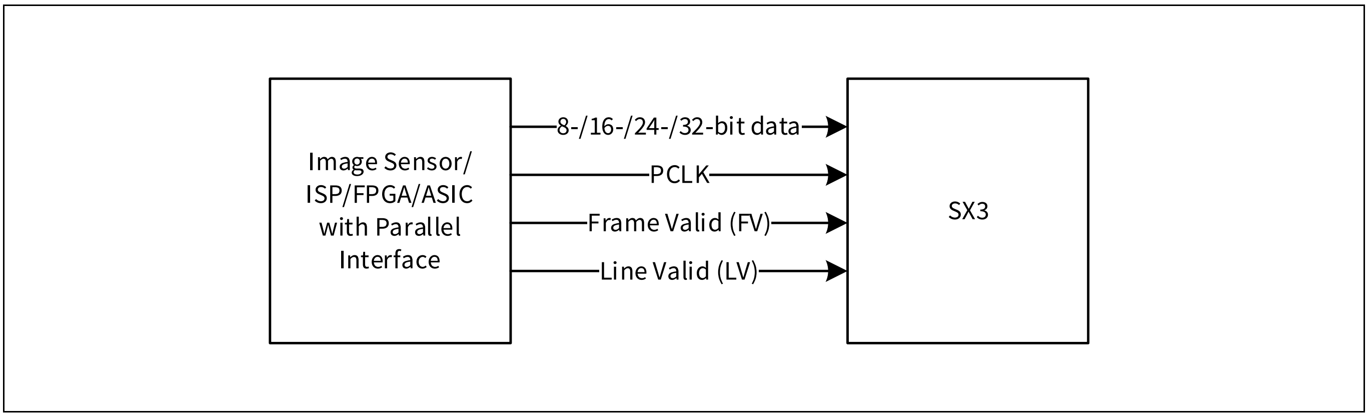


Figure 2 Parallel camera interface

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

### 3.4.3 Configuring FPGAs from SX3

SX3 supports the following configuration modes:

- Lattice slave SPI (SSPI) mode
- Lattice I$^2$C 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 I$^2$C 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**.

### 3.4.3.1 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.

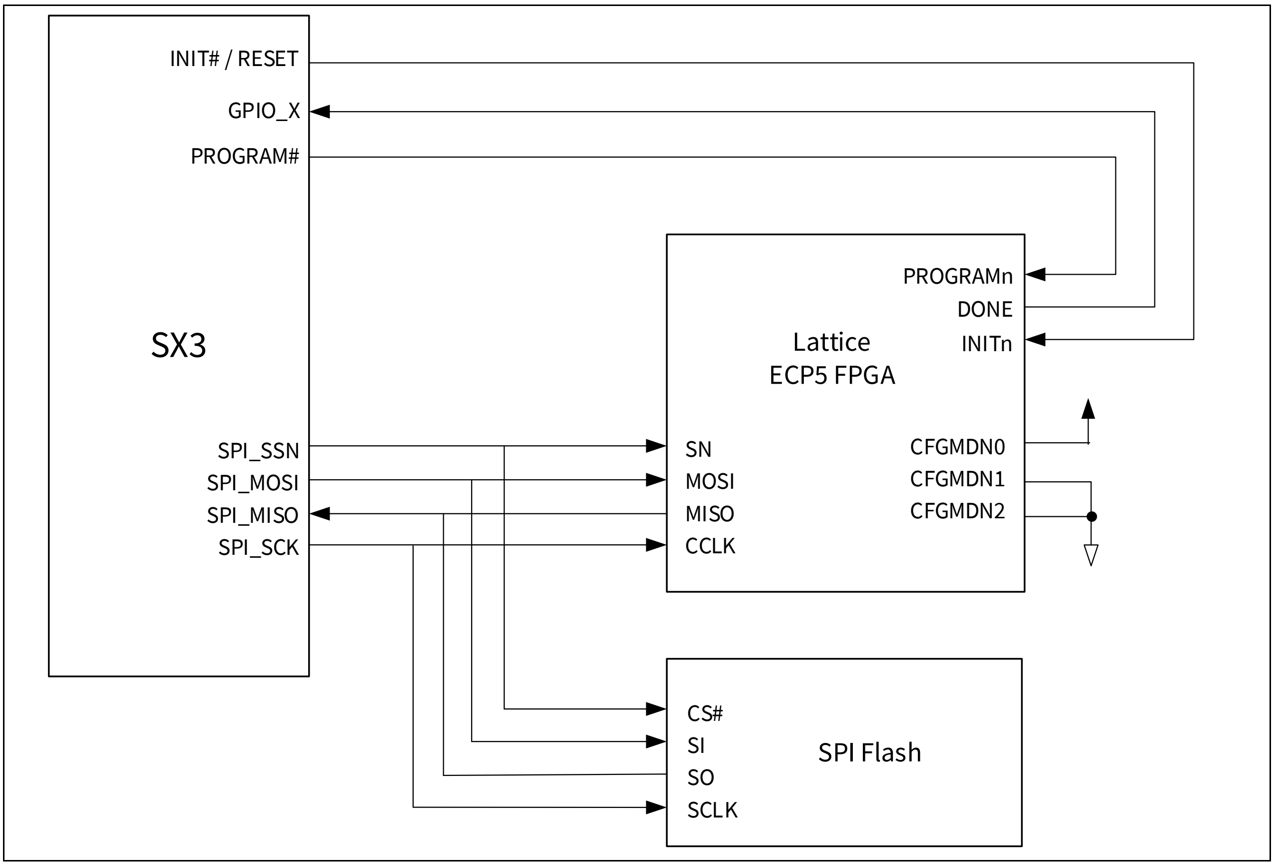


**Figure 3 ECP5 programming with slave SPI port and SX3**

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

### 3.4.3.2 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].

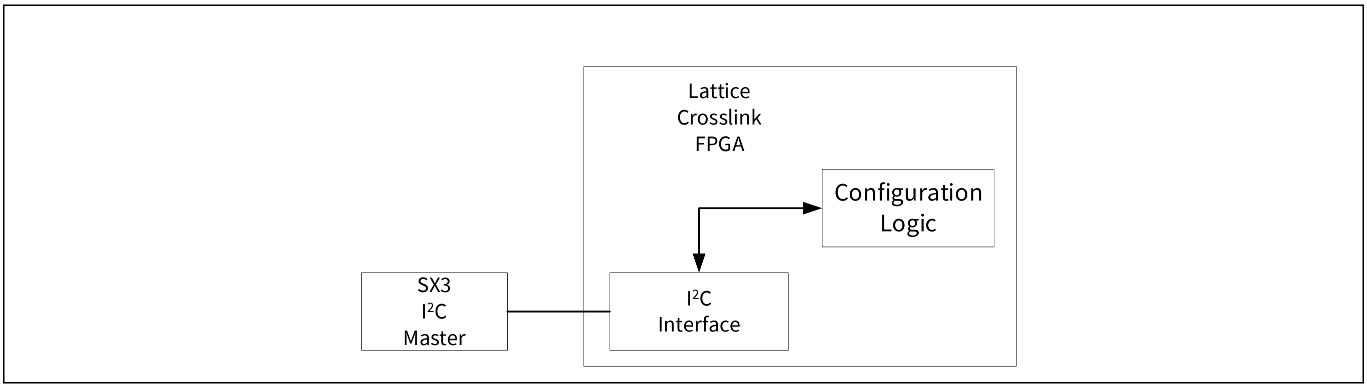


**Figure 4 I²C configuration logic**

### 3.4.3.3 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].

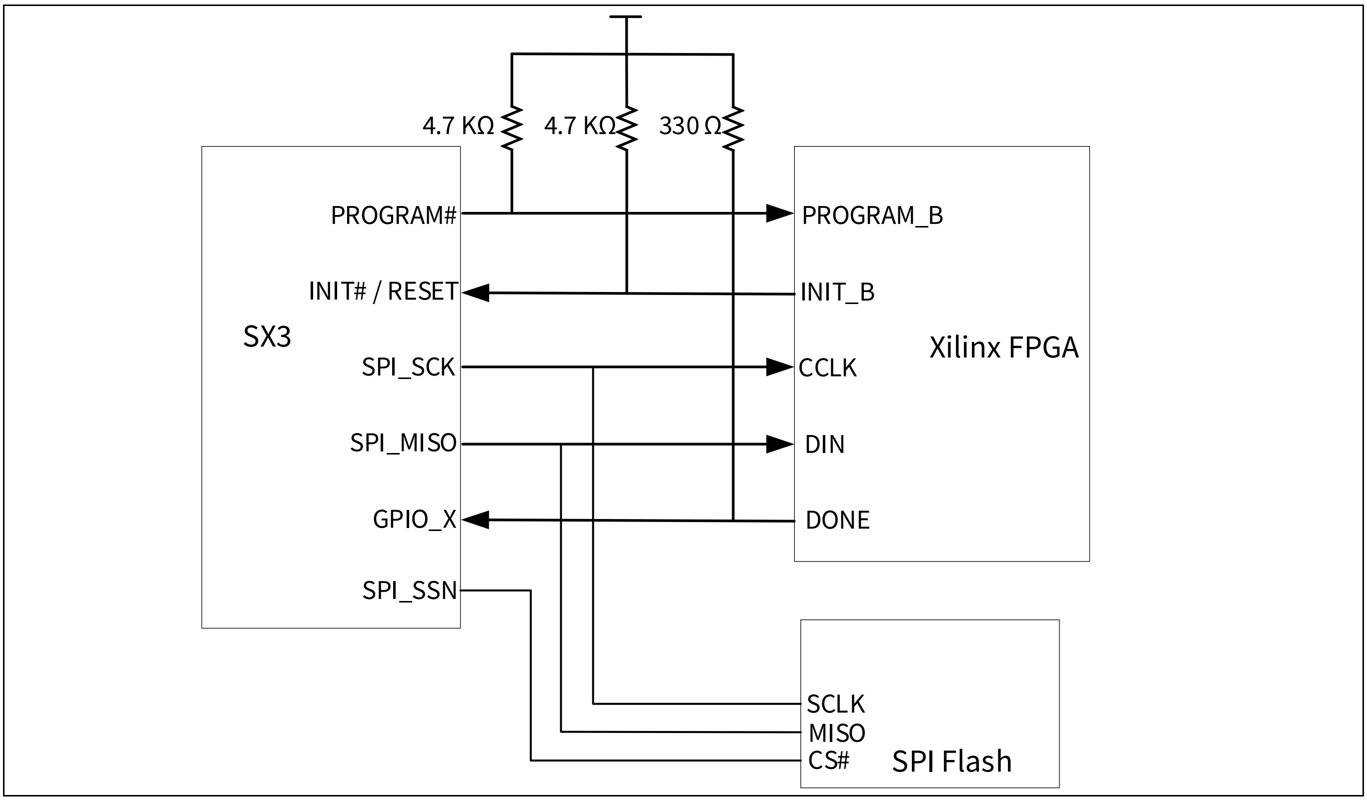


**Figure 5** **Slave serial mode**

### 3.4.3.4 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.

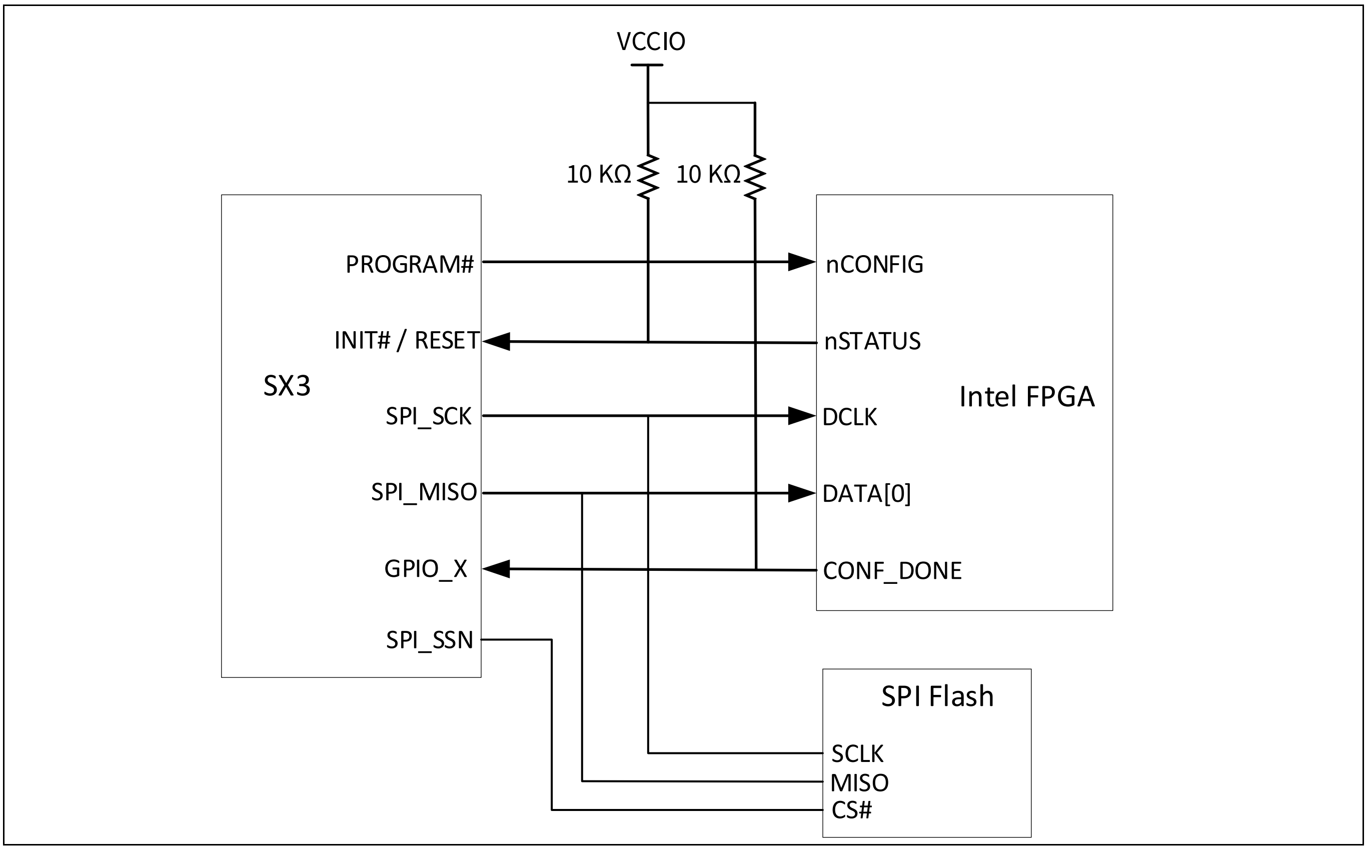


**Figure 6 Passive serial mode**

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 |

## 3.5 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 download 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]**.

### 3.5.1 Image sensor interface

The initialization of I$^2$C register configuration can be written to the image sensor over the I$^2$C interface.

SX3 supports writing I$^2$C register configuration corresponding to the video resolution each time the resolution is selected in the UVC host application.

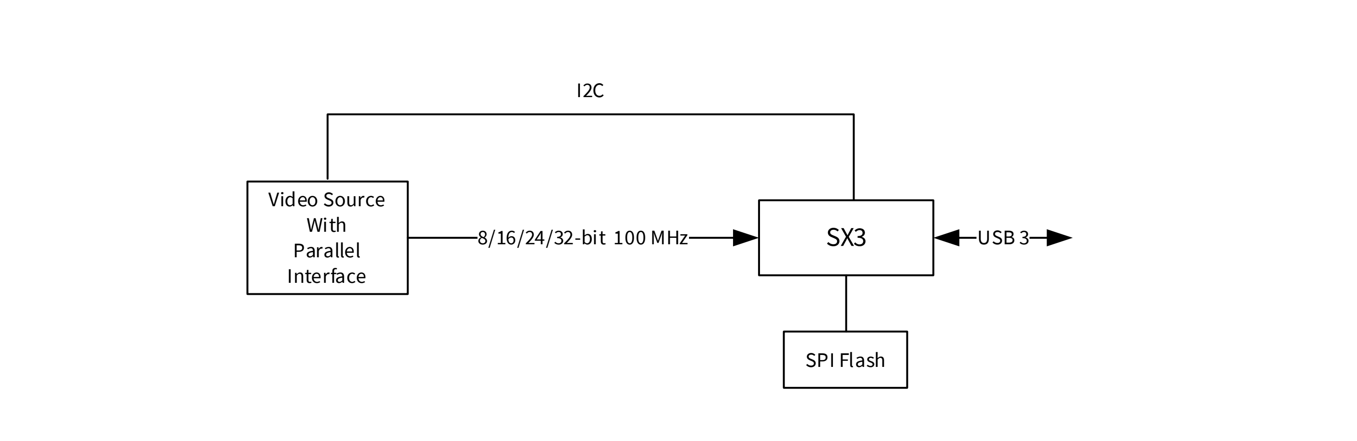


**Figure 7 Direct image sensor interface with SX3**

*Note:*

1. *For more details on Hardware design using SX3, refer to* **AN70707-EZ-USB™ FX3/FX3S/SX3 hardware design guidelines and schematic checklist**.

# 4 Applications

## 4.1 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 high-speed mode, only bulk configuration is supported. The following sections describe typical application examples for SX3 using these configurations.

### 4.1.1 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.

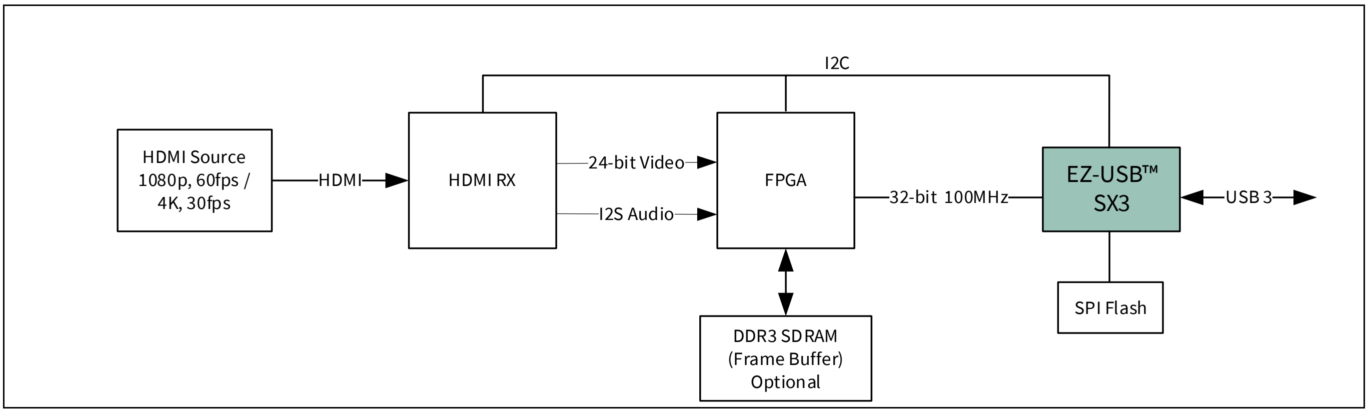


**Figure 8** **4K HDMI USB3 capture card**

The HDMI receiver converts the HDMI video and audio stream to parallel video interface and I$^2$S audio interface. A FIFO master (FPGA) is required to convert the data and send it to SX3 over the slave FIFO interface. The FIFO master supports frame buffer, which is required for supporting 4K video stream. SX3 uses the I$^2$C interface to communicate with the HDMI RX and FIFO master.

#### 4.1.1.1 HDMI as Video source

The SX3 configuration tool supports two types of HDMI receivers as Video Source:

- 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 I$^2$C interface. | HDMI initialization and EDID configuration can be done using I$^2$C interface. |

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

#### 4.1.1.2 Video and audio support

SX3 can support up to 4K 30 fps uncompressed video along with audio of up to 32-bit, 12 channels. FPGA implements an $I^2S$ receiver and sends out the data over the slave FIFO interface.

#### 4.1.1.3 HDMI event handling in generic HDMI configuration

SX3 supports HDMI generic configuration to support interface with a generic HDMI receiver with the USB device controller. Add the HDMI events to be handled by the USB device controller in the Event Handling Table of the SX3 Configuration Utility. SX3 will detect the events and perform $I^2C$ writes to handle the specific event. Add the HDMI interrupt event register, event mask and $I^2C$ structure to be written in the specific format mentioned in the **Help content** tab of the SX3 Configuration Utility and the SX3 Configuration Utility user guide.

#### 4.1.1.4 FPGA project for interfacing HDMI RX to SX3

An example FPGA project for interfacing the 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 the 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 two-channel, 16-bit sample width, and 48 kHz. The $I^2S$ 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

### 4.1.1.5 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.

### 4.1.1.6 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.

### 4.1.1.7 I$^2$C slave interface support on FPGA – register details

SX3 uses the fixed I$^2$C registers, listed in **Table 6**, which are to be implemented in FPGA. FPGA should implement I$^2$C 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 |

| 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 to YUV420 conversion |
| | | | | 0x00 – disables YUV422 to 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. The FIFO master is expected to clear the value to "0" after reading the set value.*

### 4.1.1.8 Development kit: e-CON systems SX3 FPGA HDMI RX kit (PICTOR)

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

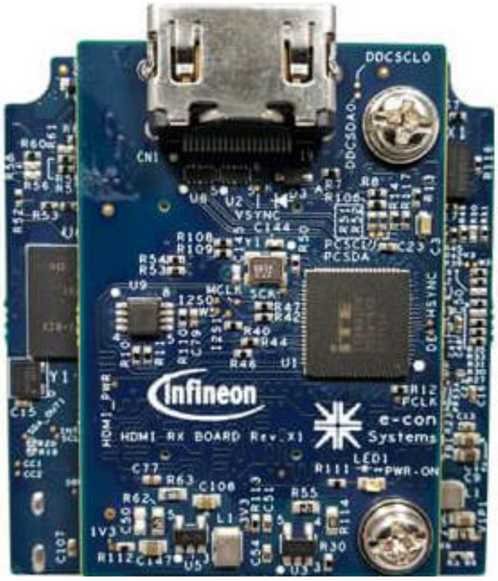


**Figure 9 SX3 HDMI USB3 capture card kit (PICTOR)**

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

### 4.1.1.9 EZ-USB™ SX3 HDMI 4K Capture Solution Demo kit (CY-SD4210)

The EZ-USB™ SX3 HDMI 4K Capture Card solution demo kit is a USB Video Class (UVC), USB Audio Class (UAC) compliant capture card which is ideally suited for capturing video and audio from any HDMI source. It supports video streaming up to 4K, 30fps in YUV420 format or up to 1080p, 60fps in YUV422 format.

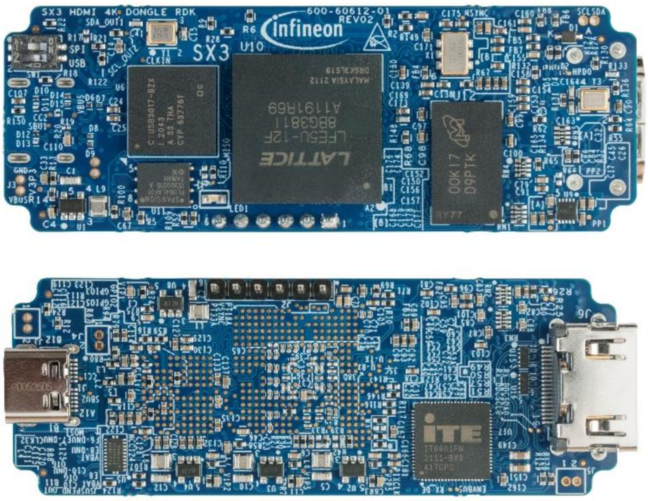


**Figure 10 SX3 HDMI 4K Capture Solution Demo kit (CY-SD4210)**

The following example configurations can be evaluated using this kit:

- SX3_UVC_UAC_HDMI_ITE_1080P_DONGLE
- SX3_UVC_UAC_HDMI_ITE_4K_DONGLE

Note that these template configurations are provided along with SX3 Configuration Utility (v1.1.0.5 and above).

The kit schematic, BOM, and board files are available as a **Knowledge Base Article** and also on **SX3 product webpage**. Customers can build customized boards using these files.

### 4.1.2 UVC camera application using image sensor and FPGA

The SX3 UVC variant can also be used to design UVC applications where an image sensor is connected to an ISP/FPGA. **Figure 11** is an example where an image sensor is interfaced with SX3 using FPGA.

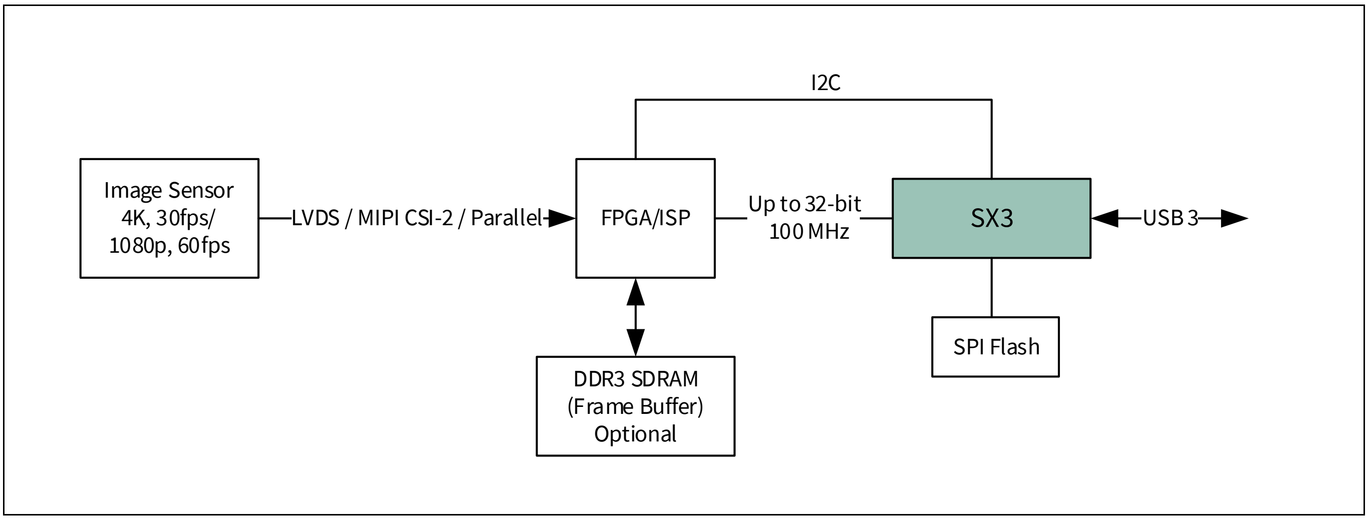


**Figure 11 USB3 camera using SX3**

#### 4.1.2.1 Development kit: EZ-USB™ SX3 Explorer Kit (CYUSB3KIT-004)

The EZ-USB™ SX3 explorer kit combines hardware, software, and documentation to evaluate the SX3 device. This kit is intended to be a development kit for USB audio and video streaming applications using SX3.

The CYUSB3KIT-004 EZ-USB™ SX3 explorer kit consists of two boards:

- SX3 Baseboard: Consists of the SX3 device, an external SPI flash module, power supplies, USB Type – C connector, and interface connectors for add-on modules.
- Camera add-on board: Consists of the Crosslink FPGA from Lattice Semiconductor, onboard OV5640 Image sensor module, microphones, power supplies, and external interface connector for Raspberry Pi compatible image sensor modules.

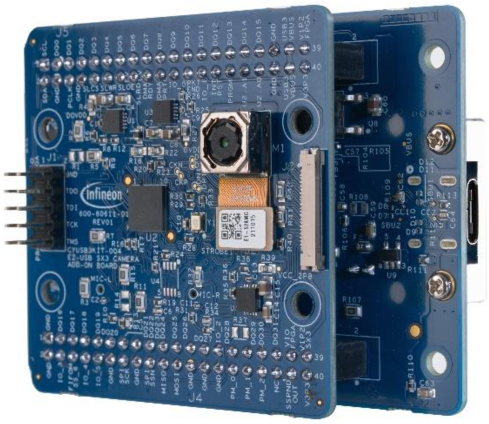


**Figure 12 EZ-USB SX3 Explorer Kit (CYUSB3KIT-004)**

The CYUSB3KIT-004 EZ-USB™ SX3 explorer kit support the following key features:

- Video streaming up to 1080p 30fps in YUV format from onboard image sensor module
- Audio stream from onboard L/R digital MEMS microphones
- External off-the-shelf MIPI camera interface from Raspberry Pi
- Compatible with FX3 explorer kit interconnect boards
- USB bus-powered operation
- Firmware-controlled LED and user switch
- GPIO headers

The following example configurations can be evaluated using this kit:

- SX3_EXPLORER_KIT_UVC_UAC_CROSSLINK_SLAVEFIFO
- SX3_EXPLORER_KIT_UVC_UAC_CROSSLINK_RPI

These example projects demonstrate video streaming up to YUV 1080p 60fps and audio streaming from the onboard stereo microphones. Note that these template configurations are provided along with SX3 Configuration Utility (v1.1.0.5 and above).

#### 4.1.2.2 Configuring FPGA

SX3 will configure the Lattice CrossLink FPGA using I$^2$C interface on bootup.

#### 4.1.2.3 Configuring image sensor

SX3 will configure the OV5640 sensor using I$^2$C interface.

### 4.1.3 Direct interface of image sensor

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

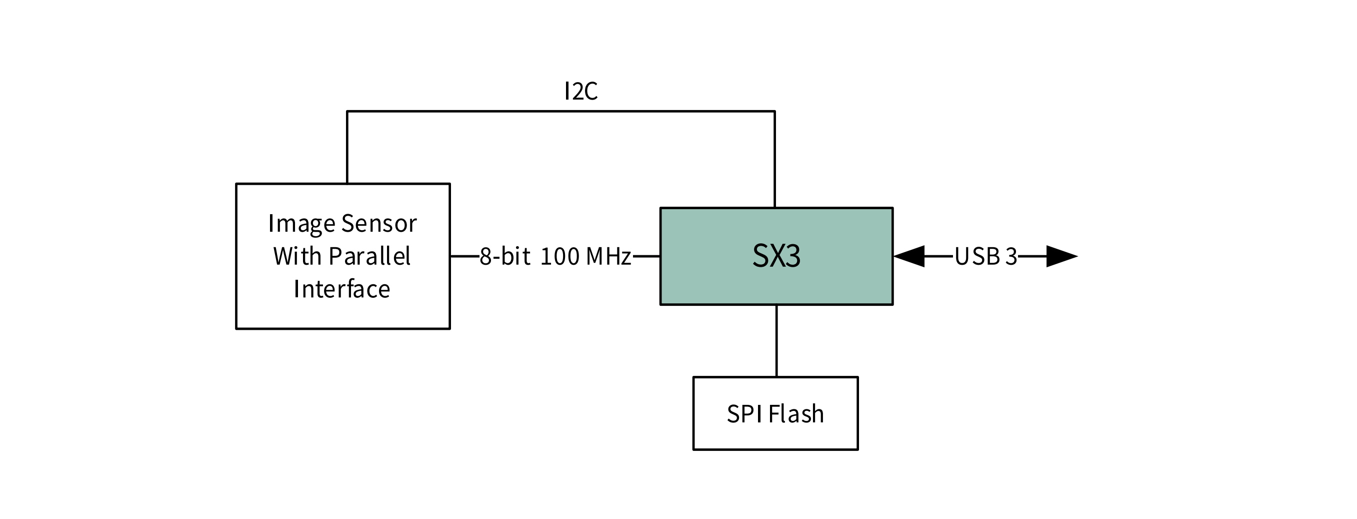


**Figure 13 Direct image sensor interface to SX3**

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

- SX3_UVC_FV_LV_BULK_32

### 4.1.4 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.

### 4.1.5 Driver requirement and multi-OS support

SX3 UVC is supported by the inbuilt 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 | Inbuilt driver |

## 4.2 Data application

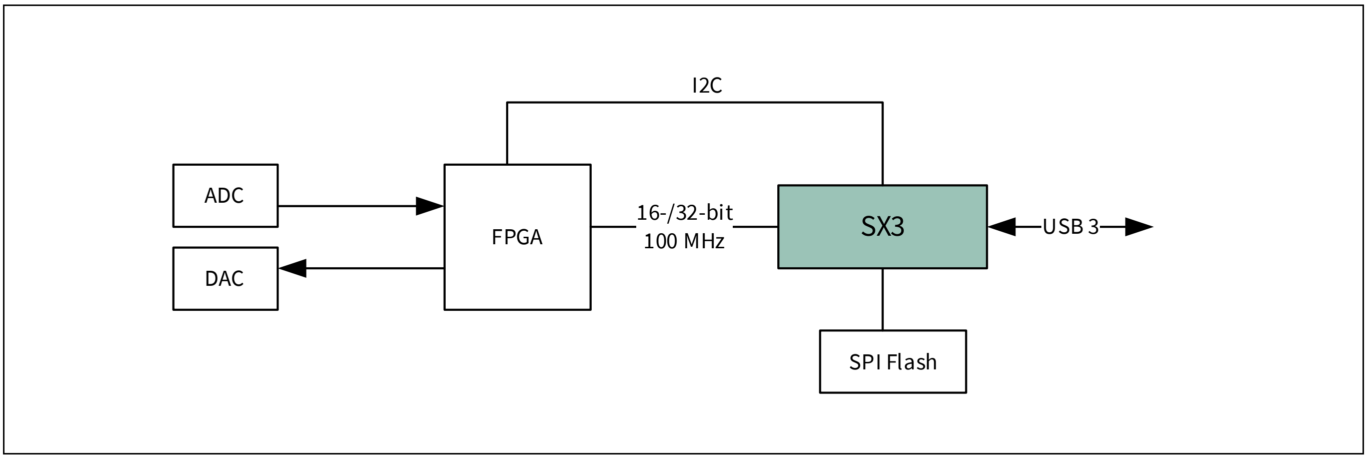


**Figure 14 SX3 data interfaced to FPGA**

SX3 data – 16-bit (CYUSB3015), SX3 data – 32-bit (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.

### 4.2.1 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 super-speed and high-speed 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 |
|---|---|---|---|---|
| | | Super-speed | High-speed | |
| SX3 data – 32-bit | IN | 390 MBps | 46 MBps | Streamer |
| SX3 data – 32-bit | OUT | 389 MBps | 42 MBps | Streamer |
| SX3 data – 32-bit | IN + OUT | N/A | N/A | Bulkloop |
| SX3 data – 16-bit | IN | 195 MBps | 46 MBps | Streamer |
| SX3 data – 16-bit | OUT | 192 MBps | 42 MBps | Streamer |
| SX3 data – 16-bit | IN + OUT | N/A | N/A | 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.*

The following SX3 template projects are part of the SX3 configuration tool and can be evaluated using the SX3 FPGA baseboard, which is part of the 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 the SX3 UVC (CYUSB3017) part number.

The SX3 data – 32-bit (CYUSB3016) and SX3 data – 16-bit (CYUSB3015) devices support vendor commands to enable/disable USB low-power mode (LPM). LPM should be disabled in SX3 during data transfer and enabled once the data transfer is completed. **Table 10** lists the vendor commands.

**Table 10 Vendor commands**

| Command | bmRequestType | bRequest | wValue | wIndex | wLength |
|---|---|---|---|---|---|
| Disable LPM | 0x40 | 0xEA | 0x0000 | 0x0000 | 0x0000 |
| Enable LPM | 0x40 | 0xEA | 0x0001 | 0x0000 | 0x0000 |

The host application should support the vendor commands, listed in **Table 10**, to handle LPM.

### 4.2.2 Host applications – SX3 data

The SX3 data variants can be tested with different host applications such as Streamer, Control Center, Bulkloop and so on. The host applications are available in the *tools* folder in the installation path of the SX3 Configuration Utility. For more details on creating custom host applications, see the application note [3].

**Table 11** lists the host applications that can be used for each operating system.

**Table 11 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 |

### 4.2.3 Driver requirement and multi-OS support – SX3 data

The template projects use Infineon 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.

# 5 Configuration utility

The EZ-USB™ SX3 Configuration Utility is a software application that can be used to configure the 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.

## 5.1 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 the 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

## 5.2 Installing Configuration Utility

You can download and install this tool from the **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 it.
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.

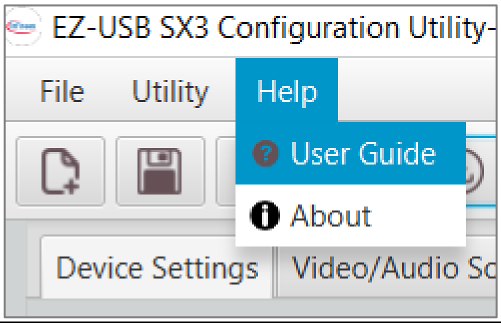


**Figure 15 Help menu**

**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 12**.

**Table 12 Supported drivers**

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

### 5.2.1 Installing the 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 Windows Driver Installation section in the EZ-USB™ SX3 Configuration Utility user guide.

### 5.2.2 Installing the 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.

### 5.2.3 Installing the 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.

## 5.3 File storage

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

## 5.4 How the SX3 Configuration Utility works

The SX3 Configuration Utility generates the 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, the configuration parameters and the FPGA bit file (if included). This file will be downloaded to the SPI Flash during programming.

  See **Figure 16** for more details.

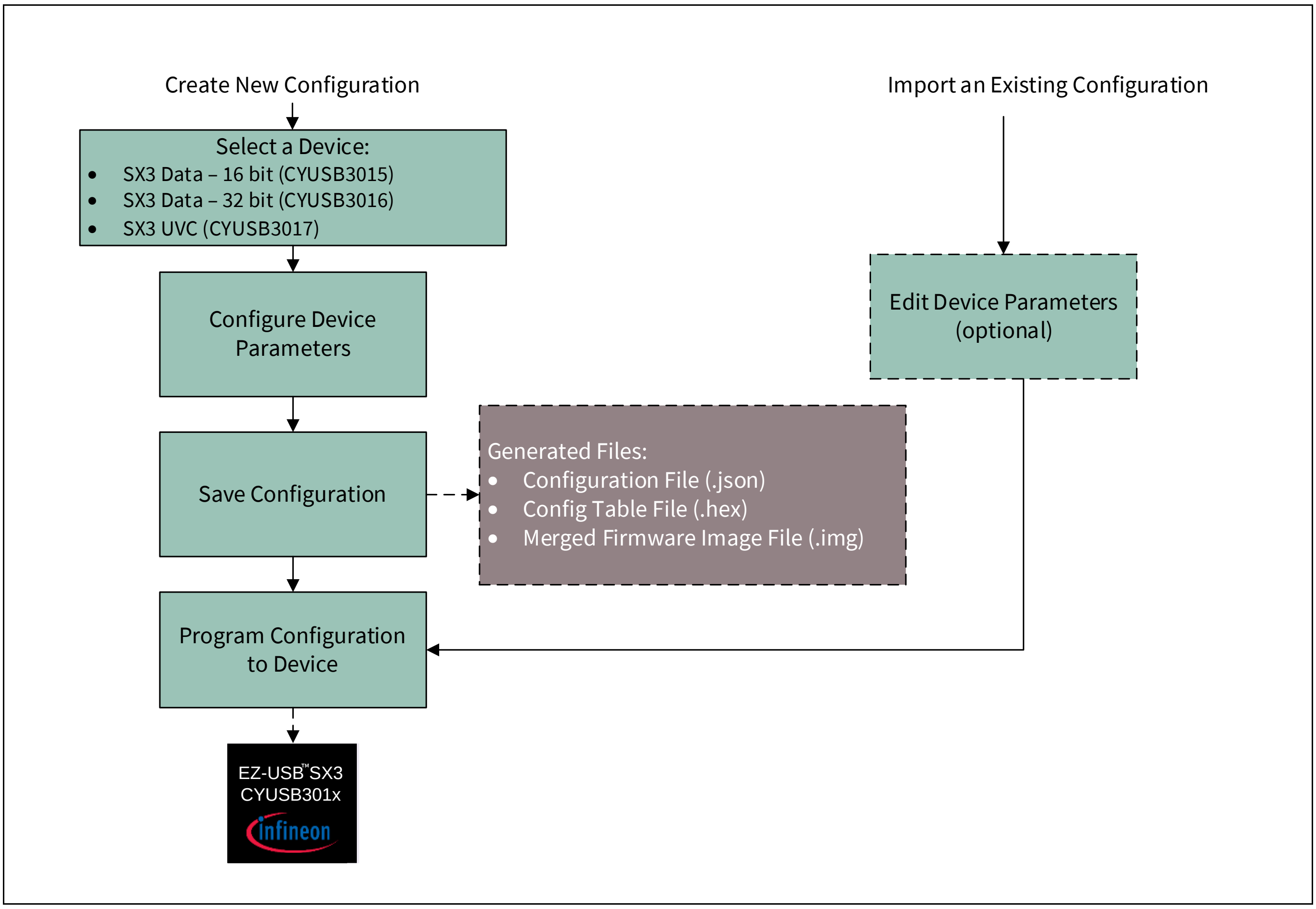

Create New Configuration
Import an Existing Configuration
Select a Device:
SX3 Data – 16 bit (CYUSB3015)
SX3 Data – 32 bit (CYUSB3016)
SX3 UVC (CYUSB3017)
Configure Device Parameters
Edit Device Parameters (optional)
Save Configuration
Generated Files:
Configuration File (.json)
Config Table File (.hex)
Merged Firmware Image File (.img)
Program Configuration to Device
EZ-USB™SX3
CYUSB301x
infineon

**Figure 16 How the SX3 Configuration Utility works**

### 5.4.1 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 the address location after the SX3 configuration.

### 5.4.2 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**.

Note that the SX3 Configuration Utility supports programming the generated configuration to EZ-USB™ FX3 devices also.

# 6 Interfacing custom devices with SX3

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

# 7 Debugging SX3-based designs

## 7.1 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, $I^2C$ 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 are based on the selected debug level.

### 7.1.1 Debugging levels

The SX3 CDC interface supports five debug levels. **Table 13** lists the debug logs available for each level.

**Table 13 SX3 debug levels**

| Debug level | 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 $I^2C$ writes |
| Level 3 | Error messages, start/stop messages, suspend/resume status, UVC commands, all $I^2C$ writes, HDMI event log |
| Level 4 | Error messages, start/stop messages, suspend/resume status UVC commands, $I^2C$ writes, HDMI event log, DMA statistics (producer count, consumer count, fps and so on) |

### 7.1.2 Debug terminal applications

You can use any standard debug terminal application to collect the log from the SX3 device. **Table 14** lists the sample host applications that can be used on different operating systems.

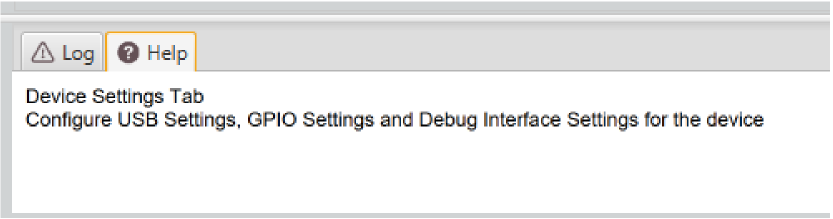
**Table 14 Debugging terminal applications**

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

*Note: No specific flow control or baud rate 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.*

## 7.2 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.



Device Settings Tab
Configure USB Settings, GPIO Settings and Debug Interface Settings for the device

**Figure 17 Displaying the Help tab**

## 7.3 SX3 Configuration Utility user guide

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

# 8 Associated project files

**Table 15** lists the links to the source code for the FPGA projects and sample applications available in the GitHub repository.

**Table 15 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 30 fps. |
| **sx3_hdmi_1080p** | FPGA project source code to interface HDMI receiver with SX3 without frame buffer implementation to support video resolutions of up to 1080 p 30 fps. |
| **sx3_data_slavefifo_example** | FPGA source code that can be used to test SX3 data parts (CYUSB3015/CYUSB3016).<br>Three modes are supported:<br>• IN only<br>• OUT only<br>• IN + OUT in loopback mode<br>I$^2$C 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 I$^2$C 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, or UVC + UAC configurations.<br>The interface between SX3 and FPGA can be either of the following:<br>• Slave FIFO mode – supports audio and video pattern with data bus width of 8/16/24/32 bits<br>• Camera parallel interface mode – supports video only with bus width of 8/16/24/32 bits<br>The interface mode and bus width can be selected with a compile time switch in the FPGA project. |
| **sx3_explorer_kit_uvc_uac_crosslink_slavefifo** | FPGA project source code to demonstrate video and audio streaming with the SX3 Explorer kit. This project is tested with the Lattice Crosslink FPGA on the SX3 Explorer Kit's Camera Add-on board. The FPGA receives data over MIPI from the onboard OV5640 image sensor and streams it to the SX3 over a 16-bit standard slave FIFO interface. It also streams the audio data from the stereo microphones on the same board to SX3. |
| **sx3_hid_sample_app** | A sample HID command-line application for Windows OS that can be used for firmware update over HID interface. |

**Associated project files**

| File/Folder name | Description |
|---|---|
|  | The application takes VID, PID and firmware images as inputs and updates the SPI Flash connected to SX3.<br>The HID application also supports the command to erase SPI Flash and fall back to the 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 15**.

**Table 16** lists the FPGA bit files used along with example configuration templates and how they can be generated from the FPGA projects listed in **Table 15**.

**Table 16 FPGA bit files and example configuration templates**

| Number | 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_16 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 | 30ilinx_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 |

| Number | SX3 configuration template | FPGA bit file name | Remarks |
|---|---|---|---|
| | | | Enable STREAM_OUT_ONLY, GPIF_WDT_32 in params.v |
| 9 | sx3_uac | testptrn_proj_slfifo_audonly.bit | Generated from sx3_testpattern<br>Enable SLFIFO_INTERFACE, SLFIFO_INTERFACE_AUD in parameters.v |
| 10 | sx3_uvc_24_bit_isoc | sx3_uvc_24bit_isoc.bit | Generated from sx3_testpattern.<br>Enable SLFIFO_INTERFACE in parameters.v,<br>Change INIT_GPIF_WDT value to 3 in i2c_slave.v |
| 11 | sx3_uvc_fv_lv_bulk_32 | testptrn_proj_camera_ interface.bit | Generated from sx3_testpattern.<br>Enable CAMERA_INTERFACE in parameters.v |
| 12 | sx3_uvc_uac_colorbar_bulk_32 | Colorbar_Audio_SlaveFIFO.bit | Generated from sx3_testpattern.<br>Enable SLFIFO_INTERFACE in parameters.v |
| 13 | sx3_uvc_uac_colorbar_isoc_32 | Colorbar_Audio_SlaveFIFO.bit | Generated from sx3_testpattern.<br>Enable SLFIFO_INTERFACE in parameters.v |
| 14 | sx3_uvc_uac_generic_4k | hdmi_4k_project_audio_ video.bit | Generated from sx3_hdmi_4k_framebuffer |
| 15 | sx3_uvc_uac_generic_1080p | hdmi_fullhd_project_audio_ video.bit | Generated from sx3_hdmi_1080p |
| 16 | sx3_uvc_uac_hdmi_ite_4K | hdmi_4k_project_audio_ video.bit | Generated from sx3_hdmi_4k_framebuffer |
| 17 | sx3_uvc_uac_hdmi_ite_1080p | hdmi_fullhd_project_audio_ video.bit | Generated from sx3_hdmi_1080p |

# 9 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.
- **I$^2$C errors**
  - The FIFO master should implement the mandatory registers of the SX3 device. SX3 reports I$^2$C errors when it does not receive an ACK from the I$^2$C slave. For more details on the I$^2$C interface and register, see **I2C slave interface support on FPGA – register details**.
  - Verify if the I$^2$C slave address entered for the FIFO master and the video source is correct.
  - The required I$^2$C voltage levels should match the voltage levels of the VIO5 power domain. See the section I$^2$C 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 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 that the video format and frame resolution match the utility parameters, using the *usbview.exe* application.
  - Make sure the UVC endpoint DMA buffer size is not a multiple of video frame size.
  - Make sure the video resolution line size (H resolution * bits per pixel) is a multiple of the FIFO bus width.
  - Make sure the sensor I$^2$C writes associated with the video resolution are correct. You can verify this using the CDC interface.
  - 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 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.

# 10 Firmware update

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

**Table 17** lists the tools and drivers available for downloading SX3 firmware for various operating systems.

**Table 17 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

## Revision history

| Document version | Date of release | Description of changes |
|---|---|---|
| ** | 2021-03-31 | Initial release |
| *A | 2021-04-09 | Updated the links in **Table 15**<br>Changed macro in **Table 15**<br>Updated section **4.2.1** with vendor commands for USB LPM |
| *B | 2022-08-01 | Added **More information** section<br>Added details on SX3 Explorer Kit and SX3 HDMI 4K Capture Solution Demo Kit (CY-SD4210) |

**Trademarks**

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

**Edition 2022-08-01**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2022 Infineon Technologies AG.**
**All Rights Reserved.**

**Do you have a question about this document?**

**Go to www.infineon.com/support**

**Document reference**

**002-31295 Rev. *B**

**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.