# EZ-USB SX3 Configuration Utility User Guide

## About this document

### Scope and purpose

This document describes the usage of EZ-USB™ SX3 Configuration Utility, a cross-platform application that guides through the process of configuring and programming all SX3 variants.

### Intended audience

This document is intended for embedded developers using EZ-USB SX3 for their end application

### Abbreviations and definitions

**Table 1 Abbreviations**

| Abbreviation | Definition |
|---|---|
| GUI | Graphical User Interface |
| USB | Universal Serial Bus |
| UAC | USB Audio Class |
| UVC | USB Video Class |
| OS | Operating System |

# Table of contents

**About this document** ..... **1**
**Table of contents** ..... **2**
**1 Introduction** ..... **3**
1.1 System requirements and prerequisites ..... 3
1.2 Prerequisites ..... 3
1.3 Installation ..... 3
1.4 Technical support ..... 4
**2 EZ-USB SX3 configuration utility** ..... **5**
2.1 Features ..... 5
**3 Default view of the configuration utility** ..... **6**
**4 Help tab** ..... **7**
**5 Creating a new configuration** ..... **8**
**6 Importing an existing configuration** ..... **10**
**7 Video/audio source configuration** ..... **14**
7.1 Import to table from text field ..... 14
7.2 Import to table from text file ..... 15
**8 HDMI generic event handling** ..... **17**
**9 Programming the device** ..... **18**
**10 Error reporting** ..... **19**
10.1 Pop-up error messages ..... 19
10.2 Errors in log window ..... 19
**11 Log window** ..... **20**
11.1 Saving the log messages ..... 20
11.2 Clearing the log ..... 20
**12 Example configuration – SX3 UVC UAC HDMI ITE 4K** ..... **21**
**13 Frequently asked questions / troubleshooting** ..... **26**
**14 Appendix** ..... **27**
14.1 Binding to CYUSB3 driver on Windows for SX3 - data variants ..... 27
**Revision history** ..... **32**

# 1 Introduction

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

## 1.1 System requirements and prerequisites

**Windows OS**: Windows 10

**Linux**: Ubuntu 10

**macOS**: High Sierra

The SX3 configuration utility may work on earlier versions of Windows, Linux and macOS, but it is not tested on those versions.

## 1.2 Prerequisites

The SX3 configuration tool uses cyusb3 driver in Windows and libusb in macOS and Linux to program connected SX3 devices.

The cyusb3 drivers for Windows are packaged along with the SX3 configuration utility and is available in the *drivers* folder at the installation path. See **Appendix** to learn how to manually bind to this driver.

Install libusb by using the following commands:

On Linux:

```
sudo apt-get install libusb-1.0-0-dev
```

On macOS:

```
brew install libusb
```

## 1.3 Installation

The SX3 configuration tool can be installed from **www.cypress.com/sx3**. Installable packages are available for Windows, Linux, and macOS.

The following directories will be available after installation:

| Folder | Description |
|---|---|
| *Docs* | Contains the user guide |
| *Drivers* | Contains Infineon USB drivers for all Windows Versions |
| *Firmware* | Contains SX3 firmware images |
| *Lib* | Contains the executable JAR libraries |
| *Templates* | Contains various SX3 template configurations along with the required bit files |
| *Tools* | Contains required tools for various operating systems |
| *Utils* | Contains files which used to detect both Infineon driver and programming utility |

## 1.4 Technical support

For assistance, go to **www.cypress.com/support** or create a post in **Cypress Developer Community**.

# 2 EZ-USB SX3 configuration utility

## 2.1 Features

- Supported on Windows, Linux, and macOS
- Supports configuration of SX3-UVC (CYUSB3017) and SX3-Data (CYUSB3015 and CYUSB3016) variants
- Supports generation of new configuration and importing 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

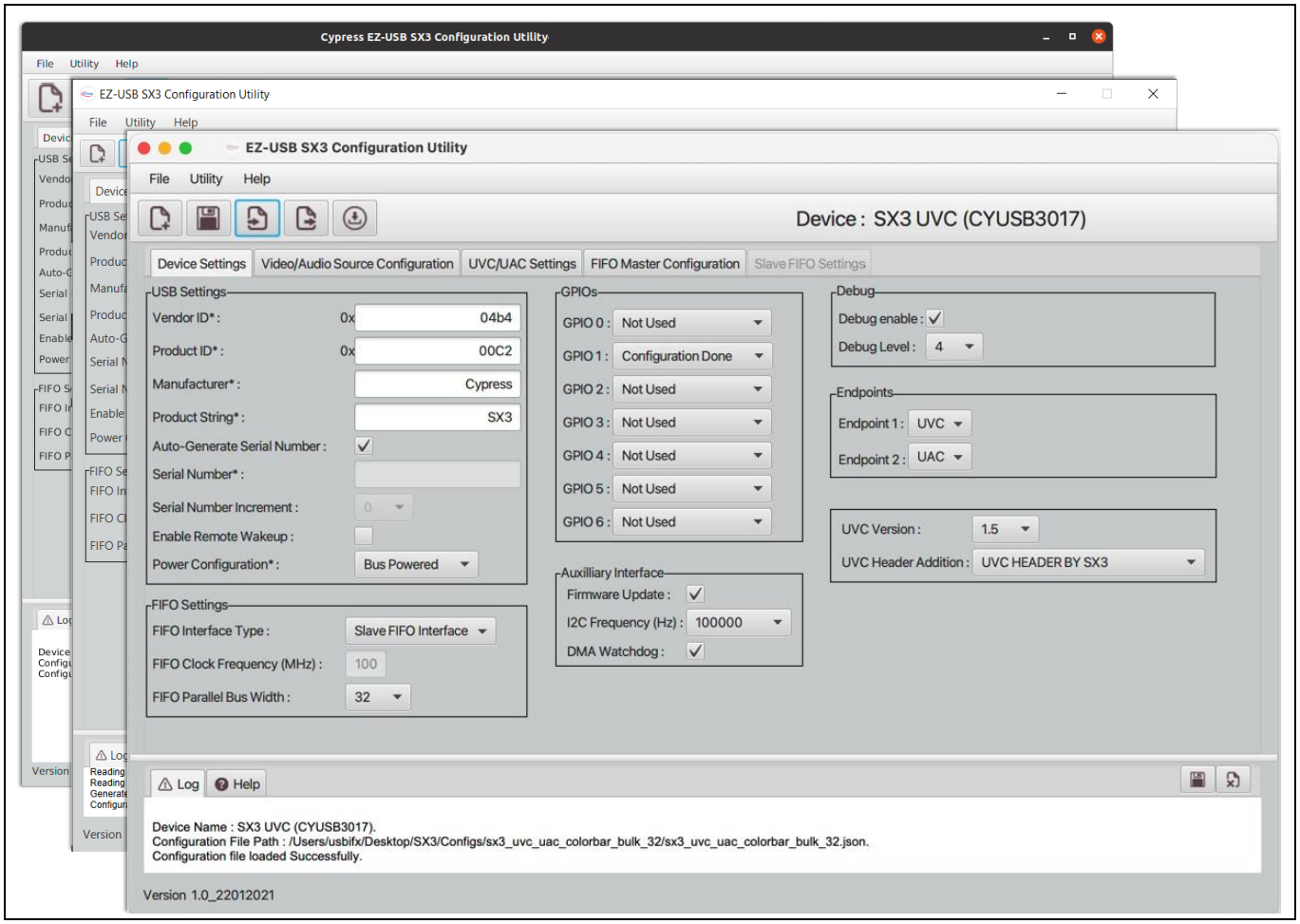



**Figure 1 EZ-USB SX3 configuration utility is supported on Linux, Windows, and macOS**

# 3 Default view of the configuration utility

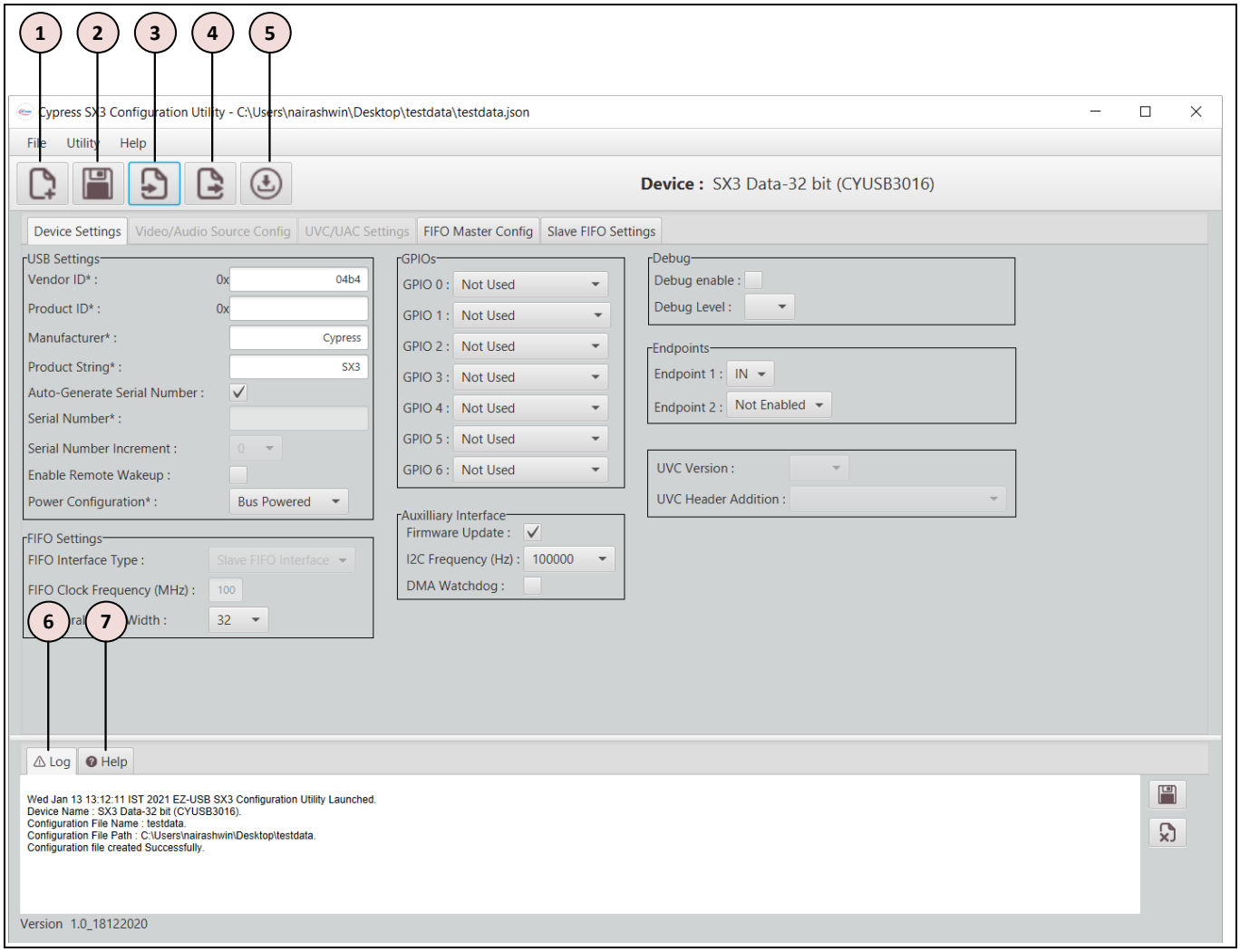



**Figure 2 Buttons and tabs in SX3 configuration utility**

| Button/Tab | Function |
|---|---|
| New Configuration | Create a new configuration file(.json) |
| Save Configuration | Save the current configuration |
| Import Configuration | Load / Import an existing SX3 configuration file (.json or .zip) |
| Export Configuration | Export current configuration in .zip format |
| Program Configuration | Program the current configuration to the connected device |
| Log Tab | Log all the changes in the utility |
| Help Tab | Select this tab to view context-sensitive help content. When a control is selected, the Help tab displays the help text relevant to the selected control. |

# 4 Help tab

The SX3 configuration utility has a Help tab to show context-sensitive help content on mouse click or control selection. Select the Help tab and select on any UI element to bring up context-sensitive help.

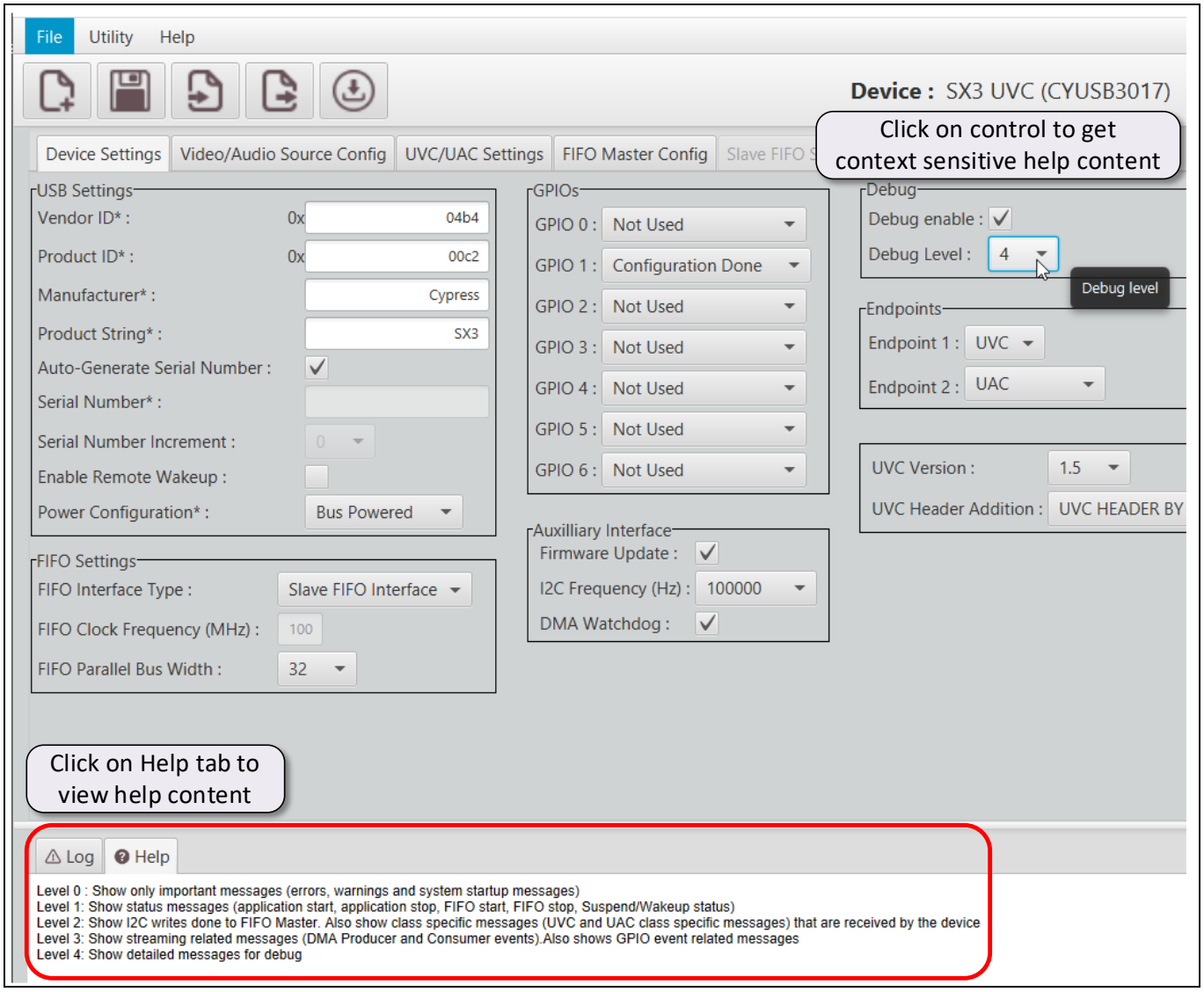



**Figure 3 Help tab**

# 5 Creating a new configuration

1. In Windows, launch the EZ-USB SX3 Configuration Utility from **Start** > **All Programs** > **Cypress** > **EZ-USB SX3 Configuration Utility**.
2. Select **File** > **New Configuration** or click the **New configuration** button.

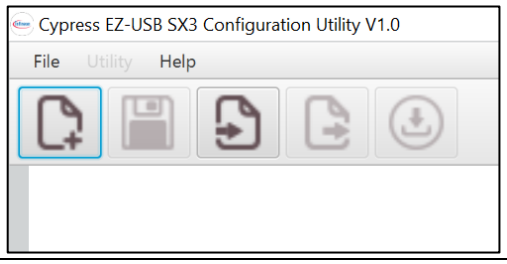



**Figure 4 Click New Configuration button**

3. Select the SX3 device. Depending on the application, choose the appropriate SX3 device as follows:
   - SX3 UVC - For UVC and UAC configurations
   - SX3 Data (16 bit) - For USB Vendor Class data streaming with a 16-bit parallel data bus
   - SX3 Data (32 bit) - For USB Vendor Class data streaming with a 32-bit parallel data bus

*Note: The SX3 device need not be connected to the PC to create an SX3 configuration. You need to connect a device only while programming the created configuration.*

4. As an example, select SX3 UVC (CYUSB3017). Update the configuration file name and path to save the SX3 configuration, and click **OK**.

   A folder with the configuration file name will be created at the chosen path.

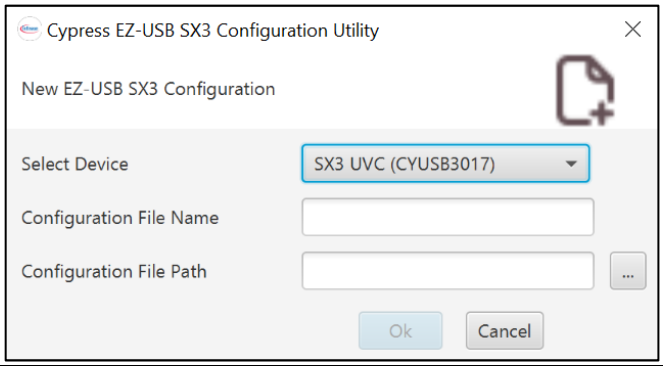



**Figure 5 Device selected as SX3 UVC**

5. Selecting the device as “SX3 UVC (CYUSB3017)” will enable the following tabs in on the tool along with the following default values:

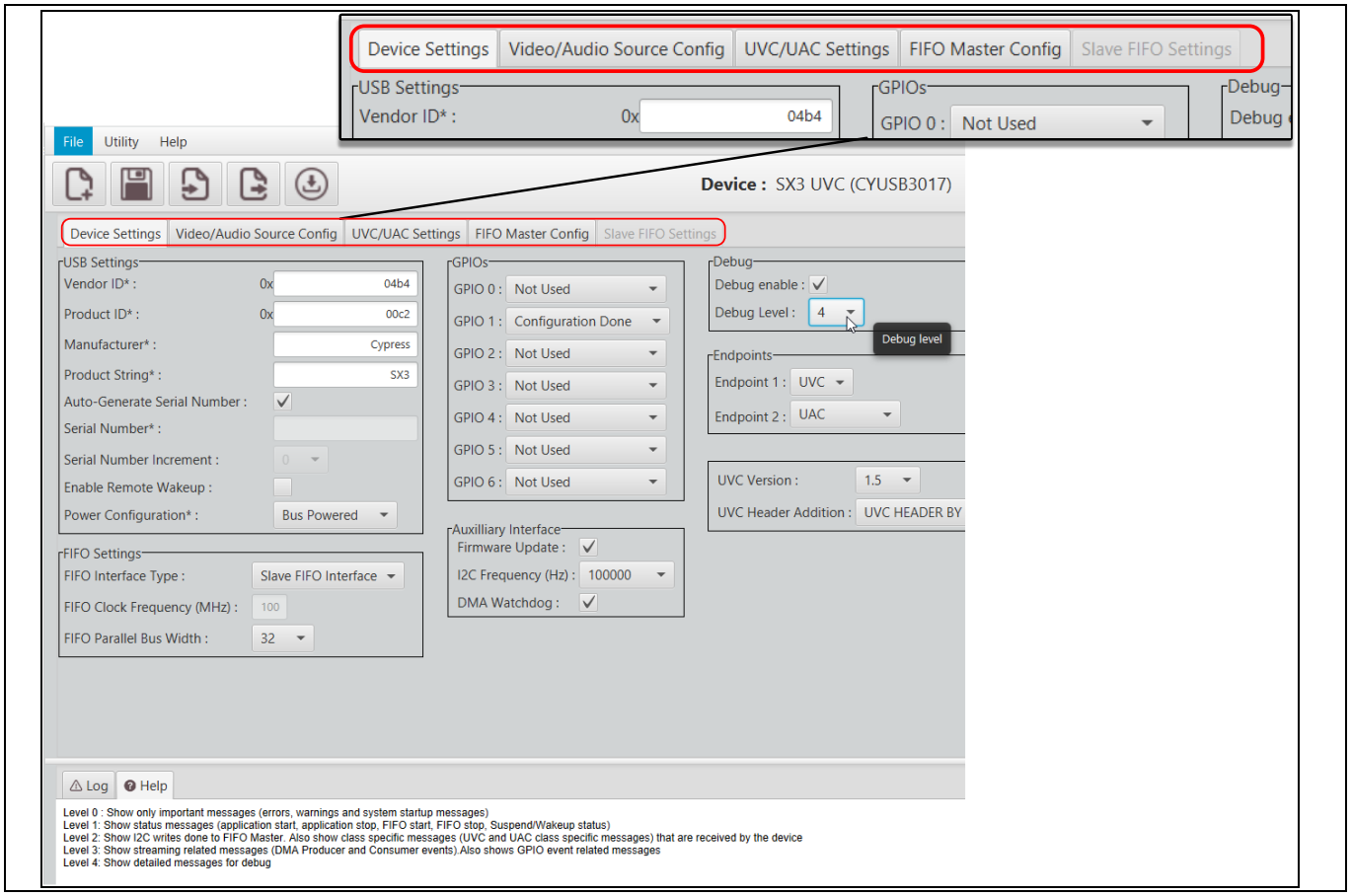



**Figure 6 SX3 UVC (CYUSB3017)- default view**

6. Selecting the device as “SX3 Data-32 bit (CYUSB3015 or CYUSB3016)” will enable the following tabs in the GUI along with the default values:

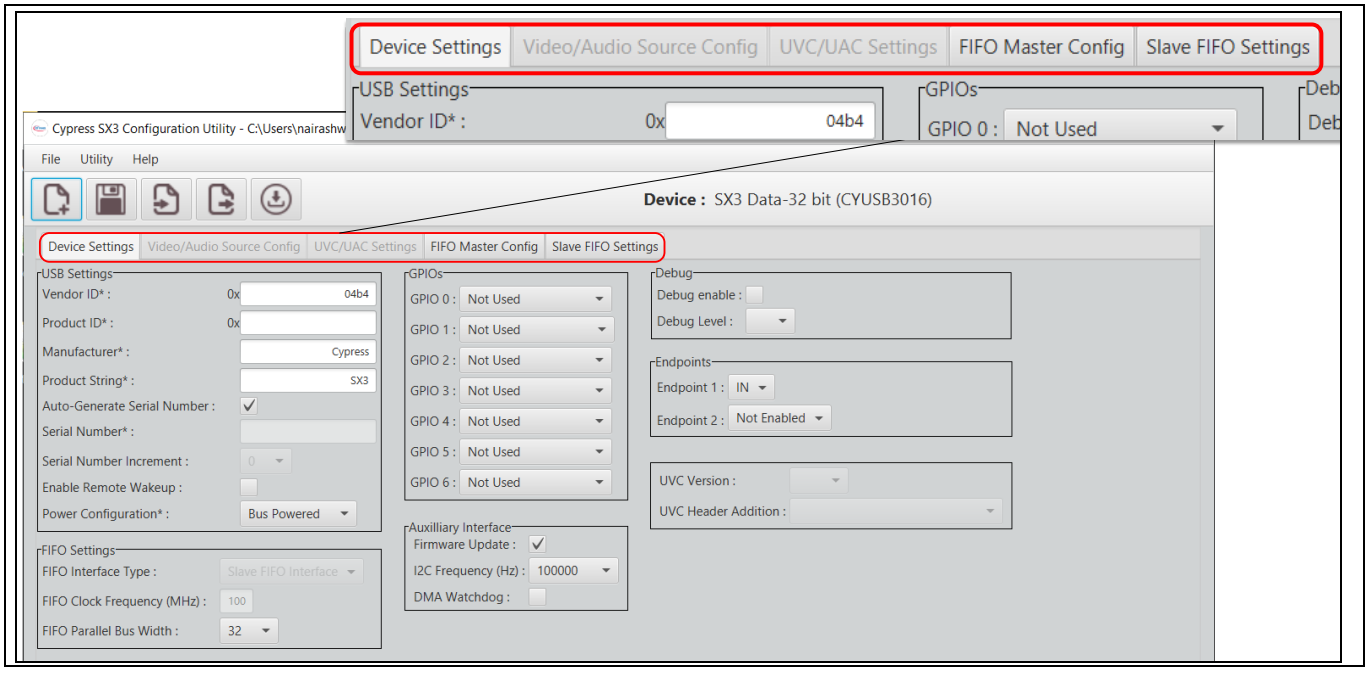



**Figure 7 SX3 Data-32 bit (CYUSB3015 or CYUSB3016) - default view**

# 6 Importing an existing configuration

1. Select **File** > **Import Configuration** or click the **Import Configuration** button on the toolbar.

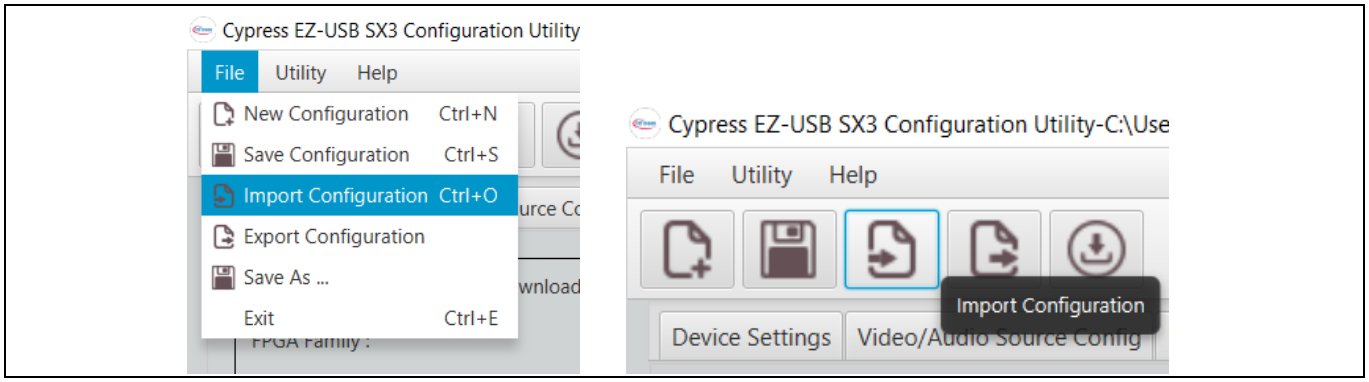



**Figure 8 Import configuration**

You can choose to import either an existing **SX3 Template Project** or **User Project** (created by the user). You can choose from pre-defined configurations included with the installer by selecting an SX3 Template Project or load a configuration created by the user by selecting User Project.

SX3 template projects include all basic configurations supported by SX3 (including sample *FPGA .bit* files).

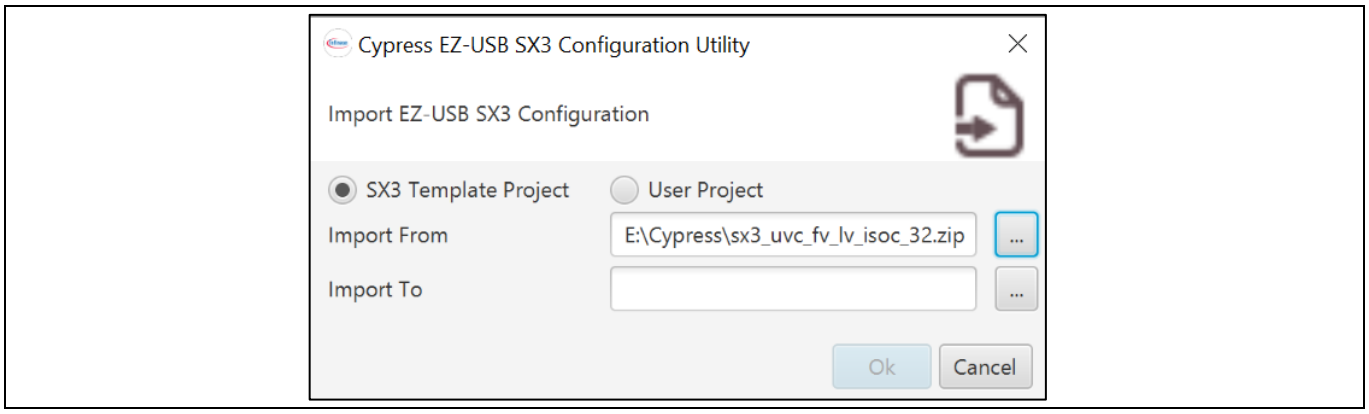



**Figure 9 Importing an SX3 template configuration**

2. Do the following to import an SX3 template project:
   a) Click **Import From** and choose from the existing SX3 template projects in the *templates* directory.

### Importing an existing configuration

| Device | Configuration Name | Details |
|---|---|---|
| CYUSB3017 | sx3_uvc_uac_colorbar_bulk_32 | UVC + UAC composite device streaming colorbar over a UVC endpoint and a 2-kHz sine wave over the UAC interface over a 32-bit parallel slave FIFO interface.<br>Video streaming is over the bulk endpoint in this example. |
| CYUSB3017 | sx3_uvc_uac_colorbar_isoc_32 | UVC + UAC composite device streaming colorbar over a UVC endpoint and a 2-kHz sine wave over the UAC interface over a 32-bit parallel slave FIFO interface.<br>Video streaming is over an isochronous endpoint in this example. |
| CYUSB3017 | sx3_uvc_colorbar_isoc_24 | UVC device streaming colorbar using a 24-bit parallel slave FIFO interface.<br>Video streaming is over an isochronous endpoint in this example. |
| CYUSB3017 | sx3_uvc_fv_lv_isoc_32 | UVC device streaming colorbar using a 32-bit parallel image sensor parallel interface using Frame Valid and Line Valid signals.<br>Video streaming is over an isochronous endpoint in this example. |
| CYUSB3017 | sx3_uac | UAC device streaming a 2-kHz sine wave using a 32-bit parallel slave FIFO interface |
| CYUSB3017 | sx3_uvc_uac_hdmi_generic_4K | UVC + UAC composite device streaming video and audio up to 4K@30fps from an HDMI generic video source over the 32-bit parallel slave FIFO interface |
| CYUSB3017 | sx3_uvc_uac_hdmi_generic_1080p | UVC + UAC composite device streaming video and audio up to 1080p@60fps from an HDMI generic video source over the 32-bit parallel slave FIFO interface |
| CYUSB3017 | sx3_uvc_uac_hdmi_ite_4K | UVC + UAC composite device streaming video and audio up to 4K@30fps from the HDMI IT6801 video source over the 32-bit parallel slave FIFO interface |
| CYUSB3017 | sx3_uvc_uac_hdmi_ite_1080p | UVC + UAC composite device streaming video and audio up to 1080p@60fps from the HDMI IT6801 video source over the 32-bit parallel slave FIFO interface |
| CYUSB3017 | sx3_uvc_crosslink_bulk_vga | UVC device streaming video (VGA@15 fps) from the OV5640 image sensor through the Lattice Crosslink FPGA over the 16-bit image sensor parallel interface |
| CYUSB3017 | sx3_uvc_crosslink_bulk_720 | UVC device streaming video (720p@60 fps) from the OV5640 image sensor through the Lattice Crosslink FPGA over the 16-bit image sensor parallel interface |
| CYUSB3017 | sx3_uvc_crosslink_bulk_1080 | UVC device streaming video (1080p@30 fps) from the OV5640 image sensor through the Lattice Crosslink FPGA over the 16-bit image sensor parallel interface |

**Importing an existing configuration**

| Device | Configuration Name | Details |
|---|---|---|
| CYUSB3016 | sx3_data_in_out_32 | Vendor class device with IN and OUT endpoints. The data received over the OUT endpoint is looped back to the IN endpoint by the Lattice ECP5 FPGA over a 32-bit parallel slave FIFO interface. |
| CYUSB3015 | sx3_data_in_out_16 | Vendor class device with IN and OUT endpoints. The data received over the OUT endpoint is looped back to the IN endpoint by the Lattice ECP5 FPGA over a 16-bit parallel slave FIFO interface. |
| CYUSB3016 | sx3_data_in_32 | Vendor class device with an IN endpoint. The device streams the data received from the 32-bit parallel slave FIFO interface to the IN endpoint. |
| CYUSB3015 | sx3_data_out_16 | Vendor class device with an OUT endpoint. The device streams the data received over the OUT endpoint to the 16-bit parallel slave FIFO interface. |
| CYUSB3015 | sx3_data_in_16 | Vendor class device with an IN endpoint. The device streams the data received from the 16-bit parallel slave FIFO interface to the IN endpoint. |
| CYUSB3016 | sx3_data_out_32 | Vendor class device with an OUT endpoint. The device streams the data received over the OUT endpoint to the 32-bit parallel slave FIFO interface. |
| CYUSB3016 | sx3_data_in_out_xilinx | Demonstrates Xilinx FPGA configuration from SX3 |
| CYUSB3016 | sx3_data_in_out_intel | Demonstrates Intel FPGA Configuration from SX3 |

b) Click **Import To** and navigate to the location to create a working copy of the template project.

## Importing an existing configuration

3. Do the following to import a user project:

a) Click **Import From** and navigate to the user-created JSON file or SX3 Project file (ZIP format) based on the configuration you want to import.

Note that you should change the file type to JSON to view JSON files in the drop-down menu.

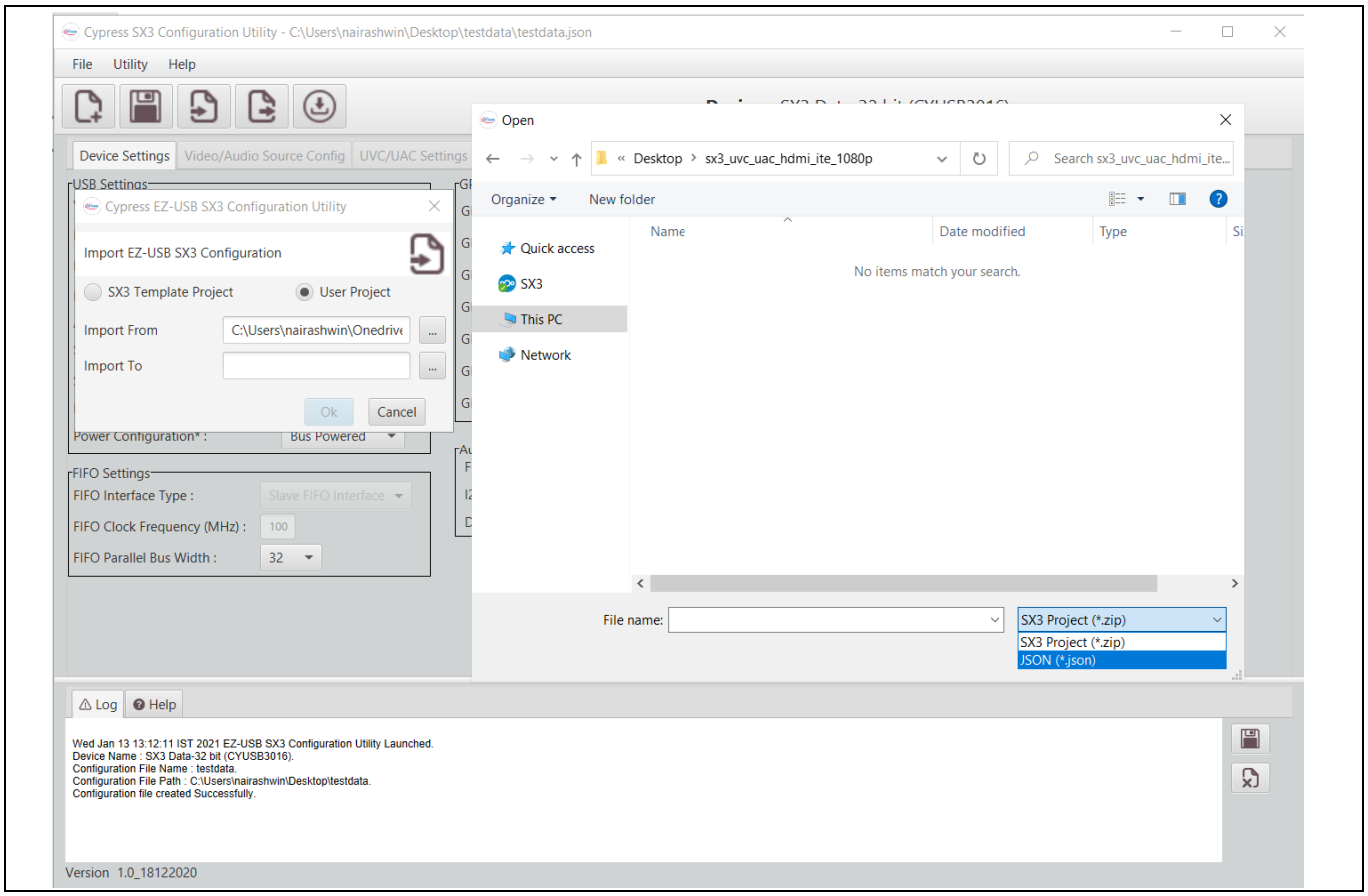



**Figure 10 Navigate to the user project location**

b) Click **Import To** and navigate to the project location to be saved.

# 7 Video/audio source configuration

SX3 can send I2C writes to other I2C slaves (video source or FIFO master) on the bus. I2C writes to the video source (image sensor or HDMI receiver) can be provided from the SX3 configuration utility. See Step 2 in the Example configuration – SX3 UVC UAC HDMI ITE 4K section for more details.

You can enter this data in two ways:

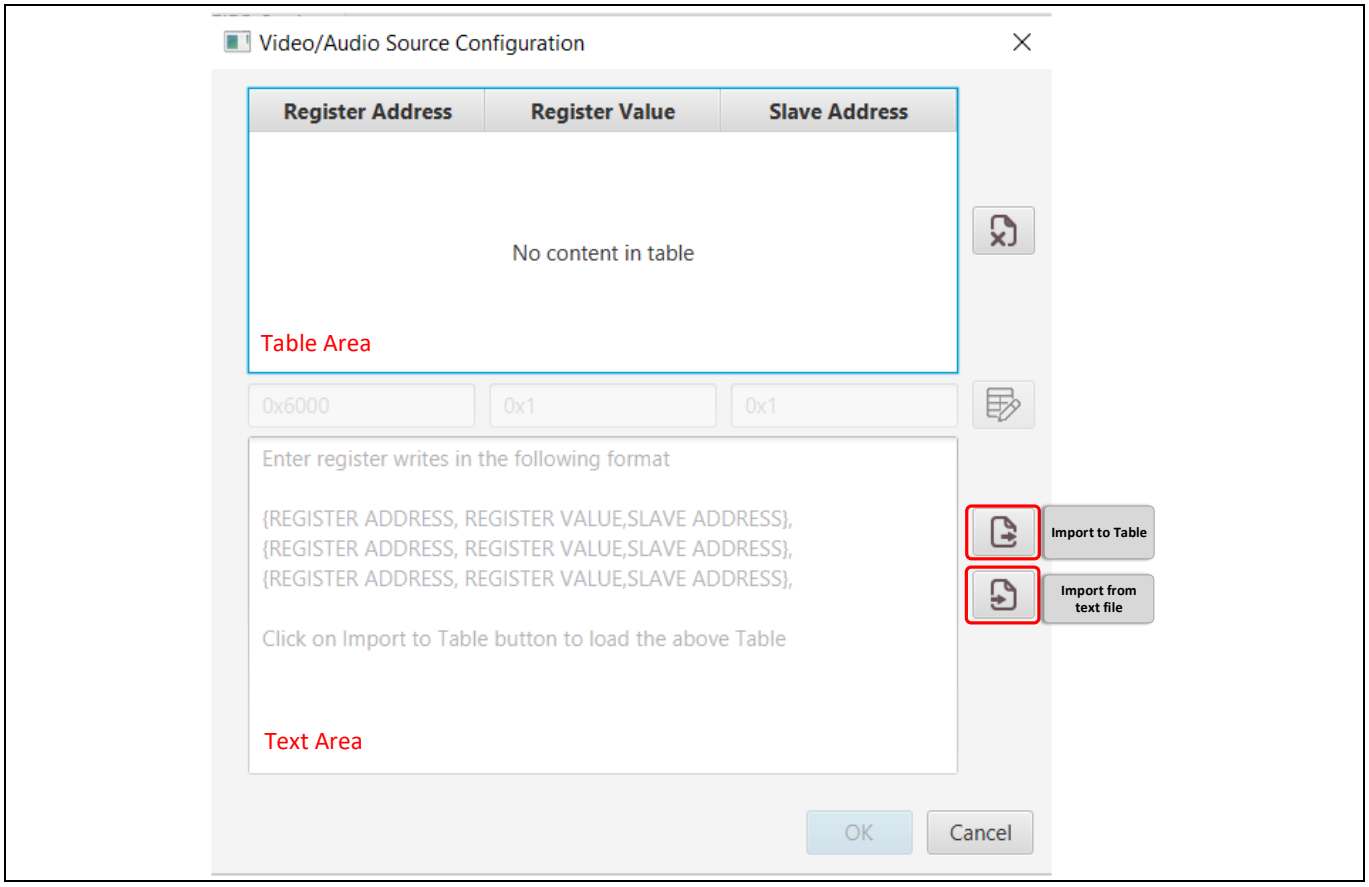



**Figure 11 Add I2C writes for video/audio source**

## 7.1 Import to table from text field

Enter register writes in C array format as {Register Address, Register Value, Slave Address} in the text area and click **Import** to convert the data to table.

Each entry should consist a list of Register Address (maximum 2 bytes), Register Value (maximum 4 bytes) and Slave Address (maximum 2 bytes). Up to 30 entries are allowed in this table.

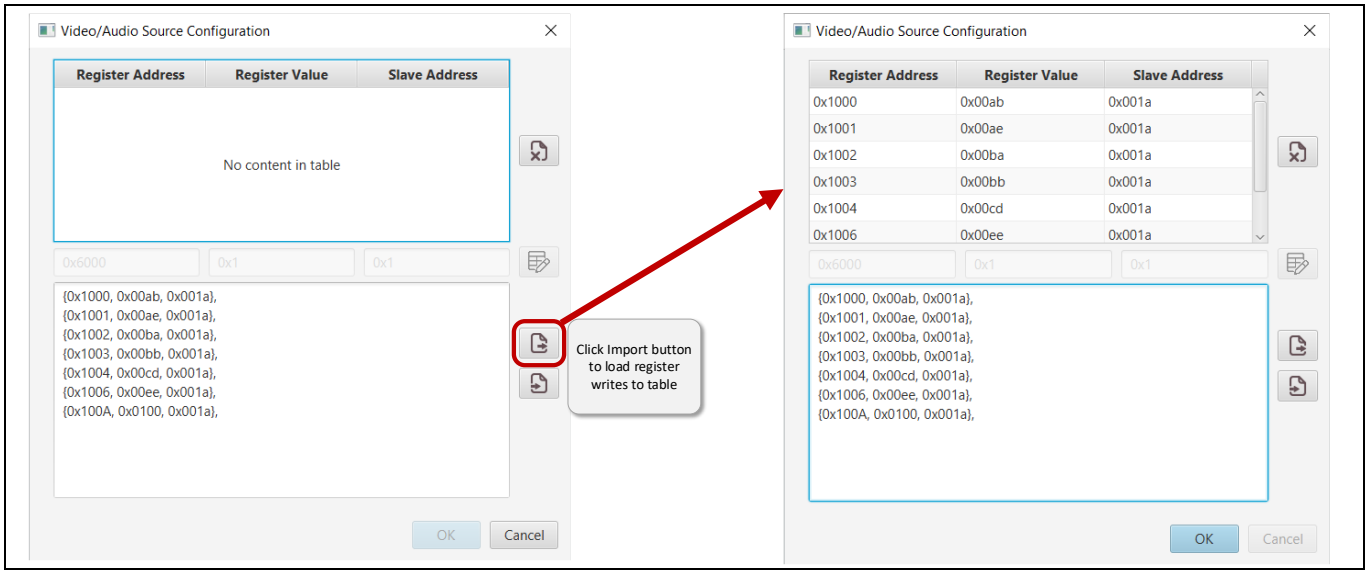



**Figure 12 Add sensor writes from text field**

## 7.2 Import to table from text file

You can import the data from a text file using the **Import from File** option. The sensor writes must be in the form of {Register Address, Register Value, Slave Address} as shown in the following figure.

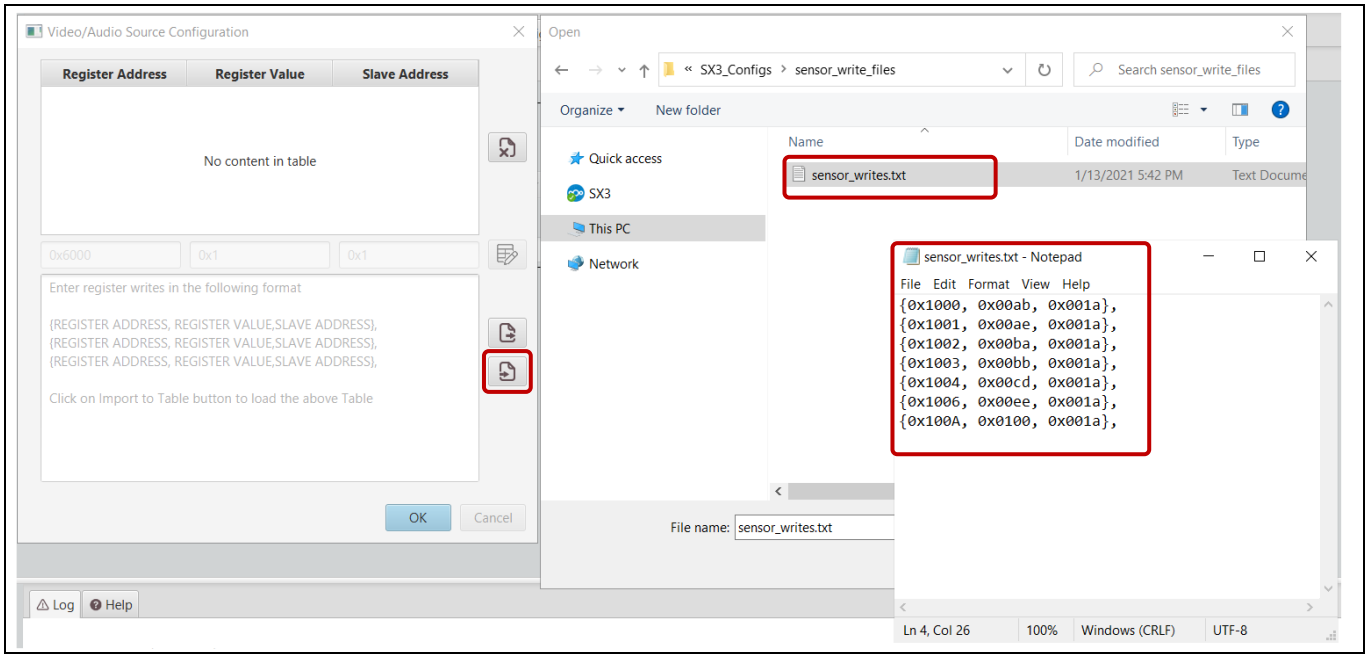



**Figure 13 Add sensor writes from .txt file**

*Note: You can add I2C writes during UVC resolution change or streaming start by using the **Sensor Config** button in the Format And Resolution Table.*

*Note: The maximum size of register address, register data, and slave address is two bytes each.*

### Video/audio source configuration

*Note:* *Slave address 0xDDDD can be used to specify a delay between consecutive I2C writes. The delay value in microseconds can be entered in the **Register Data** field (max 2 bytes). The **Register Address** field is considered as a 'don't care' in this case. For example: to specify a delay of 10us, entry will be {0x00, 0x0A ,0xDDDD}.*

# 8 HDMI generic event handling

Select Video Source Subtype as **HDMI RX - Generic** to interface a generic HDMI receiver with SX3.

**Table 2 Difference between HDMI RX - ITE and HDMI RX - Generic**

| HDMI RX – IT6801 | HDMI RX – Generic |
|---|---|
| Applicable for IT6801 HDMI receiver | Applicable for all other HDMI receivers and custom configurations of IT6801 |
| HDMI event handling is not configurable. | You can configure SX3 to handle HDMI events using the SX3 configuration utility. |
| HDMI initialization and EDID configuration can be done using I2C interface. | HDMI initialization and EDID configuration can be done using the I2C interface. |
| Auto-detection of HDMI resolution change and re-enumeration of the UVC interface. | Must manually switch resolution in the UVC player host application to match the HDMI source resolution. |
| Auto-detect interlaced and progressive resolutions, and configure the FPGA to convert interlaced to progressive. | Interlaced-to-progressive conversion not supported |

As mentioned in **Table 2**, HDMI events that should be handled by SX3 should be added by the user. These can be added in the Event Handling Table as shown in **Figure 14**, which appears in the tool when you select the **Video Source Subtype** as **HDMI RX – Generic**.

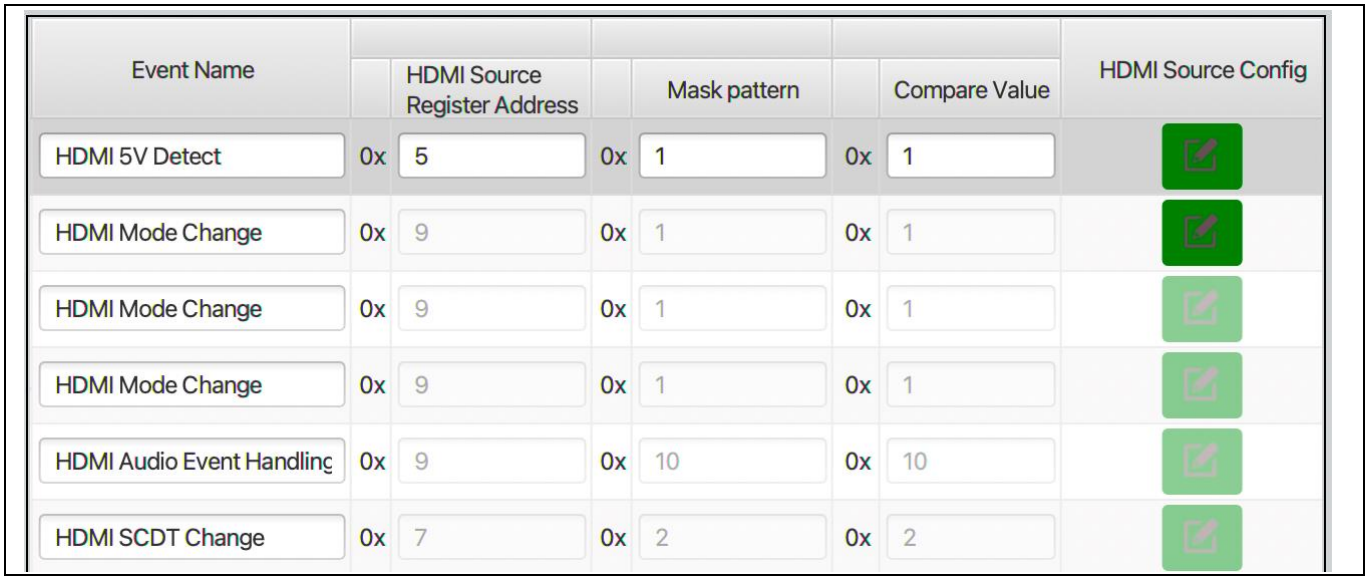



**Figure 14 HDMI Event Handling Table**

SX3 constantly polls the entered HDMI source register addresses over I2C every 100 ms. The read value is masked with the mask pattern and compared with the compare value. In case of a match, the I2C writes listed in the HDMI source config table are executed.

# 9 Programming the device

After the required configurations are done, click the **Program** button.

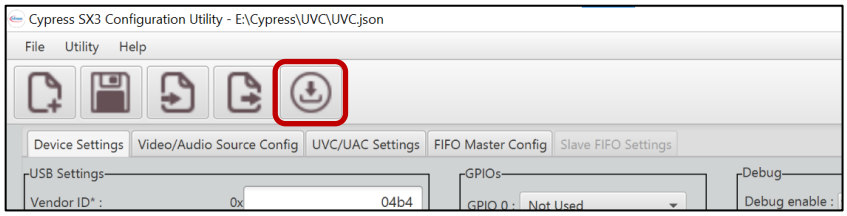



**Figure 15 Program button**

Make sure that you have connected the SX3 device to your system with PMODE options set to USB boot setting. Click the **Download** button to start programming. Click the **Refresh** button to scan for connected devices.

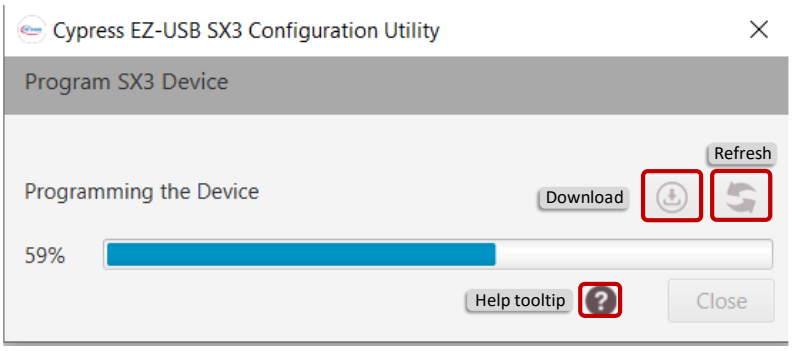



**Figure 16 Actions in the programming window**

After successful completion, the device will be programmed per your requirements. Change the PMODE switch to boot from SPI flash to execute the downloaded configuration.

# 10 Error reporting

Two types of errors are reported in the tool:

## 10.1 Pop-up error messages

a) Entered value exceeds field width

b) Invalid value type entered

c) Invalid value entered

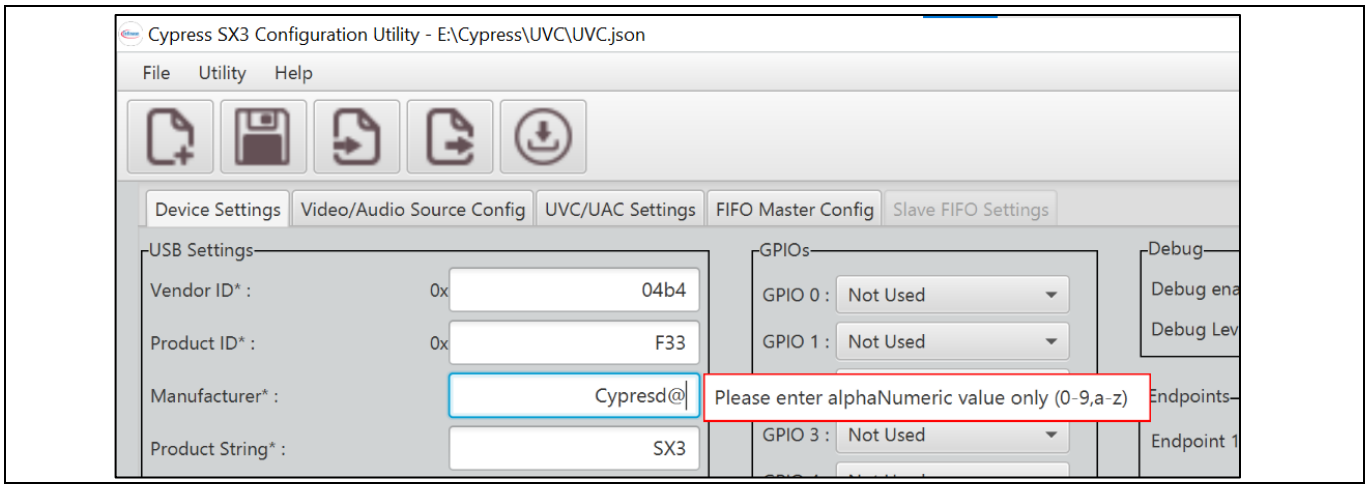



**Figure 17 Errors as popup window**

## 10.2 Errors in log window

Error will be displayed only when you try to save the configuration without entering all mandatory values.

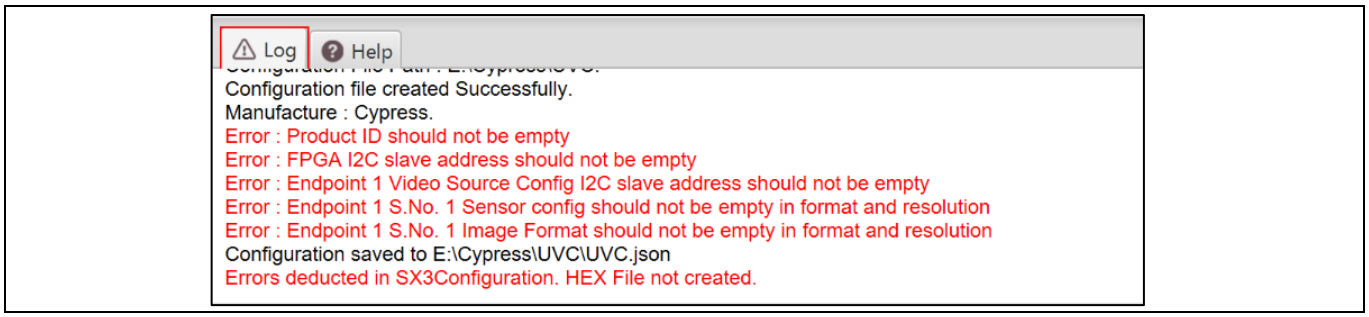



**Figure 18 Errors in log window**

# 11 Log window

## 11.1 Saving the log messages

Click the **Save Log** button to save the log in the configuration path. Note the location in the status bar.

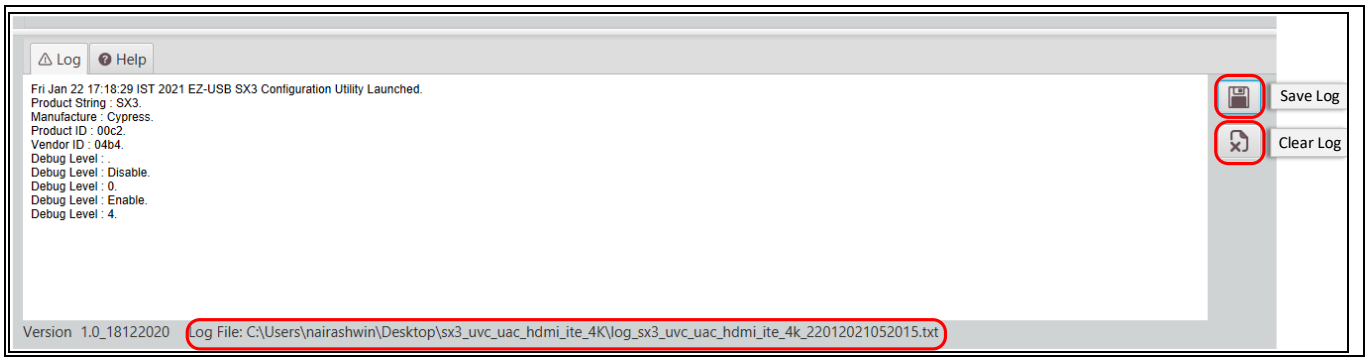



**Figure 19 Save log button**

## 11.2 Clearing the log

Click the **Clear Log** button to clear all logs displayed by the utility.

# 12 Example configuration – SX3 UVC UAC HDMI ITE 4K

The SX3 configuration utility includes 20 sample configurations in the *templates* folder to give a jumpstart to your SX3-based application. This section explains the design flow of the **sx3_uvc_uac_hdmi_ite_4k** template.

The application block diagram is as shown in **Figure 20**.

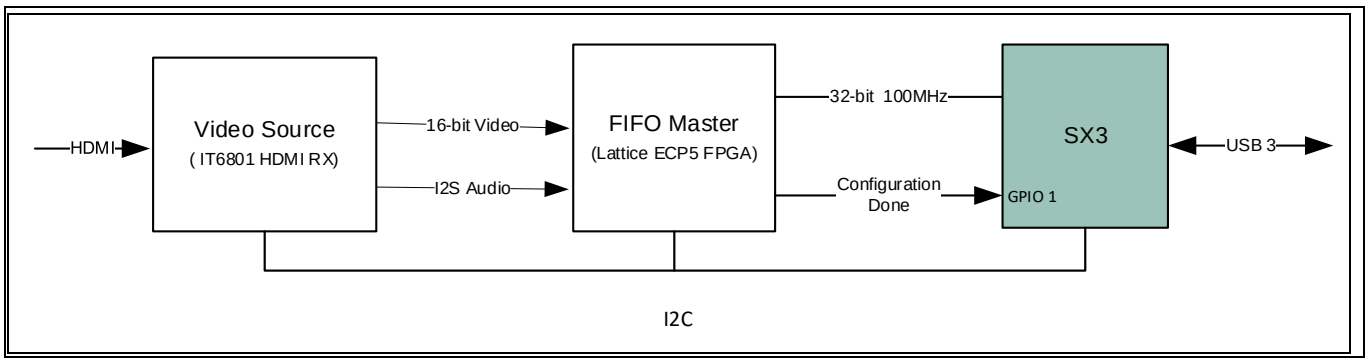



**Figure 20 SX3 UVC UAC HDMI ITE 4K system diagram**

### Step 1: Configure device settings

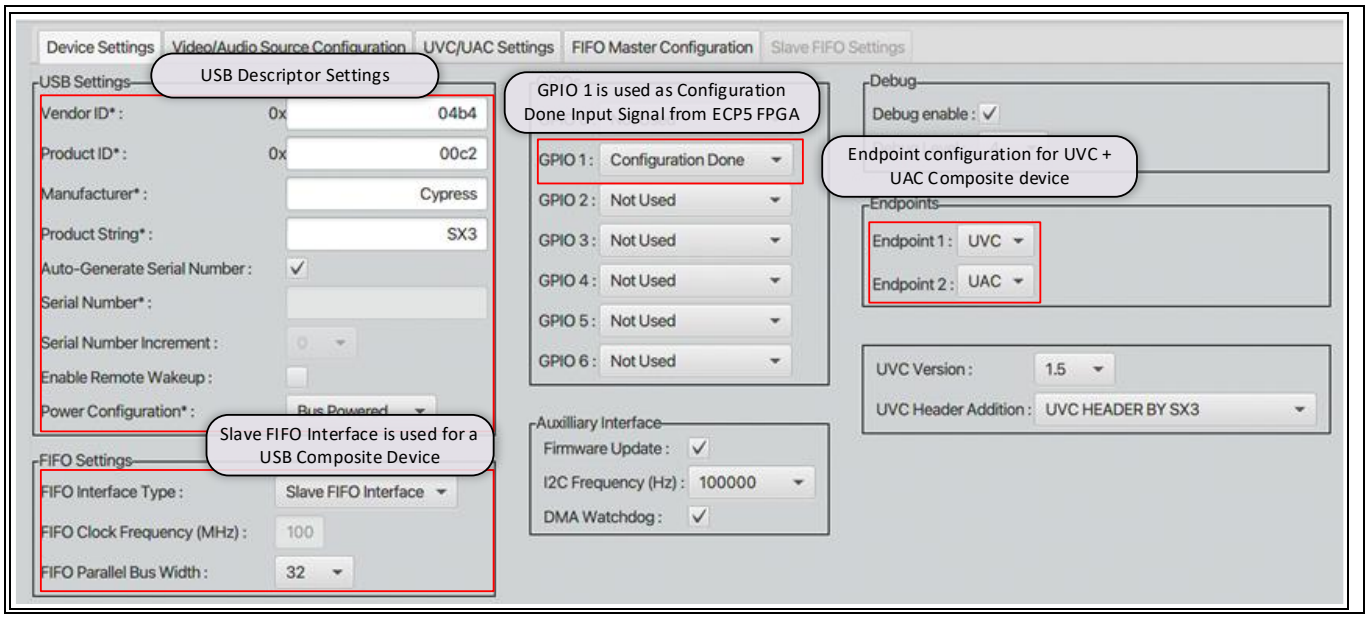



**Figure 21 Device settings for SX3 UVC UAC configuration**

Device Settings screen can be used to configure USB descriptors, FIFO interface settings, GPIO Settings, and endpoint configuration. It can also be used to enable a debug interface over CDC and a firmware update interface over HID. Settings used for SX3_UVC_UAC_HDMI_ITE_4K template are shown in **Figure 21**.

### Example configuration – SX3 UVC UAC HDMI ITE 4K

### Step 2: Configure video source settings

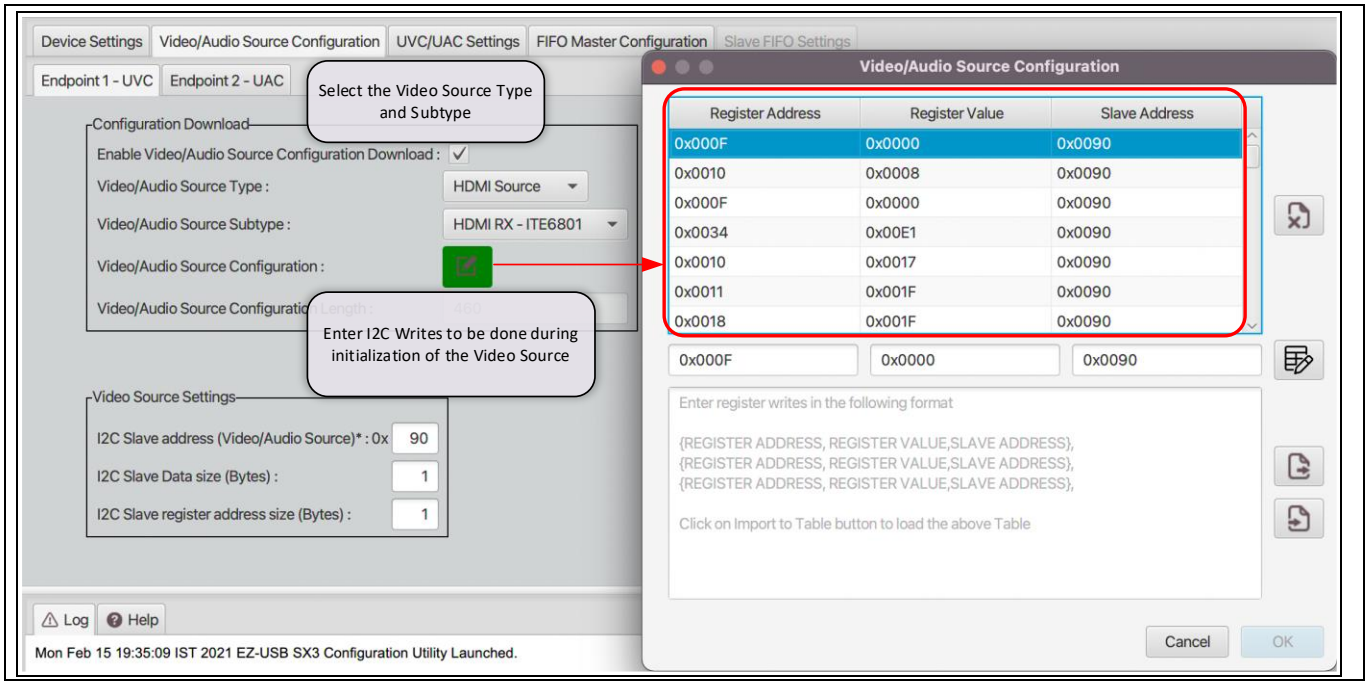



**Figure 22 Video source configuration**

In case of the SX3_UVC_UAC_HDMI_ITE_4K template, the video source is HDMI RX -IT6801. The I2C writes needed to initialize this source are added in the **Video/Audio Source Configuration** tab. In case of HDMI RX, both video and audio are handled by the same source. Therefore, I2C writes needed for audio are also merged with the Endpoint 1 – UVC Video source configuration.

### Example configuration – SX3 UVC UAC HDMI ITE 4K

### Step 3 – Configure UVC settings

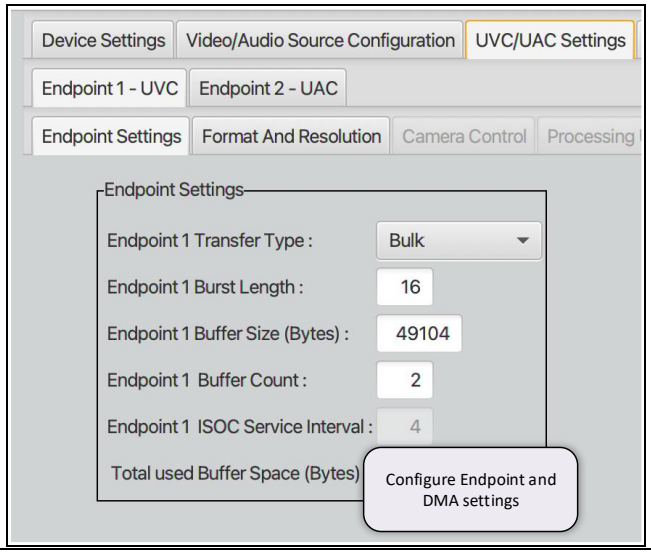



**Figure 23 UVC endpoint and buffer settings**

The SX3_UVC_UAC_HDMI_ITE_4K template uses Bulk transfer type for video data. DMA is configured to have two buffers of 49104 bytes each. Here, the buffer size is selected in such a way that the last buffer received for a UVC frame is less than the full buffer size (typically called as “partial buffer”). This allows SX3 to detect the end of frame and modify the UVC headers appropriately.

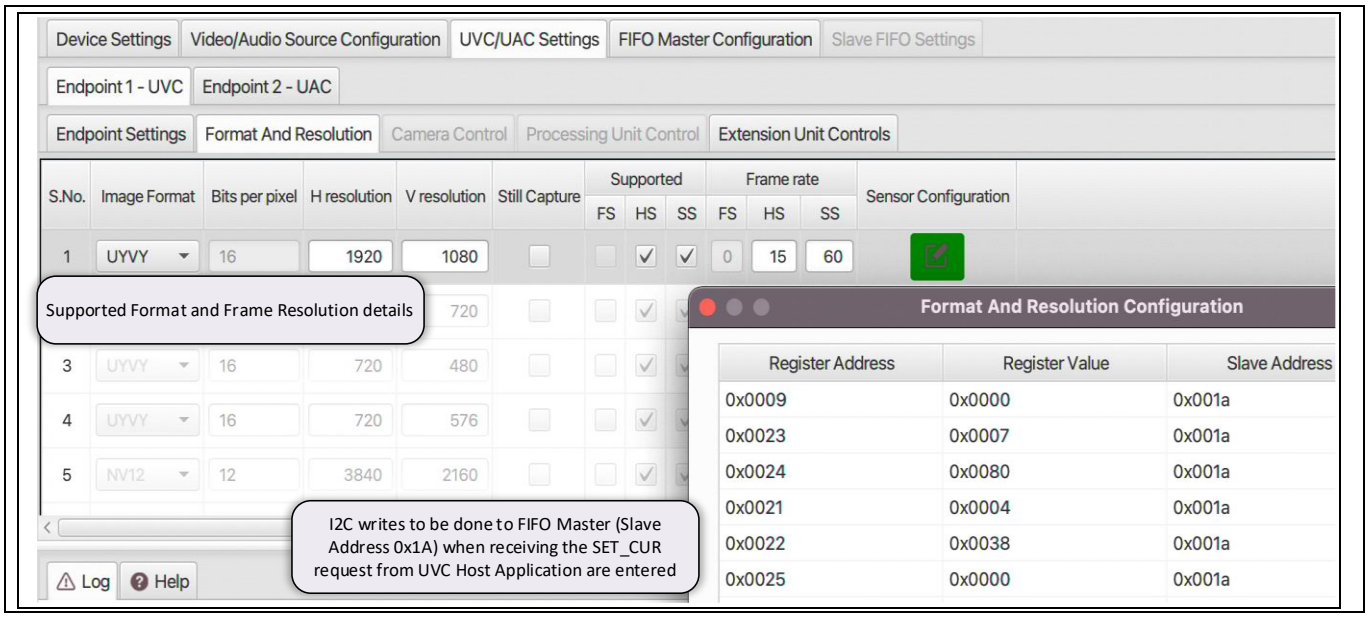



**Figure 24 Add UVC format and frame resolution details**

The SX3_UVC_UAC_HDMI_ITE_4K template supports five standard resolutions. Details for each resolution are added in the **Format and Resolution** tab in the utility. The sensor configuration table provides an interface to add I2C writes to be done to the FIFO master (or any I2C slave) on receiving a **Streaming Start (**or SET_CUR) request from the UVC Host. In this example, I2C writes are done to the FIFO master – I2C slave Interface (slave address 0x1A) to notify the FIFO master of the frame height and width of the selected resolution (see the *Getting Started with EZ-USB SX3* application note for more details on I2C slave interface).

## Example configuration – SX3 UVC UAC HDMI ITE 4K

### Step 4 – Configure UAC settings

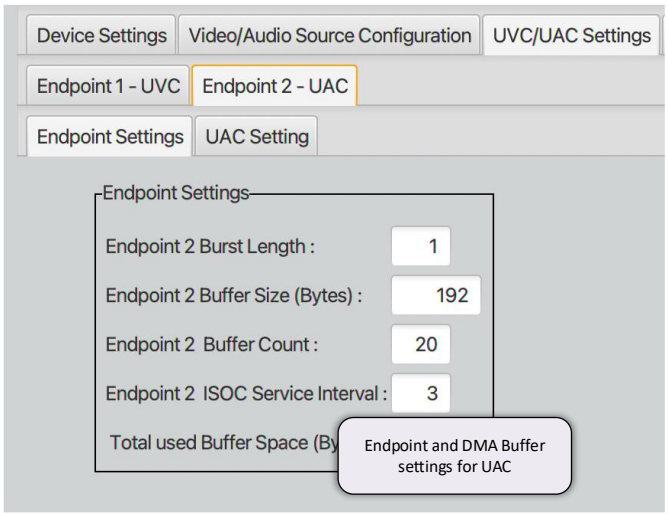



**Figure 25 UAC endpoint and DMA buffer settings**

SX3_UVC_UAC_HDMI_ITE_4K uses isochronous transfer type for the UAC endpoint. The ISOC service interval is set as 3 to set the request interval as 500 microseconds ($2^{(3-1)}$ * 125 us = 500 us).

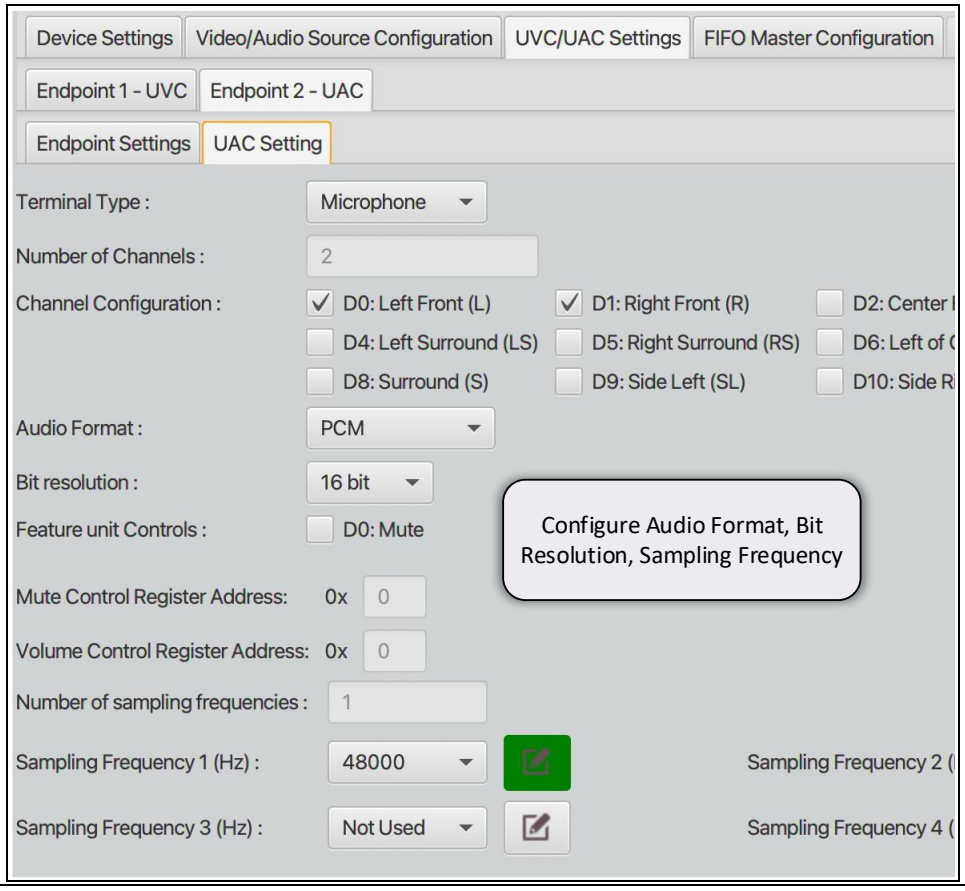



**Figure 26 Audio format, bit resolution and sampling frequency settings**

### Example configuration – SX3 UVC UAC HDMI ITE 4K

### Step 5 – Configure FIFO master settings

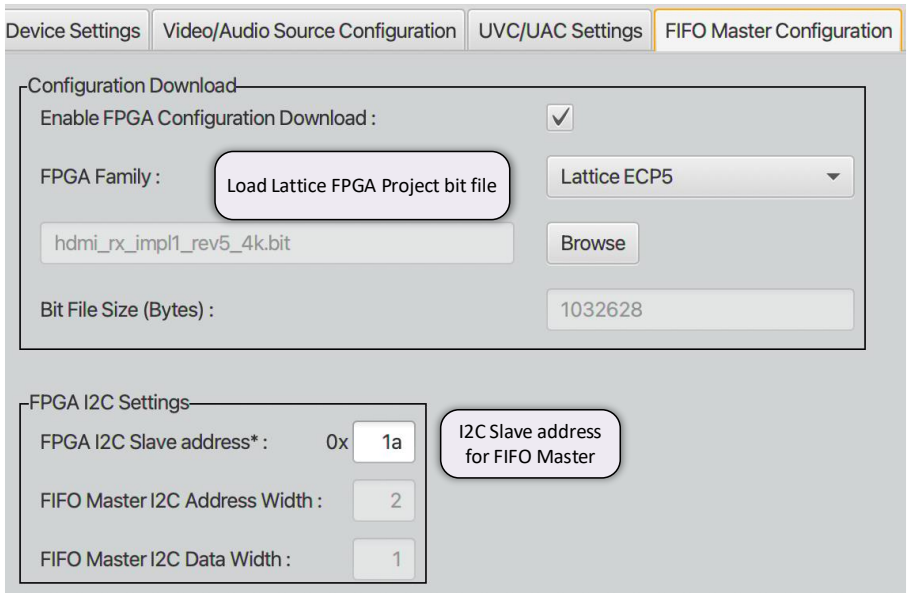



**Figure 27 FIFO master (FPGA) configuration**

The SX3_UVC_UAC_HDMI_ITE_4K template uses the Lattice ECP5 bit FPGA as the FIFO Master. SX3 can configure the FIFO master on bootup. The bit file used for configuration is loaded in the FIFO master configuration screen.
See the *Getting Started with EZ-USB SX3* application note for more details on FPGA configuration and I2C slave interface support in SX3.

# 13 Frequently asked questions / troubleshooting

**What is the default installation path of EZ-USB SX3 configuration utility?**

By default, on Windows, it is installed in *C:/Program Files (x86)/Cypress/EZ-USB SX3 Configuration Utility*. On linux and macOS, the application will be available at the location where the installer package is extracted.

**Device is not getting detected for programming or "No Bootloader Device Detected" error while programming**

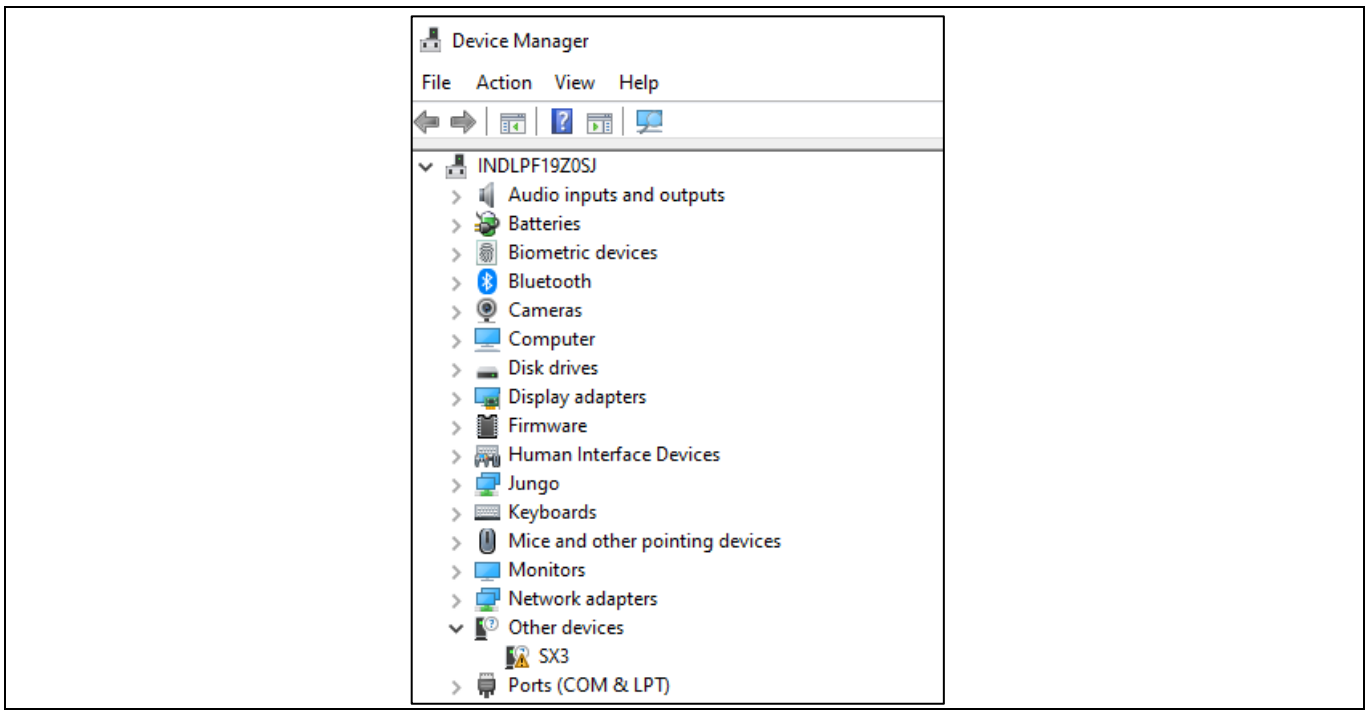
Ensure that PMODE is set to enable USB Boot and cyusb3 drivers are installed (for Windows). See **Appendix** for steps to install the cyusb3 driver.

# 14 Appendix

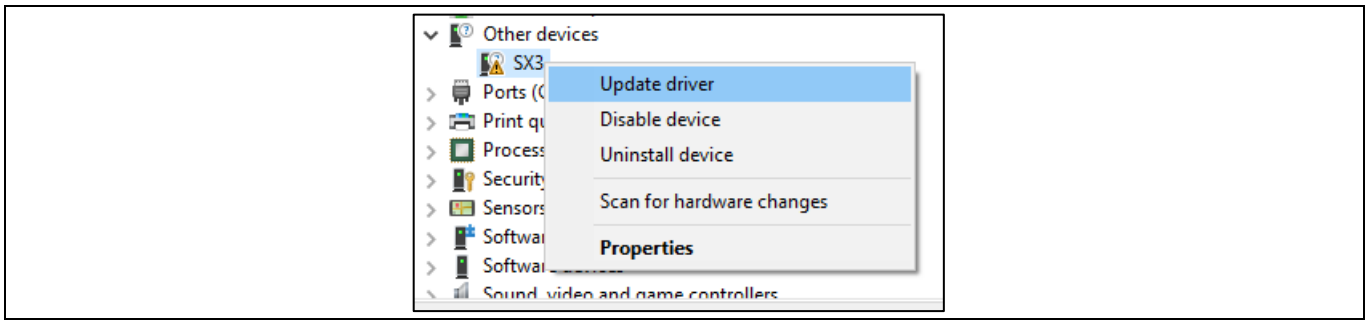
## 14.1 Binding to CYUSB3 driver on Windows for SX3 - data variants

Before programming the device, ensure that appropriate drivers are installed for the attached SX3 Device. Do the following for force-binding SX3 to the CYUSB driver.

1. In Windows Device Manager, under **Other devices**, locate the SX3 device. Note that is appears with a warning symbol.

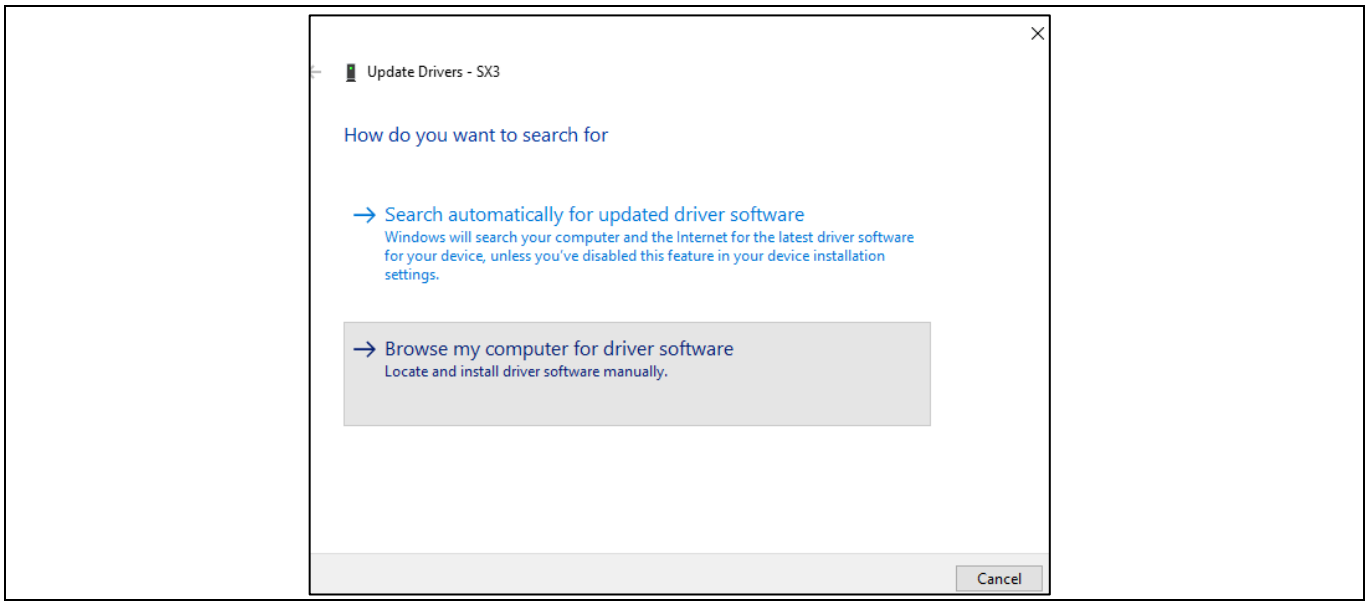



2. Right-click and select **Update Driver**.

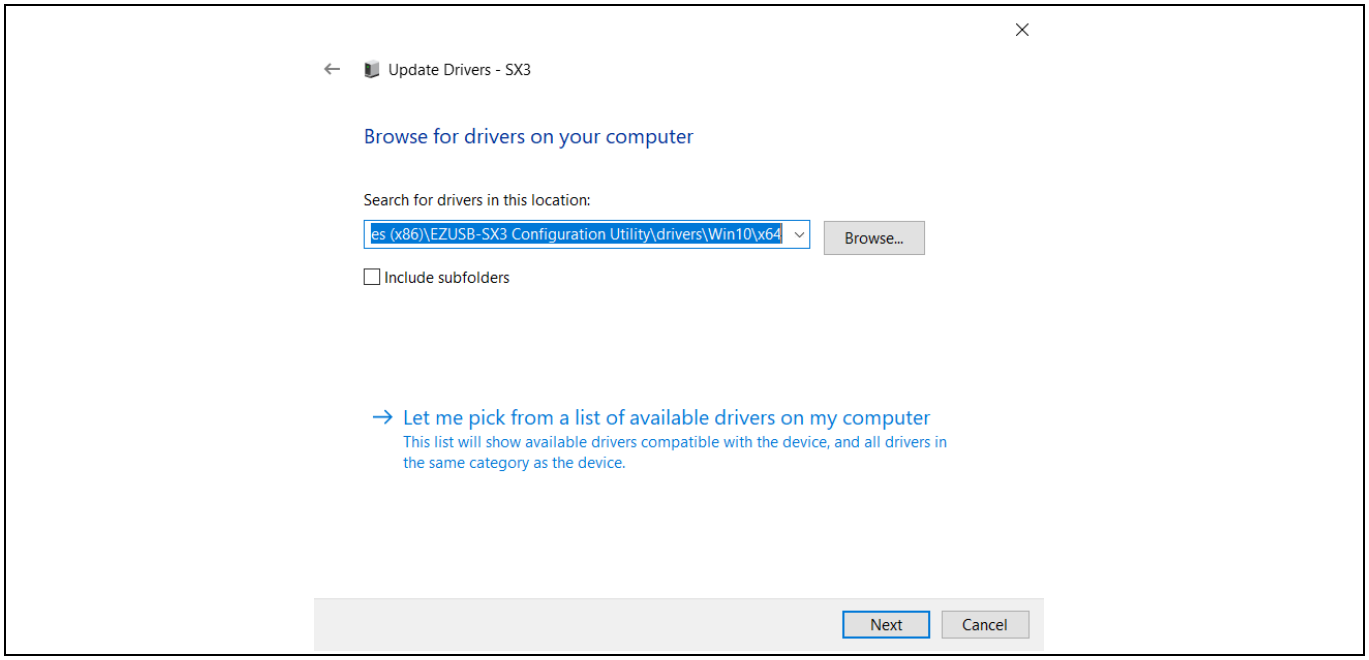


## Appendix

3. Select **Browse my computer for driver software**.

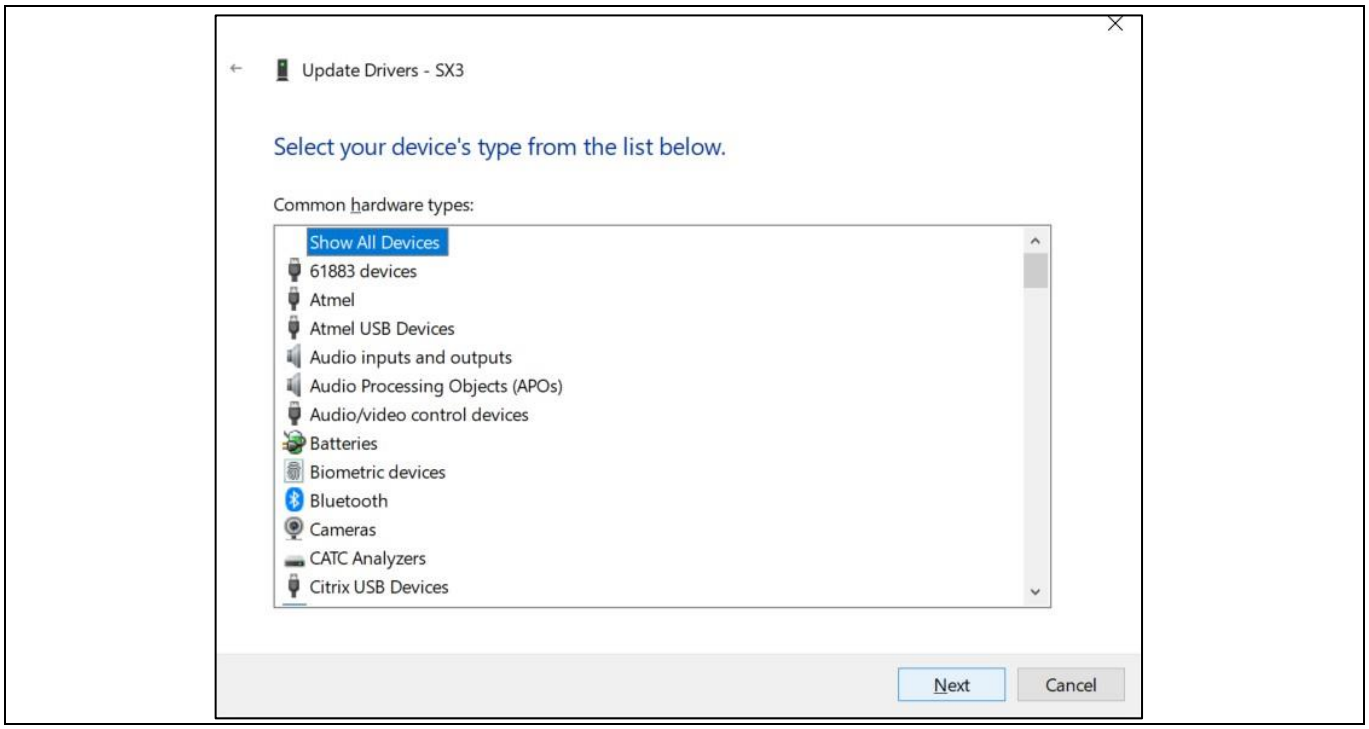



4. Choose **Let me pick from available….**

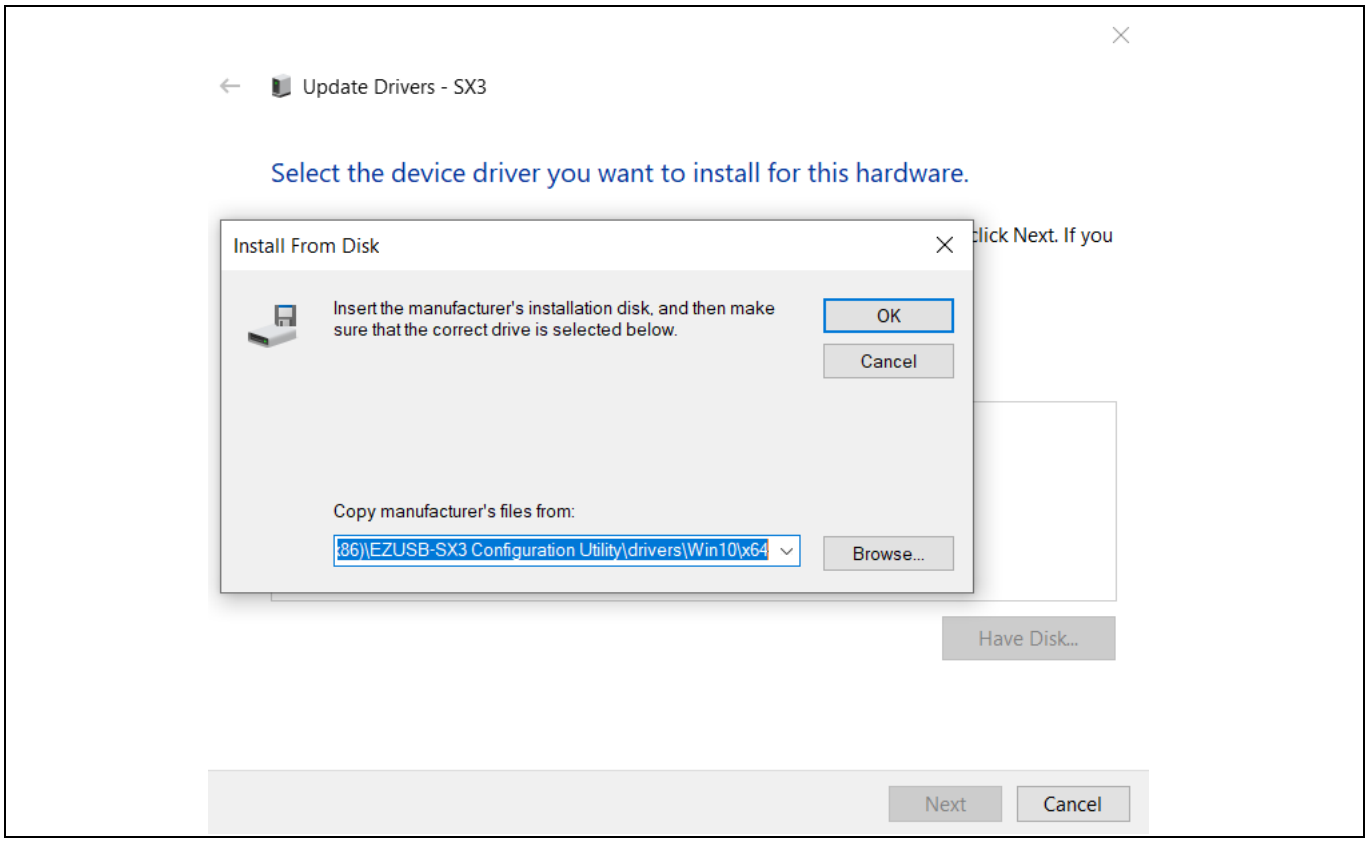


## Appendix

5. Select **Show all Devices** and click **Next**.

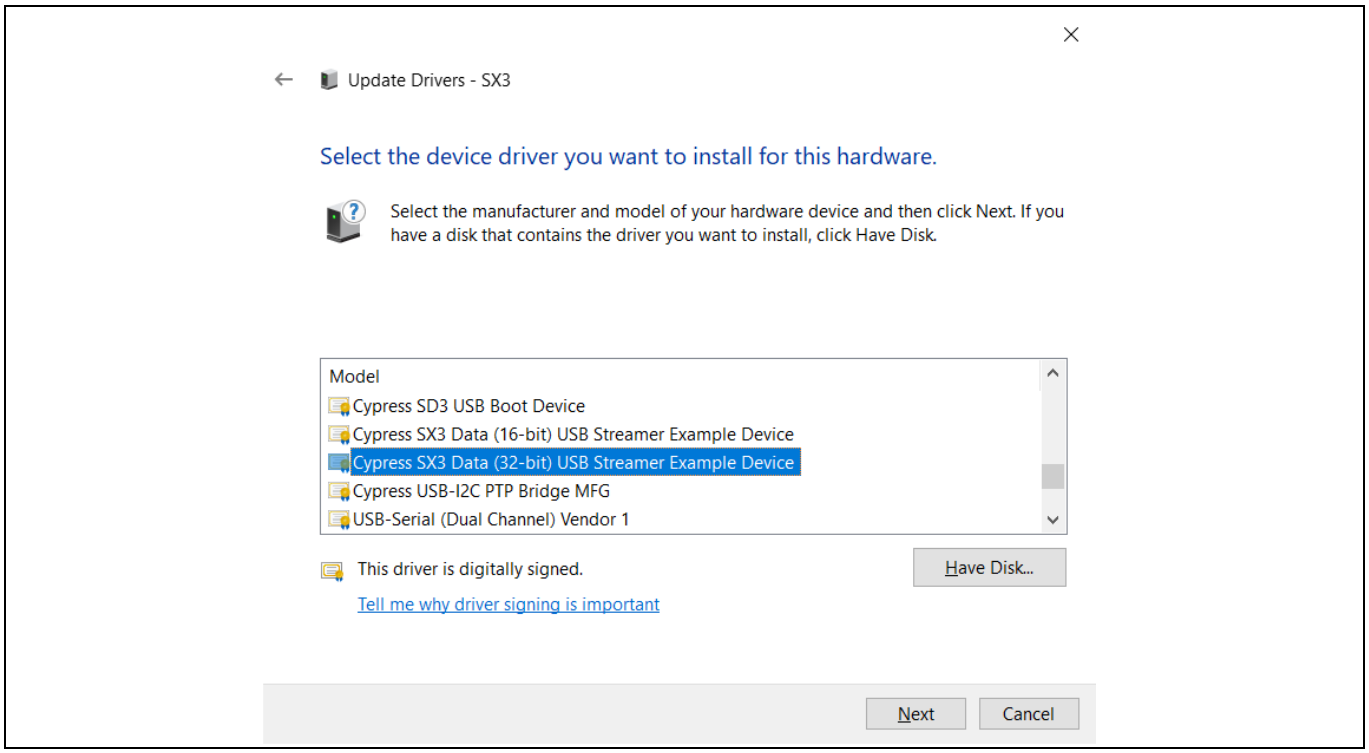



6. Select **Have Disk** and navigate to *C:\Program Files (x86)\EZUSB-SX3 Configuration Utility\drivers\Win10\x64.*



## Appendix

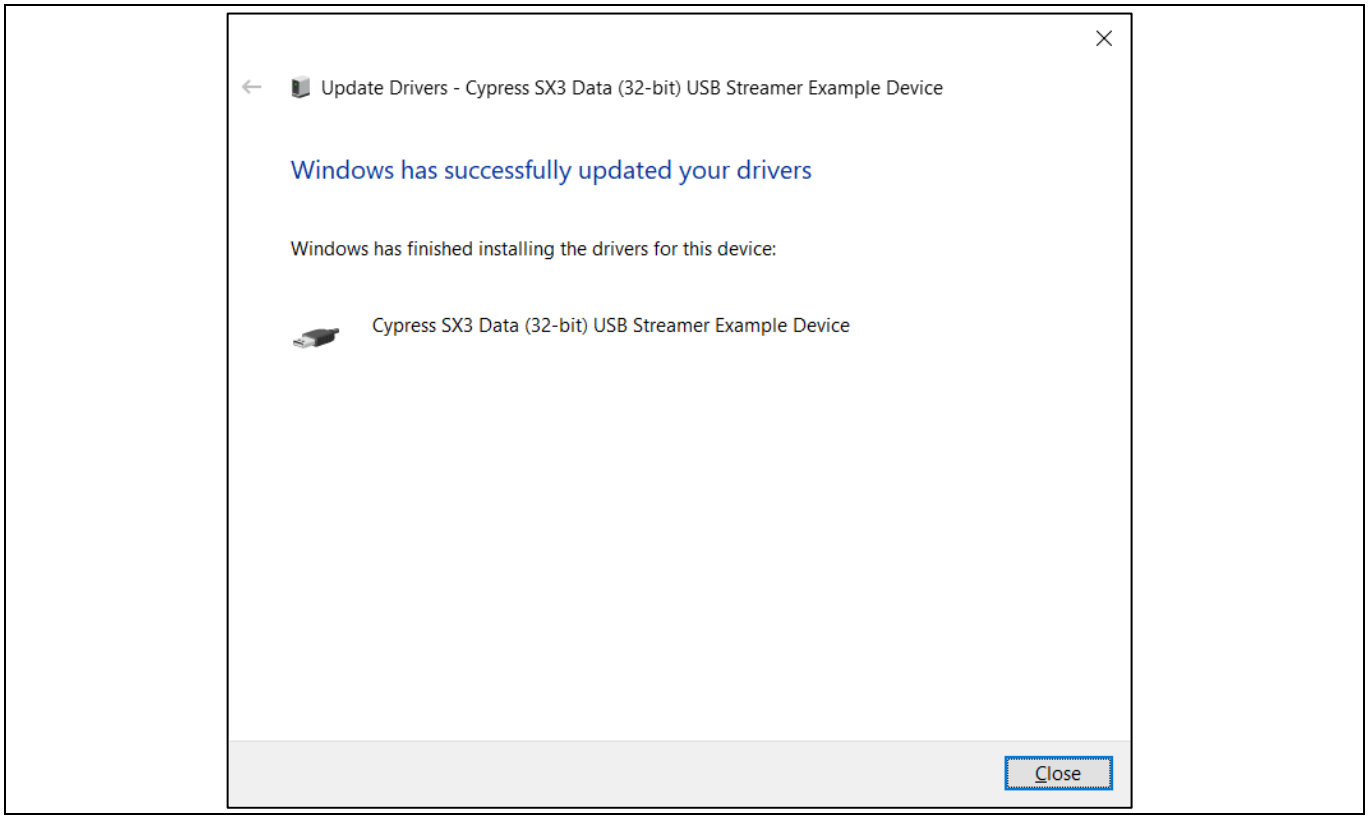
7. Select **Cypress SX3 Data (32-Bit) USB Streamer Example Device** from the list and click **Next**.




8. Press **Yes** for the warning message. The driver will be installed for the attached FX3 device.



## Appendix


Update Drivers - Cypress SX3 Data (32-bit) USB Streamer Example Device
Windows has successfully updated your drivers
Windows has finished installing the drivers for this device:
Cypress SX3 Data (32-bit) USB Streamer Example Device
Close

# Revision history

| Date | Version | Description |
|---|---|---|
| 2021-02-19 | ** | Initial release |

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

**Edition 2021-02-19**
**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2021 Infineon Technologies AG. All Rights Reserved.**

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

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

**Document reference**
**002-32600 Rev. ****

**IMPORTANT NOTICE**
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

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.