DFU loader for XMOS USB AUDIO devices

Document Number: XM000524A

Publication Date: 2014/6/11

Table of Contents

1 Overview 3

2 Creating factory and upgrade images with XFLASH 4
- 2.1 Installing the factory image to the device . . . 4
- 2.2 Creating the upgrade image . . . 5
- 2.3 Related documents . . . 5

3 Using the DFU loader - Windows (via the Thesycon driver) 6
- 3.1 Set up the image loader . . . 6
- 3.2 Download new firmware . . . 6
- 3.3 Uploading existing firmware from the device . . . 7
- 3.4 Reverting firmware to factory image . . . 7
- 3.5 Related documents . . . 7

4 Using the DFU loader - OS X (via the XMOS DFU loader) 8
- 4.1 Set up the image loader . . . 8
- 4.2 Download new firmware . . . 8
- 4.3 Uploading existing firmware from the device . . . 8
- 4.4 Reverting firmware to factory image . . . 9

5 Building the XMOS DFU loader - OS X 10

1 Overview

The DFU loader is a flash device firmware upgrade mechanism. To work correctly your development board must contain the latest DFU enabled firmware.

The firmware upgrade for XMOS USB devices implementation uses the USB standard DFU device class mechanism and is based on the following specification:

- http://www.usb.org/developers/devclass_docs/DFU_1.1.pdf

Supported functionality:

- Download of new firmware to device
- Upload of existing firmware from device
- Revert device back to factory image
- Automatic reboot of device on firmware upgrade

You must use XMOS Development Tools version 10.4.1 (or later).

The DFU device on Windows requires the Theyscon USB Audio 2.0 Windows driver version 1.13.3 or later.

2 Creating factory and upgrade images with XFLASH

IN THIS CHAPTER

- Installing the factory image to the device
- Creating the upgrade image
- Related documents

2.1 Installing the factory image to the device

The DFU device interface is enabled by default in the XMOS USB Audio framework (see `devicedefines.h`), and explicitly enabled in each reference design by the following line included in the `customdefines.h` file of each application:

```
#define DFU                 (1)
```

Use the XMOS Development Tools to run the command:

```
xflash --boot-partition-size 0x30000 --factory usb_audio.xe
```

Where the size passed using the `--boot-partition-size n` argument specifies in bytes the minimum size required to store the boot loader, factory image and any upgrade images.

The following can be used as a guide to help calculate the required boot partition size for a design:

- boot partition size = loader size + maximum size of factory image + maximum size of upgrade images + padding to sector boundaries

Where:

- loader size = 0x1000 bytes, or 0x4000 bytes if secure boot functionality is used (for tools version 13.1.0 and older)
- maximum size of factory image = number of xCORE tiles * xCORE SRAM size
- maximum size of upgrade images = (number of xCORE tiles * xCORE SRAM size) * number of images to be held in flash concurrently

The above example sets the boot partition to a size appropriate for designs based on a single xCORE tile, where a single upgrade image is required in flash at any one time.

This programs the factory default firmware image into the flash device. The device will now support the DFU mechanism, and can use it to safely receive firmware updates, as well as revert to the factory firmware image when required, such as in the event of a failed upgrade attempt.

2.2 Creating the upgrade image

To use the firmware upgrade mechanism you need to build a firmware upgrade image:

1. Edit the `customdefines.h` file of the application for the hardware in use, and change the Device Version Number by defining `BCD_DEVICE`.

2. Rebuild the application.

To generate the firmware upgrade image run the following command:

```
xflash --factory-version 13 --upgrade 1 usb_audio.xe -o new_firmware.bin
```

Where the tools version passed using the `--factory-version version` argument specifies the version of the tools used to create the factory image. This should be passed as `12` for images created using tools versions 10, 11 and 12.

The `--upgrade id xe-file [size]` argument specifies xe-file as an upgrade image with version `id`. Each version number must be a unique number greater than 0.

You should now have the file `new_firmware.bin` which contains the firmware with the newly specified Device Version Number.

2.3 Related documents

For further details of the DFU features included in the XMOS USB Audio framework and their configuration please see XM-005512-PC and XM-005512-PC.

For further details on the use of XFLASH to create factory and upgrade firmware images please see the XFLASH Command-Line Manual section of the xTIMEcomposer User Guide[1].

[1] https://www.xmos.com/published/xtimecomposer-user-guide

3 Using the DFU loader - Windows (via the Thesycon driver)

IN THIS CHAPTER

- Set up the image loader
- Download new firmware
- Uploading existing firmware from the device
- Reverting firmware to factory image
- Related documents

Thesycon provide both GUI and CLI DFU tools, TUSBAudioDfu.exe and dfucons.exe respectively. The use of the GUI tool is not covered by this document.

The correct installation of the Thesycon driver and DFU tools exceeds the scope of this document.

3.1 Set up the image loader

Run the DFU console tool (`dfucons.exe`) from the Thesycon install folder, in a Command Prompt by navigating to:

```
C:\Program Files\Thesycon\TUSBAudio_Driver\
```

To check the device has been detected, run the following command in the DFU console:

```
dfucons info
```

The console shows the DFU devices that have been detected.

3.2 Download new firmware

To program the new firmware run the command:

```
dfucons download new_firmware.bin
```

Note that once this is done the device restarts. The original factory default application is still present but the device is now running the upgraded application firmware.

You can check the device has been updated by running the command:

```
dfucons info
```

This will display the device revision.

3.3 Uploading existing firmware from the device

You can retrieve a firmware image from the device, providing an upgrade image is present. Run the command:

```
dfucons upload currentfirmware.bin
```

The file `currentfirmware.bin` contains the latest upgrade image. This file is an exact copy of the data from the flash and can be downloaded to the device again to test.

3.4 Reverting firmware to factory image

To revert the device back to its factory (i.e XFLASH) installed state from the new firmware, run the command:

```
dfucons revertfactory
```

The device will now be running, and only contain the factory firmware, which can be seen by checking the device version once more.

3.5 Related documents

For further details on the use of the Thesycon DFU tools please see Thesycon USB Audio 2.0 Driver for Windows User Manual[2].

[2] https://www.xmos.com/published/usb-audio-class-20-evaluation-driver-windows

4 Using the DFU loader - OS X (via the XMOS DFU loader)

IN THIS CHAPTER

- Set up the image loader
- Download new firmware
- Uploading existing firmware from the device
- Reverting firmware to factory image

The XMOS DFU loader is provided as source as part of the XMOS USB Audio software framework, see §5.

4.1 Set up the image loader

1. Open a terminal
2. Change directory to where the loader has been built
3. Run the command:

```
source setup.sh
```

4.2 Download new firmware

To program the new firmware run the command:

```
./xmosdfu --download new_firmware.bin
```

Note that once this is done the device restarts. The original factory default application is still present but the device is now running the upgraded application firmware.

4.3 Uploading existing firmware from the device

You can retrieve a firmware image from the device, providing an upgrade image is present.

Run the command:

```
./xmosdfu --upload currentfirmware.bin
```

The file `currentfirmware.bin` contains the latest upgrade image. This file is an exact copy of the data from the flash and can be downloaded to the device again to test.

4.4 Reverting firmware to factory image

To revert the device back to its factory (i.e XFLASH) installed state from the new firmware, run the command:

```
./xmosdfu --revertfactory
```

The device will now be running, and only contain the factory firmware.

5 Building the XMOS DFU loader - OS X

The XMOS DFU loader is provided as source as part of the USB Audio framework, located in sc_usb_audio/module_dfu/host/xmos_dfu_osx.

The loader is compiled using libusb, the code for the loader is contained in the file `xmosdfu.cpp`

To build the loader a Makefile is provided, which can be run as follows:

```
make -f Makefile.OSX all
```

This Makefile contains the following:

```
all:
        g++ -g -o xmosdfu xmosdfu.cpp -I. -IOSX libusb-1.0.0.dylib
          ↪  -m32
```