UM0391
User manual

DfuSe File Format Specification

Introduction

The Device Firmware Upgrade specification specifies the contents of a data structure to be placed at the end of the DFU files, but leaves the definition of the content up to the vendor. This document describes the format used with the DFU solution from STMicroelectronics.

Contents

1 Device Firmware Upgrade STMicroelectronics Extension

The reason for creating this DFU extension is that the standard device firmware upgrade protocol is too specialized in terms of protocol versus the target or the application.

This extension makes it easy to use DFU with all 8 or 32-bit microcontrollers, simply letting the PC side know the target memory mapping. In addition, the DFU file format has been updated, to be able to build DFU files from standard 8 or 32-bit s19, hex or bin formats. The result is a new revision of the standard DFU revision 1.1, called DfuSe for Device firmware upgrade STMicroelectronics Extension.

2 DFU File format

The basic DFU file format to be used with STMicroelectronics DFU solution is based on three sections; Prefix, Images and Suffix, described as follows:

Figure 1. DFU File general format

DFU PREFIX
DFU Images
DFU SUFFIX

2.1 DFU Prefix section

The DFU prefix placed as a header file is the first part read by the software application, used to retrieve the file context, and enable valid DFU files to be recognized. The Prefix buffer is represented in Big Endian order.

The prefix packet is structured as follows:

Figure 2. DFUPREFIX

0	1	2	3	4	5	6	7	8	9	10
szSignature “DfuSe”					bVersion (0x01)	DFUImageSize				bTargets

- The szSignature field, five-byte coded, presents the file identifier that enables valid DFU files to be recognized, and incompatible changes detected. This identifier should be updated when major changes are made to the file format. This field is set to “DfuSe”.
- The bVersion field, one-byte coded, presents the DFU format revision, The value will be incremented if extra fields are added to one of the three sections. Software exploring the file can either treat the file depending on its specified revision, or just test for valid value.
- The DFUImageSize field, four-byte coded, presents the total DFU file length in bytes.
- the bTargets field, one-byte coded, presents the number of DFU image stored in the file.

2.2 DFU Suffix section

The DFU Suffix, as specified in the DFU specification, allows the host software to detect and prevent attempts to download incompatible firmware. The Suffix buffer is represented in Little Endian order.

The Suffix packet is structured as follows:

Figure 3. DFU Suffix

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bcdDeviceLo	bcdDeviceHi	idProductLo	idProductHi	idVendorLo	idVendorHi	bcdDFULo (0x1A)	bcdDFUHi (0x01)	ucDfuSignature (UFD)			bLength (16)	dwCRC			

- The bcdDevice field gives the firmware version contained in the file, or 0xFFFF if ignored.
- The idProduct and idVendor field give the Product ID and Vendor ID respectively of the device that the file is intended for, or 0xFFFF if the field is ignored.
- The bcdDFU field, fixed to 0x011A, gives the DFU specification number. This value differs from that specified in standard DFU rev1.1.
- The ucSignature field contains a fixed string of three unsigned characters (44h, 46h, 55h). In the file they appear in reverse order, allowing valid DFU files to be recognized.
- The bLength field, currently fixed to 16, gives the length of the DFU Suffix itself in bytes.
- The dwCRC (Cyclic Redundancy Check) field is the CRC calculated over the whole file except for the dwCRC data itself.

2.3 DFU Images section

The images section placed between the prefix and suffix as a body of the DFU file, contains a list of DFU images indexed by the alternate setting.

Figure 4. DFU Images*

DFU Image 1
DFU Image2
⋮
DFU Image N-1
DFU Image N

2.3.1 DFU Image

The DFU Image contains the effective data of the firmware, starting by a Target prefix record followed by a number of Image elements.

Figure 5. DFU Image

Target Prefix	
	Image Element 1
	ImageElement 2
	⋮

Figure 5. DFU Image

ImageElement N-1
ImageElement N

2.3.2 Target Prefix

The target prefix record is used to describe the associated image. The Target Prefix buffer is represented in Big Endian order.

Figure 6. Target Prefix

0	6	7	11	266	270 ---------- 273
szSignature “Target”	bAlternateSetting	bTargetNamed	szTargetName	dwTargetSize	dwNbElements

- The szSignature field, 6-byte coded, fixed to “Target”.
- The bAlternateSetting field gives the device alternate setting for which the associated image can be used.
- The bTargetNamed field is a boolean value which indicates if the target is named or not.
- The szTargetName field gives the target name.
- The dwTargetSize field gives the whole length of the associated image excluding the Target prefix.
- The dwNbElements field gives the number of elements in the associated image.

2.3.3 Image Element

The Image element structured as follows, provides a data record containing the effective firmware data preceded by the data address and data size. The Image Element buffer is represented in Big Endian order.

Figure 7. Image Element

0	1	2	3	4	5	6	7	8 ...
dwElementAddress				dwElementSize				Data

- The dwElementAddress field gives the 4-byte starting address of the data.
- The dwElementSize field gives the size of the contained data.
- The Data field present the effective data.

3 Revision history

Table 1. Document revision history

Date	Revision	Changes
15-Feb-2007	1	Initial release.