# USB Device Firmware Upgrade (DFU) with SAM5000 Series IC

This documentation describes briefly the content of the DREAM DFU tools package and should help to integrate USB DFU Firmware Update function into your SAM5000 based application.

## DFU Bootloader Firmware

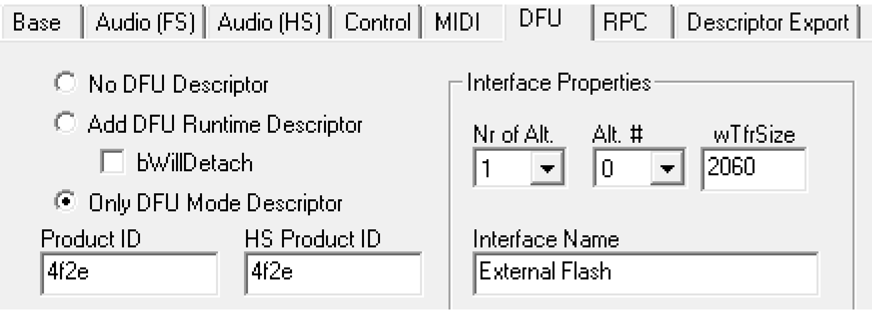
DFU Mode Bootloader firmware examples are available for Dream SAM5000 based evaluation boards.
The DFU Bootloader firmware is typically stored in an external serial Flash at address 0x0000:0000.

At start-up the DFU Bootloader firmware is automatically loaded in and executed in internal RAM.
Some hardware setting or action (jumper, button press, etc.) is sensed at start-up:

- If e.g. jumper is set or button is pressed, DFU Bootloader firmware enter the DFU mode
- If jumper is not set/button is not pressed, DFU Bootloader firmware checks if there is a valid Application firmware, e.g. at higher address location in same serial Flash, or in another Flash connected to SAM5000 chip, and reboot on Application firmware after its integrity has been confirmed. If Application firmware is not valid, DFU mode is kept.

The DFU Bootloader firmware initializes the memory I/F and specifies the correct memory driver.

Additionally the DFU Bootloader project contains a DFU-Mode USB Descriptor created with DREAM **UsbDescUI.exe** tool (available on Dream download area) to properly declare the DFU interface to the host computer. Settings for DFU Mode are done on “DFU” tab:



○ No DFU Descriptor
○ Add DFU Runtime Descriptor
□ bWillDetach
● Only DFU Mode Descriptor

Product ID: 4f2e
HS Product ID: 4f2e

Interface Properties
Nr of Alt.: 1
Alt. #: 0
wTfrSize: 2060
Interface Name: External Flash

Please be aware of correct settings for Vendor ID, Product IDs and wTfrSize, see also paragraph “DFU Driver for Windows” on next page.

The Firmware Signature of the Bootloader firmware and Application firmware must be set correctly to get the right startup sequence. The Firmware Signature is set in SamVS Project Settings on “Output” tab.

- The Bootloader firmware must have signature “4452” (hex, =’DR’) to be detected as valid boot firmware
- The Application firmware must not have signature “4452” so it will not be “seen” as valid firmware from SAM5000 ROM boot. In our examples, Application firmware needs signature “6472” (=‘dr’) to be recognized as valid by the DFU Bootloader firmware.

After all setting have been done create a binary image from the DFU Bootloader project, which must then be loaded into serial Flash device connected to SAM5000 IC.

The Application firmware binary image file must be converted into a “DFU” file, usable by DFU Loader tools, for this see paragraph “Tool Sam2DFU - DREAM DFU File Builder” on next page.

You may also create a DFU file directly from SamVS, see Output Tab in project settings (from V4.610).

## DFU Driver for Windows

To support the DFU protocol on Windows (7 and up), no matching driver is required. If Windows detects the DFU device, it should contact Windows Update to retrieve proper drivers. If automatic driver updates are not enabled, this step will obviously fail.

To support the DFU protocol on older Windows, a matching driver is required. There is no way to support the DFU protocol without providing a custom driver. For customers not providing own drivers which may be extended to forward DFU requests from the application to the device a WinUSB based driver is provided. WinUSB is a special API provided by Microsoft to support application side access to USB devices and should provide best possible compatibility to any supported version of Windows (supported from Windows XP and up).

Windows matches drivers to devices using USB Product- and Vendor-IDs, so it needs some additional information. On newer Windows versions (Windows 8.1 and up) this information is retrieved from the device automatically. On legacy versions (Windows 8.0, 7 and lower) you need to provide an INF file. This INF file must contain proper entries for the particular device in order to be used by Windows. Since Product-IDs differ between Runtime- and DFU-Mode and the INF file supports both 32 and 64 bit versions, in fact four entries must be customized. The relevant section in the provided file looks like this:

```
[DreamDFU_WinUSB.NTx86]
%USB\DreamDFU.DeviceDesc% =USB_Install_x86, USB\VID_2485&PID_4f2e        ; DREAM 5704FX-EK DFU
%USB\DreamDFU.DeviceDesc% =USB_Install_x86, USB\VID_2485&PID_422e&MI_04  ; DREAM 5704FX-EK regular

[DreamDFU_WinUSB.NTamd64]
%USB\DreamDFU.DeviceDesc% =USB_Install_a64, USB\VID_2485&PID_4f2e         ; DREAM 5704FX-EK DFU
%USB\DreamDFU.DeviceDesc% =USB_Install_a64, USB\VID_2485&PID_422e&MI_04   ; DREAM 5704FX-EK regular
```

You may simply duplicate each four lines and modify IDs accordingly, so one INF file will support multiple devices.

## DFU Driver for Mac

No driver is required for the Mac operating system, applications are free to directly enumerate and connect to USB Devices without Driver Support.

## Tool Sam2DFU - DREAM DFU File Builder

DFU Loaders like “DreamDFU” (see next page) requires a specially structured file which contains the firmware image and additional information to conform to the USB DFU specification. Such a DFU file is also secured by a CRC checksum to avoid flashing a corrupted firmware into the device. The “Sam2DFU” DREAM DFU file builder tool is provided to create such files.

To ensure only a compatible firmware is programmed into a particular device, the DFU file also contains USB Product- and Vendor-IDs, so the correct device can be selected by the client application.

All the necessary parameters have to be defined on the command line, so it's probably a good idea to create a command (CMD) file containing the proper parameters. An example CMD file is provided as a reference.

The command line syntax of Sam2DFU is as follows:

```
Sam2DFU <Options> <OutputFile> -A<LoadAddress>[,<FileOff>[,<ImgLen>]] <SrcFile> ...
```

The following options have to be defined, since there are no defaults (except Version-Number):

```
-V<VendorID>       : USB VendorID
-P<ProductID>      : USB ProductID in Runtime Mode
-U<DfuProductID>   : USB ProductID in DFU Mode
-N<bcdVersion>     : Version of Firmware
-T<TransferSize>   : Packet Size, must be same as <wTfrSize> specified in UsbDescUI
-C<StringComment>  : Comment
```

All values may be specified in decimal or hex. Hex values must be prefixed with 0x, so the first parameters may look like this (which matches the example):

```
sam2dfu -v 0x2485 -p 0x432e -u 0x4f2e -t 2060 5704FX-EK_Test.dfu
```

The remaining command line specifies pairs of address information and filenames. The address information declares where in device memory to write the following file.

Normally the whole file will be transferred to that position, but Sam2DFU can also transfer portions of the file to different memory locations. For this operation the optional parameters FileOff and ImgLen may be specified.

FileOff specifies a (16 Bit) Word offset within the file to define the start position, ImgLen specifies the number of words beginning at the given FileOff which should be written to the current block. Consequently Sam2DFU may split large firmware files into multiple sections which are programmed to different locations.

Of course, to flash Application firmware file into a single SPI Flash, a single address specification will be used:

```
    -A0xC0004000 5704FX-EK_Test.bin
```

Above example is used in case the Application is located in serial Flash at (word) address 0x4000 (e.g. above the DFU Bootloader firmware in same Flash) connected to the "GP" SPI memory port of SAM5000 IC (mapped to address range 0xC000:0000 and up).

Please note that specified address must match range specified within the range table in the used DFU Bootloader firmware.

## Tool DreamDfuMaker – DFU Client Application Builder

The "DreamDFU" is an example of a DFU Loader tool for Windows and MAC. It is generated by the DreamDfuMaker provided in Dream DFU tool kit. DreamDfuMaker.exe allows the customization of DreamDFU client application, following resources can be changed:

- Dialog title
- Company logo (arbitrary size, displayed in top left corner)
- Application icon (ICO file should contain 16x16 and 32x32 pixel images)
- Background image (BMP image, size about 800x580 pixels)

After replacing default resources press [Save Application], choose a filename and click OK. This will export both a Windows EXE file and a Mac OSX application (visible as a directory tree from Windows).

## Tool DreamDFU – DFU Client Application

The "DreamDFU" DFU Loader tool takes a DFU file and tries to detect a matching device and finally performs the update.

On startup "DreamDFU" open the "Select DFU File" windows. User should open the provided DFU file, which contains information about the device to look for. Then the application enumerates USB connections for compatible devices and, if found, displays Vendor and Product Name, Vendor and Device IDs. Performing the DFU update is started when pressing [Update Device] button.