

无法使用内置 Bootloader 的 DFU 方式进行固件升级

1 前言

本文将针对客户无法使用内置 Bootloader 的 DFU 方式进行固件升级的问题进行分析。

2 问题描述

客户使用的是 STM32F205VET6，做了个最小系统测试板，在 BOOT0=1，BOOT1=0 的情况下连接 PC，使用 PC 端软件 DfuSeDemo 无法检测到 DFU 设备，但是同样在 Bootloader 模式下，却可以通过串口 1 进行固件升级。

3 问题分析

首先怀疑的是 USB 线路问题，因此，在切换到正常模式(BOOT0=0,BOOT1=0)时，使用 CubeMx 做了个简单的鼠标 HID 测试程序验证，结果发现在正常模式下测试程序是能正常运行的，从这点可以说明 USB 不存在线路不通的问题。

其次，检查各个管脚的电平，VDD，BOOT0，BOOT1 均未发现异常。

于是打开应用文档 AN2606-STM32 microcontroller system memory boot mode.pdf，通过此应用文档可知，不同的 Bootloader 版本可用于固件升级的方式不尽相同，如 3.2 节如下内容：

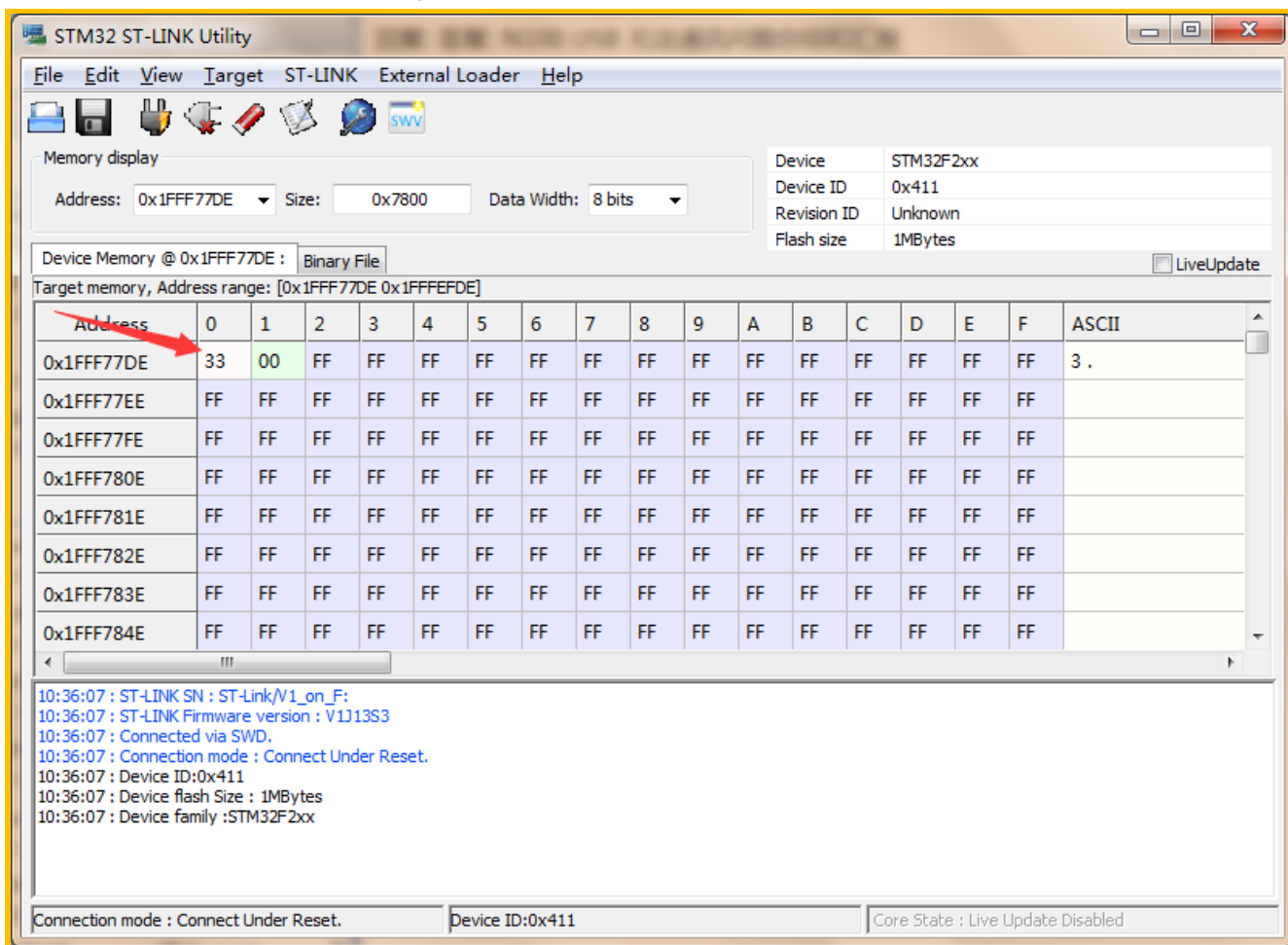
2. **Bootloader identifier (ID):** version of the STM32 device bootloader, coded on one byte in the 0xXY format, where:
- **X** specifies the embedded serial peripheral(s) used by the device bootloader:
 - X = 1: one USART is used
 - X = 2: two USARTs are used
 - X = 3: USART, CAN and DFU are used
 - X = 4: USART and DFU are used
 - X = 5: USART and I²C are used
 - X = 6: I²C is used
 - X = 7: USART, CAN, DFU and I²C are used
 - X = 8: I²C and SPI are used
 - X = 9: USART, CAN, DFU, I²C and SPI are used
 - X = 10: USART, DFU and I²C are used
 - X = 11: USART, I²C and SPI are used
 - X = 12: USART and SPI are used
 - X = 13: USART, DFU, I²C and SPI are used
 - **Y** specifies the device bootloader version
 - Let us take the example of a bootloader ID equal to 0x10. This means that it is the first version of the device bootloader that uses only one USART.
 - The bootloader ID is programmed in the last byte address - 1 of the device system memory and can be read by using the bootloader "Read memory" command or by direct access to the system memory via JTAG/SWD.

因此怀疑此 MCU 的 BID 是否会不支持 DFU？通过上图可知，BID 可以通过 SWD 直接读取，因此我们需要找到保存此 BID 信息的地址。

通过应用文档 AN2606 3.2 节的表 3：

F2	STM32F2xxx	USART1/USART3	0x20	0x1FFF77DE	USART (V3.0)
		USART1/USART3/ CAN2/ DFU (USB Device FS)	0x33	0x1FFF77DE	USART (V3.1) CAN (V2.0) DFU (V2.2)

如上图可知，STM32F2 的 Bootloader 存在两种 BID，可以通过地址为 0x1FFF77DE 这个地址的值来获取，如为 0x20 则只支持 USART，若为 0x33，则支持 USART，CAN，DFU 这 3 种方式。于是使用 PC 端软件 STM32 ST-LINK Utility 通过 SWD 读取 0x1FFF77DE 这个地址的值，如下图所示：



The screenshot shows the STM32 ST-LINK Utility interface. The 'Memory display' section shows the address 0x1FFF77DE with a size of 0x7800 and data width of 8 bits. The 'Device Memory @ 0x1FFF77DE' section shows a binary file. The 'Target memory, Address range: [0x1FFF77DE 0x1FFFEFDE]' section displays a memory dump table. The table has columns for Address, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, and ASCII. The first row shows the value 33 at address 0x1FFF77DE, which is highlighted in green. The ASCII column shows '3'.

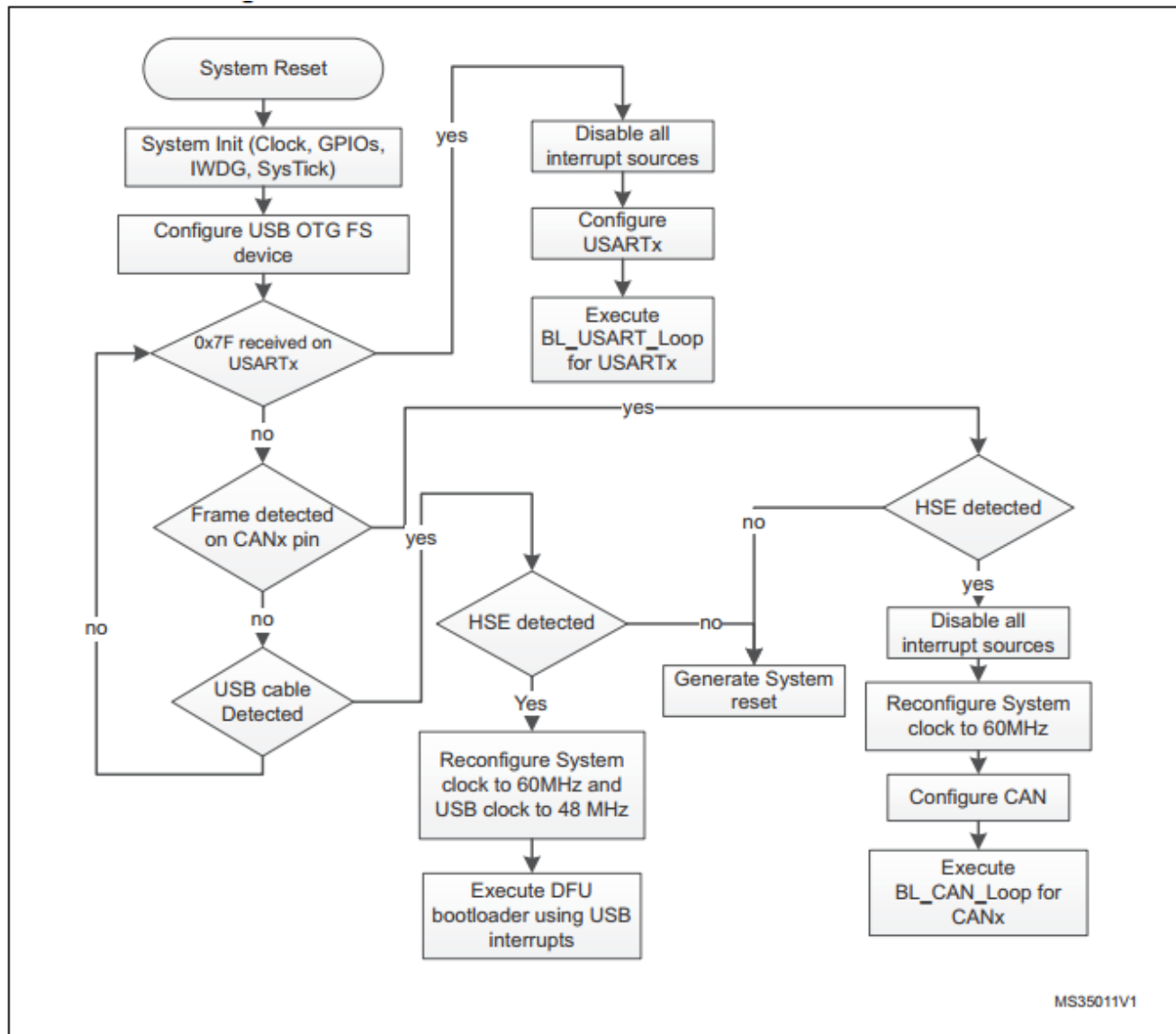
Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
0x1FFF77DE	33	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	3
0x1FFF77EE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x1FFF77FE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x1FFF780E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x1FFF781E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x1FFF782E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x1FFF783E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x1FFF784E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

The bottom status bar shows: Connection mode : Connect Under Reset. Device ID:0x411 Core State : Live Update Disabled

如上图，可见客户使用的 STM32F205 的 BID 为 0x33，是同时支持 USART，CAN 和 DFU 这 3 种方式的，因此，排除 Bootloader 版本问题的可能性。

在上述可能性都排除外，客户提出怀疑芯片本身或 Bootloader 烧录的代码有问题，于是找出一块 STM32F4-DISCOVERY 板进行 MCU 替换，替换后的结果为 STM32F205 在放到 DISCOVERY 板上则能正常通过 Bootloader 的 DFU 方式进行固件升级，因此，这就明确排除了芯片本身问题的可能性，因此，只可能是用户板子外围电路的问题。

再次回到 AN2606 这个应用文档，在 15.2.2 节找到 Bootloader 的工作流程图，如下所示：



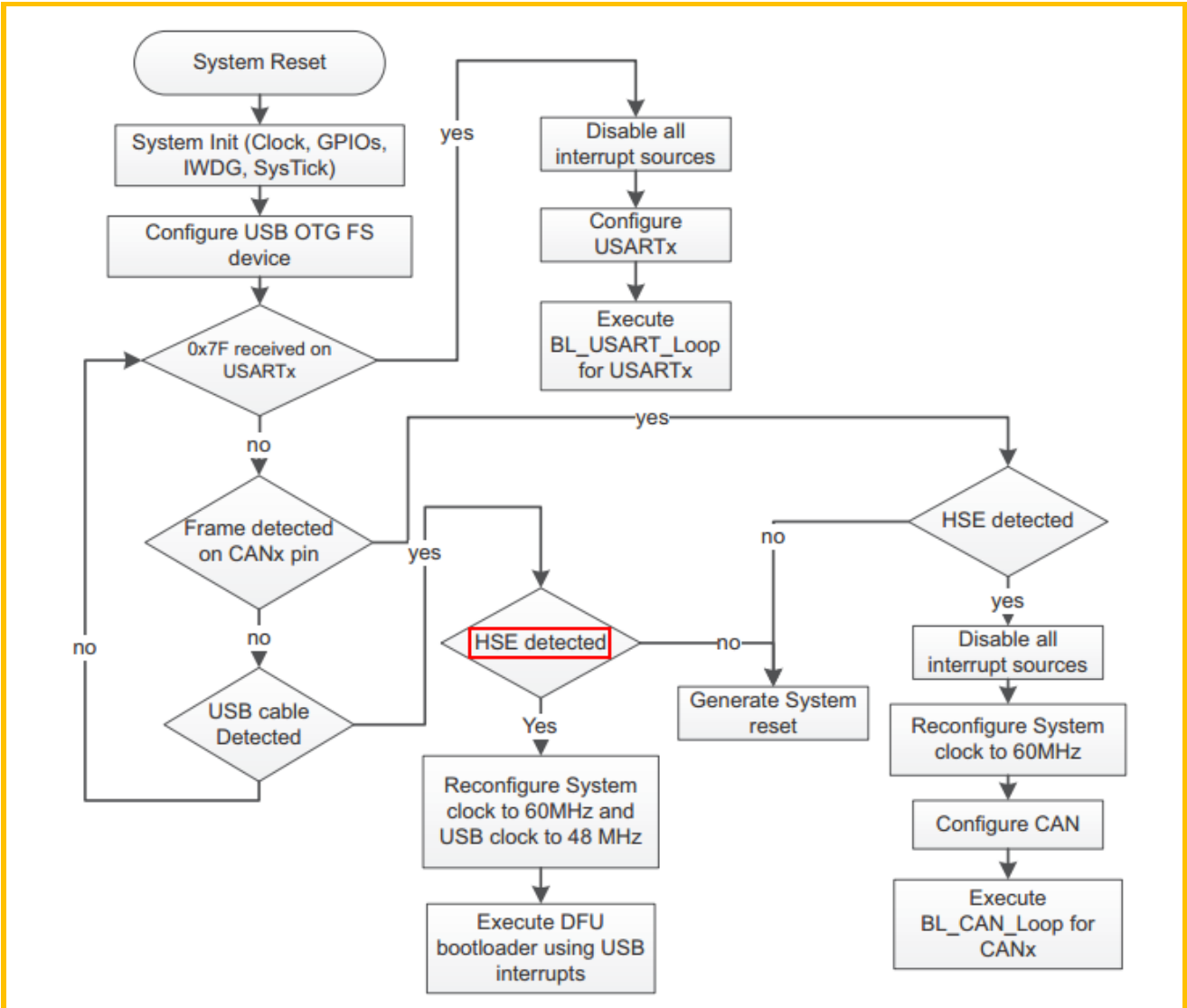
通过上图可知，Bootloader 是依次检查 USART->CAN->DFU 的方式，怀疑 Bootloader 程序在 DFU 之前由于某种未知原因是否已经进入到 USAR 或 CAN 的方式中而一直没有出来？

为了排除这种可能性，我们针对 USART1 的 RX 脚 PA10，USART3 的 RX 脚 PB11 和 PC11 拉高，同时将 CAN2 的 RX 脚 PB5 拉低进行测试，结果还是无法检测到 DFU 设备。

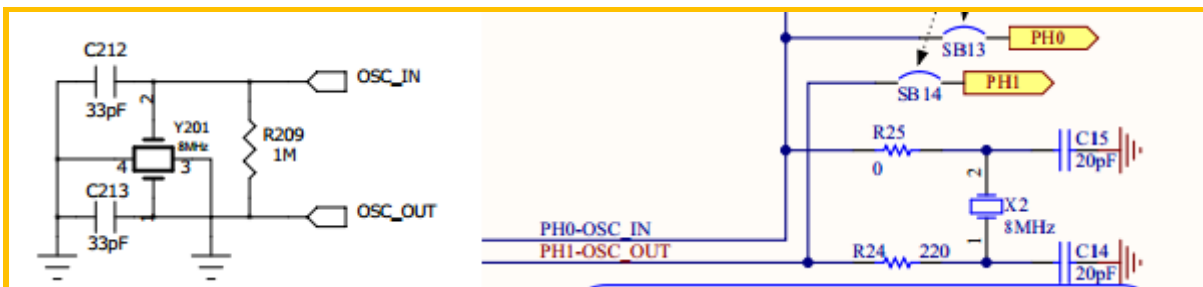
再次回到上图进行分析，如上图，若 USART 和 CAN 都没有检测到的话，Bootloader 程序会检测 USB 线是否连接，然后检测外部 HSE，若 HSE 不存在，则产生系统复位，否则将会重现配置系统主频到 60M。

由于我们是连着 USB 线且在正常运行模式下 USB 是能正常工作的，因此，这里检测 USB 线结果应该是通过的，于是按照程序流程，接下来检测外部 HSE，若检测失败则复位系统。与是用示波器查看

VDD 与 NRST 脚的波形，发现系统在 VDD 上电后有 3 次复位，如此，可以得出 Bootloader 程序在检测外部 HSE 时结果为失败，如下：



为什么会检测外部 HSE 失败？用户使用的 HSE 是 8M 晶振，与 DISCOVERY 板一样都是 8M 外部晶振，对比用户的外部晶振电路与 DISCOVERY 的对应电路，如下图所示：



如上图，左边为客户板子的晶振电路，右边为 DISCOVERY 板的晶振电路，对比可知，用户的负载电容使用的是 33pF，且多了个 1M 的反馈电阻。

首先将反馈电阻去掉后测试，结果还是一样。进一步将客户板子的晶振负载电容换成 20pF 后进行测试，结果可以正常检测到 DFU 设备，如此可见，正是因为这个负载电容的原因造成 Bootloader 的 DFU 无法正常工作！

4 总结

此问题是由于晶振负载电容过大，导致内置 Bootloader 程序在检测外部 HSE 的时间点与实际 HSE 稳定震荡所需的时间不同步造成，结果就是检测不到 HSE，进而引起系统复位，最终无法使用 Bootloader 的 DFU 方式进行固件升级。

5 本文所涉及到的文档与软件下载链接

AN2606	http://www.stmcu.org/document/detail/index/id-200918
DfuSeDemo	http://www.stmcu.org/document/detail/index/id-214339
STM32 ST-LINK Utility	http://www.stmcu.org/document/detail/index/id-215840

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。