

通过 DfuSe 工具控制程序跳进 DFU 模式

1 前言

我们之前经常有讲到过如何通过 USB 的 DFU 方式来对固件进行升级，在示例中我们通常是通过一个按键来触发 APP 跳转到 BOOT 从而进入到 DFU 升级模式。但是也有一种情况，客户的环境是没有任何按键或者其它触发条件，只有一个 USB 口连接运行 windows 操作系统的上位机。这不，前不久就碰到这么个客户，下位机 MCU 端仅仅只有一个 USB 口连接上位机，且 MCU 端与上位机之间的 USB 是常连的，并不能通过 USB 的拔插操作来触发 APP 的跳转，在此情况下，我们又该如何来实现客户的需求？

2 分析



Figure 1 BOOT 与 APP 在内部 FLASH 中的关系

如上图所示，在 MCU 内部 FLASH 中，BOOT 与 APP 同时存在，BOOT 为 DFU 模式，负责对固件进行升级，而 APP(Run-time 模式)为客户的应用程序。当 APP 运行时，通过某个事件触发(DFU_DETACH)程序从 APP 跳转到 BOOT 中从而进入到 DFU 模式中。

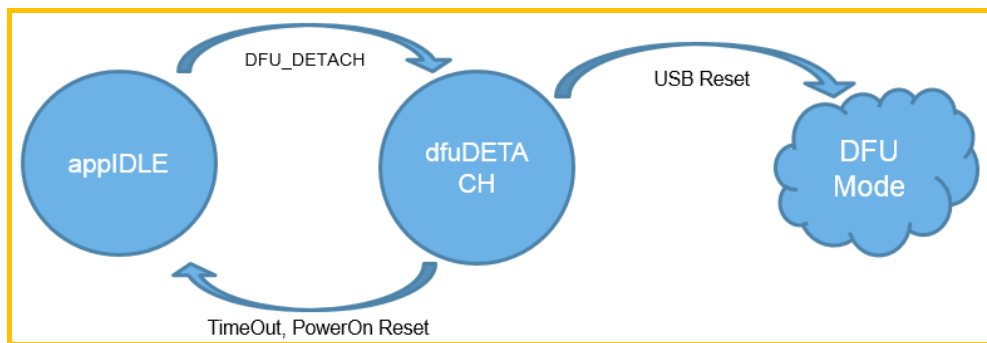


Figure 2 reconfiguration phase

这个 DFU_DETACH 的操作在这里我们可以通过 PC 端软件 DfuSe Demo 来触发。

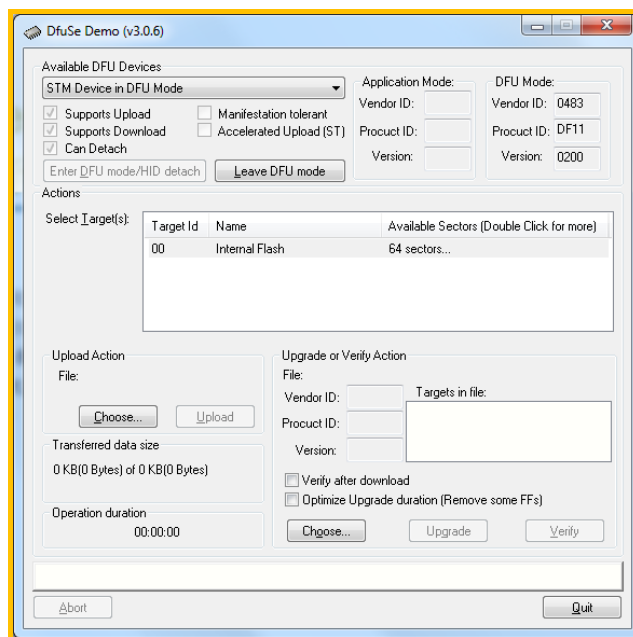


Figure 3 DfuSe Demo

在进入到 DFU 模式时，DfuSe Demo 软件的运行界面如上所示，上图界面中有一个“Leave DFU mode”的按钮，通过它可以使运行在 MCU 的程序从 DFU 模式切换到 APP 中。同时我们也发现，在按钮“Leave DFU mode”的左边有一个灰色按钮“Enter DFU mode/HID detach”，从字面就可以知道，它应该是进入到 DFU 模式的按钮，在 APP 模式下此按钮应该是激活的，但是，在进入到 APP 模式后，此界面就检测不到 DFU 设备了，此软件将不再可用。那么我们又将如何来激活这个“Enter DFU mode/HID detach”按钮呢？

从帮助文档可知，在 APP 模式时，当 DfuSe Demo 这个软件识别到一个特定的 HID 设备时，“Enter DFU mod/HID detach”按钮将激活，通过进一步了解，我们知道此时 DfuSe Demo 这个软件是需要使能此项功能才可以，所幸地是，我们已经有了此软件，在与此文对应的附件中，可以找到一个名为 **DfuSeDemo_A.exe** 的可执行程序，将它拷贝到 DfuSeDemo 的安装目录下，如下图所示：

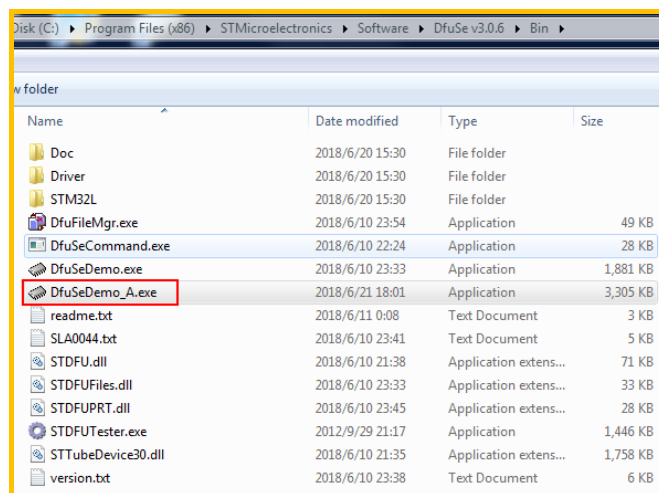


Figure 4 DfuSeDemo_A

当运行在 APP 模式下时如下所示：

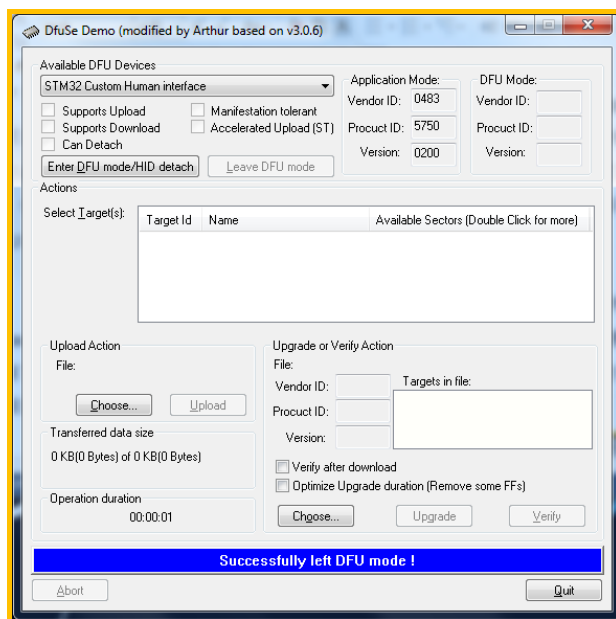


Figure 5 APP 模式

如上图所示，在 APP 模式下时，当程序检测到有效的 HID 设备时，DfuSe Demo 的“Available Dfu Device”下拉框中会显示这个 HID 设备，并且“Enter Dfu mode/HID detach”按钮激活。当按下此键时，MCU 会重启并进入到 DFU 模式。当然此 APP 是需要实现一个符合某种要求的 HID 设备才行。下面我们通过制作这样一个 APP 来讲解。

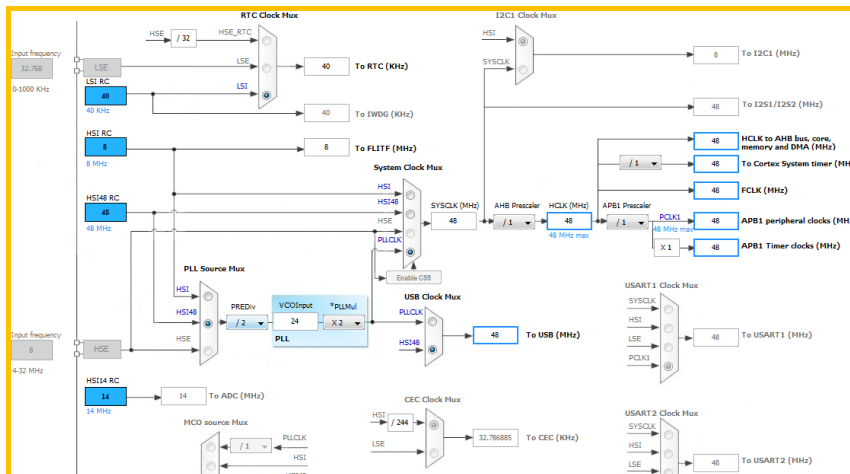
3 制作 APP

我们基于之前 DFU 培训时使用的 STM32F072-Discovery 板来实现此 APP。

打开 STM32CubeMx 软件新建一个基于 STM32F072RB 的工程，使能 USB 和 RTC 外设，RTC 我们主要用它的备份域寄存器，用来保存跳进入到 DFU 模式的标志；

Pinout 如下所示：

使用 HSI48 作为系统时钟源，同时给 USB 提供 48M 时候，这样可以省一个外部晶振，RTC 就使用 LSI 时钟源，我们不要求它的时间精确度，只求备份域寄存器可用。



然后将 USB 类选择“Custom Human Interface Device(HID)”，USB 中断等级设置为 1，堆为 0x500，栈大小设置为 0x2000，然后生成工程。

```
#define USBD_CUSTOM_HID_REPORT_DESC_SIZE      23
/** Usb HID report descriptor. */
__ALIGN_BEGIN static uint8_t CUSTOM_HID_ReportDesc_FS[USBD_CUSTOM_HID_REPORT_DESC_SIZE]
__ALIGN_END =
{
    /* USER CODE BEGIN 0 */
    0x06, 0x00, 0xFF,      /* USAGE_PAGE (Vendor Page: 0xFF00)      */
    0x09, 0x01,            /* USAGE (Demo Kit)                      */
    0xa1, 0x00,            /* COLLECTION (Physical)                 */

```

```

0x85, 0x80,          /* REPORT_ID (128) */
0x09, 0x55,          /*USAGE (LED 1)          */
0x15, 0x00,          /* LOGICAL_MINIMUM (0)    */
0x26, 0xFF, 0x00,    /* LOGICAL_MAXIMUM (255)  */
0x75, 0x08,          /* REPORT_SIZE (8 bits)    */
0x95, 0x01,          /* REPORT_COUNT (1)       */
0xB1, 0x82,          /* FEATURE (Data,Var,Abs,Vol) */
/* USER CODE END 0 */
0xC0 /*      END_COLLECTION      */
};

```

然后找到 CUSTOM_HID_OutEvent_FS 函数，修改为：

```

static int8_t CUSTOM_HID_OutEvent_FS(uint8_t event_idx, uint8_t state)
{
    /* USER CODE BEGIN 6 */
    extern RTC_HandleTypeDef hrtc;
    if(event_idx == 0x80 && state == 0x55)
    {
        __HAL_RCC_PWR_CLK_ENABLE();
        HAL_PWR_EnableBkUpAccess();
        HAL_RTCEx_BKUPWrite(&hrtc, RTC_BKP_DR0, 0x32F2);
        HAL_PWR_DisableBkUpAccess();
        HAL_NVIC_SystemReset();
        while (1);
    }
    return (USBD_OK);
    /* USER CODE END 6 */
}

```

如上代码所示，只有当 PC 端软件 DfuSe Demo 检测到具有这个报告描述符的 HID 设备后才会认可其为满足要求的 HID 设备，并将其显示在设备列表中，按键“Enter DFU mode/HID detach”激活。当用户按下此按键后，DfuSe Demo 会向 HID 设备发送一条 SetFeature 指令。

当 MCU 收到 SetFeature 指令后，往备份域寄存器写入 0x32F2 作为进入 DFU 模式的标志，然后重启切换到 BOOT，接下来在 BOOT 内如果程序检测到备份域寄存器的标志时则进入到 DFU 模式。整个过程如图 2 所示。

当然，APP 还需要修改 FLASH 中的偏移位置，通过工具烧录到 0x0800 7000 的位置，这个在之前 DFU 培训时已经讲述过如何操作了，这里就不具体再重复了，有兴趣的同学可以查看此文档附件内的 APP 源码。

接下来我们再来看看 BOOT 程序中检测跳转标志的过程：

```

/* USER CODE BEGIN 2 */
if ((((__IO uint32_t*)USBD_DFU_APP_DEFAULT_ADD) & 0x2FFE0000) != 0x20000000)
{
    jump2App = JUMP_FLAG_DFU;
}

```

```
if(jump2App !=JUMP_FLAG_DFU)
{
    __HAL_RCC_PWR_CLK_ENABLE();
    HAL_PWR_EnableBkUpAccess();
    if(0x32F2 ==HAL_RTCEx_BKUPRead(&hrtc,RTC_BKP_DR0))//检测备份域寄存器中的标志
    {
        HAL_RTCEx_BKUPWrite(&hrtc,RTC_BKP_DR0,0);
        jump2App = JUMP_FLAG_DFU;
    }
    HAL_PWR_DisableBkUpAccess();
}

if(jump2App == JUMP_FLAG_INIT)
{
    if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0) !=GPIO_PIN_SET)
    {
        jump2App = JUMP_FLAG_APP;
    }
    else
    {
        jump2App =JUMP_FLAG_DFU ;
    }
}

if(jump2App ==JUMP_FLAG_APP)
{
    /* Jump to user application */
    JumpAddress = *(__IO uint32_t*) (USBD_DFU_APP_DEFAULT_ADD + 4);
    JumpToApplication = (pFunction) JumpAddress;

    /* Initialize user application's Stack Pointer */
    __set_MSP(*(__IO uint32_t*) USBD_DFU_APP_DEFAULT_ADD);
    JumpToApplication();
}
//进入到 DFU 模式
MX_USB_DEVICE_Init();
/* USER CODE END 2 */
```

如上代码所示，只有当 `jump2App ==JUMP_FLAG_APP` 时程序才会跳转到 APP 中运行，否则进入到 DFU 模式。

4 总结

使用此方法毕竟在 APP 中实现了一套 HID，如果不是必要的话，建议还是使用按键的方式来触发从 APP 跳转到 DFU 模式，除非遇到像本文中客户的情况，或者 APP 本身就需要实现某个 USB 类的功能，此时可以做成 USB 复合设备，其中一个 HID 设备就是本文中的 HID 设备，这样 PC 端的软件 DfuSe Demo 也可以识别。

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。