

利用 ST 提供的 USB 例程实现 USB IAP 功能

Grant_jx@163.com

我们知道 ST 推出的 Cortex-M3 平台 STM32 内部有两个 Flash 区域，一是 System Flash，ST 官方保留的一个区域，用于存放 IAP 代码。该区域不对用户开放，仅提供 UART 的通讯接口用于 IAP 升级；另一个区域是 User Flash，这一部分是供用户自由使用的。STM32 自带 USB 2.0 Device 接口，如果需要通过 USB 接口来实现 IAP 功能需要如何做呢？这里介绍如何利用 ST STM32xx USB Development Kit 提供的 DFU 代码来实现上述功能。我用 STM3210EVB 来演示这个功能。

阅读前请下载下列代码压缩包和应用软件：

1、STM3210B-EVAL demonstration software v1.1.zip

[STM3210B-EVAL demonstration software v1.1.zip](/bbs_upload/files_9/ourdev_243528.zip) (文件大小:2.15M)

ST 官方的下载路径：

<http://www.st.com/stonline/products/support/micro/files/um0435.zip>

2、STM32F10xxx USB developer kit v1.0.zip

[STM32F10xxx USB developer kit v1.0.zip](/bbs_upload/files_9/ourdev_243529.zip) (文件大小:461K)

ST 官方的下载路径：

<http://www.st.com/stonline/products/support/micro/files/um0424.zip>

3、DfuSe USB Device Firmware Upgrade v2.2.zip

[DfuSe USB Device Firmware Upgrade v2.2.zip](/bbs_upload/files_9/ourdev_243530.zip) (文件大小:7.85M)

ST 官方的下载路径：

<http://www.st.com/stonline/products/support/micro/files/um0412.zip>

上述文档对于的用户参考手册，请自行到 ST MCU 网站下载。

<http://www.st.com/mcu/familiesdocs-110.html#Application%20Note>

下列步骤将介绍如何通过 ST 官方的 USB 升级代码实现程序的下载更新的功能(IAP)。

1、打开 STM3210B-EVAL demonstration software 压缩包，在 STM3210B-EVAL demonstration software\Demo\source 下打开 main.c 文件，找到 void InterruptConfig(void) 函数

```
/* Set the Vector Table base address at 0x08000000 */  
NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x00);
```

这里我们需要修改代码的中断向量起始地址，这样做的目的是为了处理 IAP 代码在 Flash 存放的区域与 Application Code 部分的存放空间不会发生地址冲突。这里我们假设 IAP 存放在 User Flash 的 0x08000000 ~ 0x08003FFF 区域，Application code 存放在 User Flash 的 0x08004000 ~ 0x0801FFFF 区域。因为 Application code 的开始地址是由 0x08004000 开始，这样我们需要为应用代码的中断向量地址做一个重映射。因此我们修改该代码为：

```
/* Set the Vector Table base address at 0x08004000 */  
NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x4000);
```

请注意这里 NVIC_SetVectorTable 函数的型参送入的是相对偏移地址，而不是绝对地址；

2、在 STM3210B-EVAL demonstration software\Demo\project\EWARM 下找到 Inkarm_flash.xcl 文件，在 XCL 文件中找到下面的配置，该配置用于定制应用代码在 Flash 区域的存放空间和代码运行是 RAM 可以提供的空间。

```
// Code memory in FLASH  
-DROMSTART=0x8000000  
-DROMEND=0x801FFFF
```

```
// Data in RAM  
-DRAMSTART=0x20000000  
-DRAMEND=0x20004FFF
```

由于我们的目标应用代码将是在 0x08004000 区域运行，因此我们修改为：

```
// Code memory in FLASH  
-DROMSTART=0x8004000  
-DROMEND=0x801FFFF
```

```
// Data in RAM  
-DRAMSTART=0x20000000  
-DRAMEND=0x20004FFF
```

在编译的时候请确保 Project ->Options ->Linker ->Config 标签下的链接命令文件选择的是上述 Inkarm_flash.xcl 文件；

3、应用部分改好，现在我们修改 USB 固件升级部分的代码，打开 STM32F10xxx USB developer kit 开发包。

4、在开发包下面找到 \STM32F10xUSBLib\USBLib\demos\Device_Firmware_Upgrade 例程，该例程是一个在 STM32F10xx 系列 MCU 上实现运行在 User Flash 区域的 IAP 自升级代码，通过 STM32 自身提供的 USB 接口实现。

在\STM32F10xUSBLib\USBLib\demos\Device_Firmware_Upgrade\source 路径下找到 main.c 文件，在 56 行：

```
if (DFU_Button_Read() != 0x00)
{
    /* Test if user code is programmed starting from address 0x8003000 */
    if (((*(vu32*)0x8003000) & 0x2FFF0000) == 0x20000000)
    {
        /* Jump to user application */

        JumpAddress = *(vu32*) (ApplicationAddress + 4);
        Jump_To_Application = (pFunction) JumpAddress;
        /* Initialize user application's Stack Pointer */
        __MSR_MSP(*(vu32*) ApplicationAddress);
        Jump_To_Application();
    }
}
/* Otherwise enters DFU mode to allow user to program his application */
```

这段代码的功能是对应用部分的代码开始地址做判断，这里的地址与我们之前的步骤 1、2 都是对应的。

同样我们对代码做如下更改：

```
/* Test if user code is programmed starting from address 0x8004000 */
if (((*(vu32*)0x8004000) & 0x2FFF0000) == 0x20000000)
```

编译代码，下载到 STM3210 Evaluation Board。

5、hw_config.h 中定义：

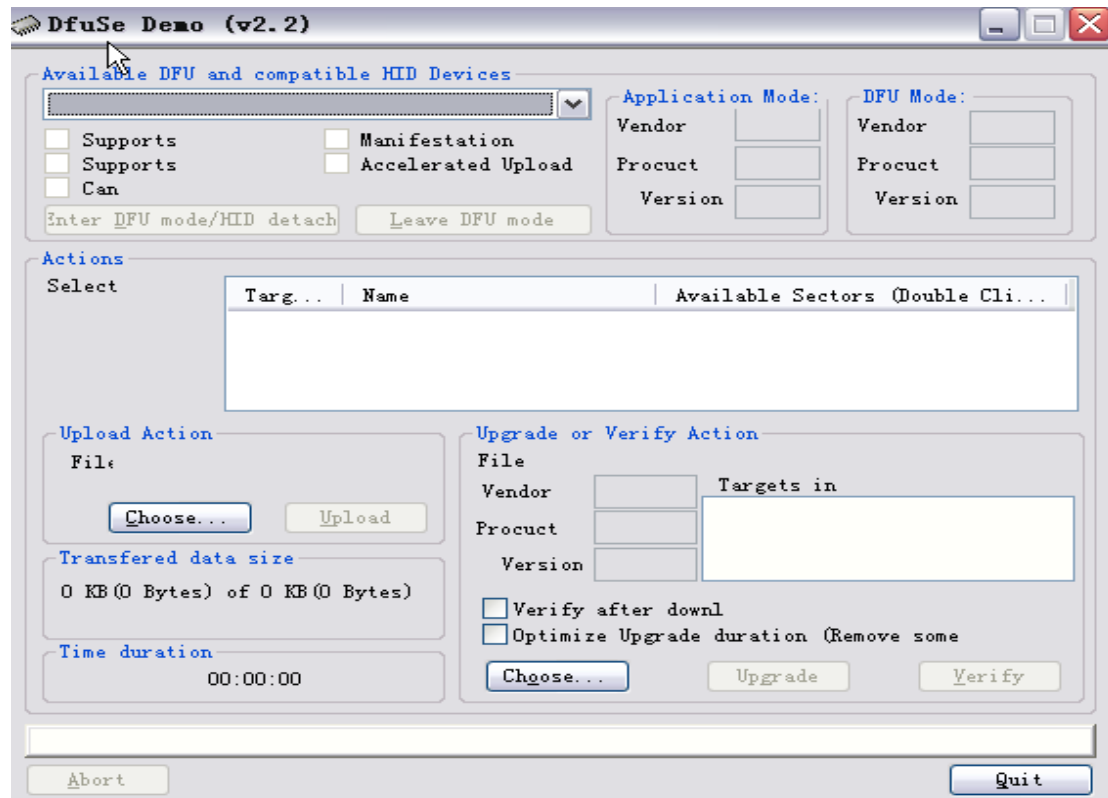
```
#define ApplicationAddress 0x08003000
```

改为

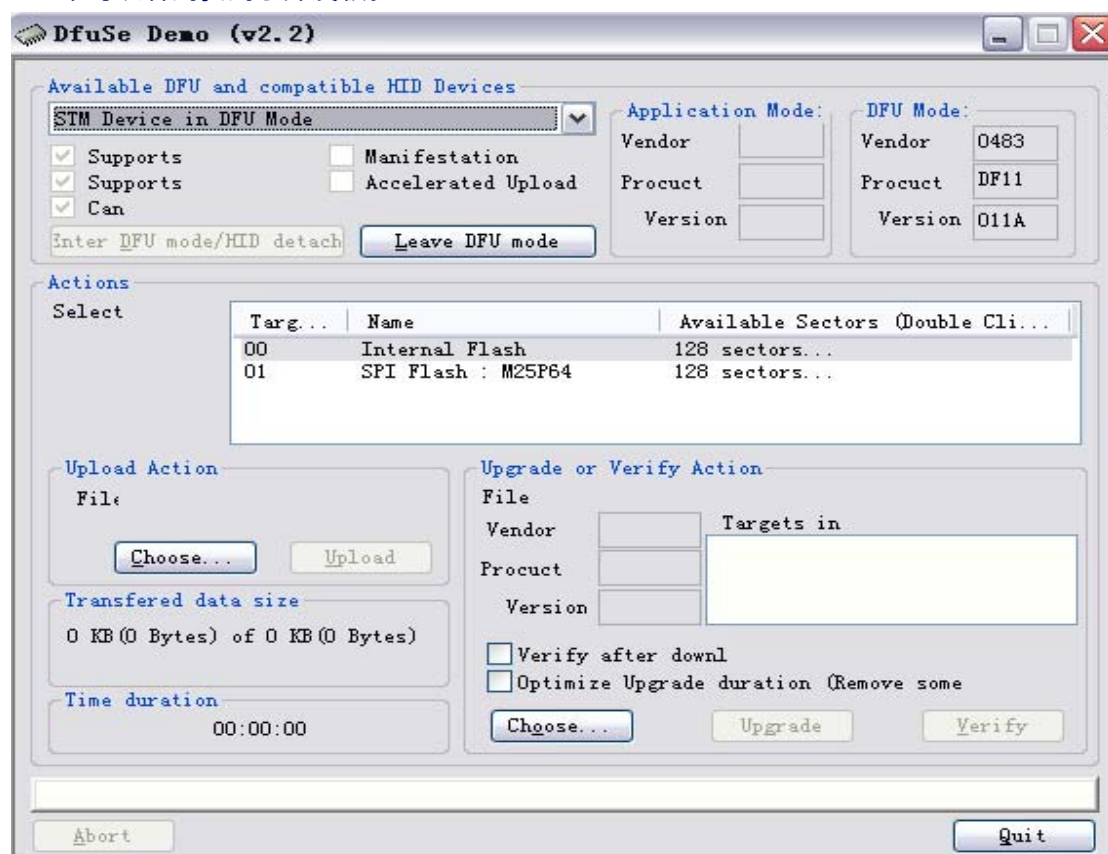
```
#define ApplicationAddress 0x08004000
```

编译代码，下载到 STM3210 Evaluation Board。

6、在 ST 的网站中找到 USB IAP 的 PC 端用于程序 DfuSe USB Device Firmware Upgrade，安装后执行 DfuSe Demonstration 程序。



7、改变 STM3210 Evaluation Board 的 Boot 选项为 user Flash 启动，启动时在 PC 端 Dfu 中可以看到找到了开发板。

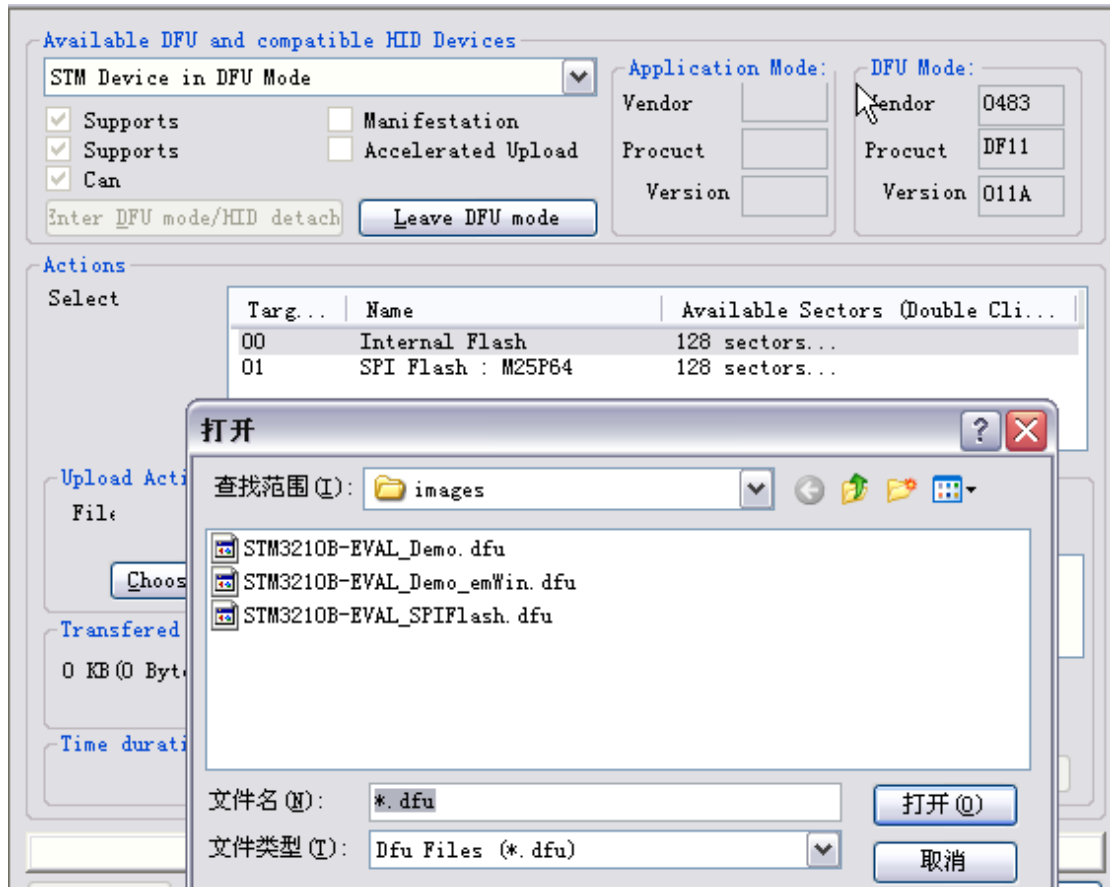


如果提示需要找到新的设备，需要安装驱动，请将路径指向 Dfu 的安装路径下 driver 目录。如果 Dfu 中无法看到找到开发板，请按住 Reset 键再松开；如果按下 Key 键，STM32 中运行的 Dfu 程序将退出 IAP 状态，PC 跳到事先设置的应用程序的入口。

Dfu 提示找到 STM3210EVB，显示需要编程的区域，一个是 SPI Flash，另一个是 STM32 内部的 Flash。

SPI Flash 的内容主要是提供 LCD 演示的图象和语言数据，需要写入的文件在 STM3210B-EVAL demonstration software 开发包的目录\STM3210B-EVAL_Demo\images 下的 STM3210B-EVAL_SPIFlash.dfu 文件。

STM32 内部 Flash 需要写入的目标代码是前面修改的应用代码而不是 USB IAP 代码。这里需要注意的是 IAR 编译后的文件格式可以是 S19、hex、bin，但 Dfs 烧写需要的文件是 dfu 格式，该格式可以通过安装的 dfu 文件下的 DFU File Manager 程序来转换得到。



8、上述的解释同样适用于 STM32F10x 系列在 User flash 区域的 IAP 代码操作流程，如 UART BootLoader。需要注意的是这里介绍的 UART IAP 与 STM32 本身固化在 System Flash 区域通过 UART 升级的 IAP 不是完全相同的，尽管实现的功能一样。

9、执行玩上述步骤以后，确认 STM3210EVB 的 Boot 跳线为 User Flash 启动。Reset 后你就可以看到在 STM3210EVB 上出现漂亮的 GUI 啦。

补充：

在 ST 官方提供的 USB IAP 例程中我们可以在 main.c 中找到：

```
if (DFU_Button_Read() != 0x00)
{
.....
}
```

通过该函数我们可以知道，USB IAP 是通过判断 PB.09 端口的高低电平实现地址跳转的，对于其它形式的 STM32 电路板，我们可以仅仅简单的修改一下这个判断方式，即可利用 ST 提供的现成代码简单快速的实现 USB IAP 下载功能。