USB MSD BootLoader User Manual

V1.1
Cedar
2015/12/03

USB MSD BOOTLOADER是什么

它是一个固件升级工具，让嵌入式系统升级变得极其简单

1: 插入电脑USB接口

2: 把升级固件拖到设备盘符

3: 升级完成

为什么设计这个BOOT LOADER

在电子产品开发过程中，为了满足市场需要，经常是先开发出一个简单可用的版本，然后逐步迭代升级，修复bug，并增强系统功能

一个稳定，简单，安全的升级方式，就变得非常重要

对于嵌入式系统来说，常见的升级方式为

1. 串口升级（私有协议或者X－Modem）
2. USB升级（DFU）
3. U盘升级（OTG）
4. 网络升级
5. 无线升级（OTA）

从技术来说，这几种升级方式大同小异，原理类似，都是一个Loader代理接收数据通道的数据，然后解密，烧录到FLASH中；但用户体验完全不同，拿串口升级来说，首先用户需要一个串口软件，然后对于没有硬件串口的PC来说，就需要一个USB转串口设备，对于不同PC平台，串口软件就不一样，这需要学习成本，过程繁琐；所以在一些需要用户自行升级远程设备的情况下，即便是通过电话指导，80%的用户仍然不知道怎么升级，导致失败

USB的DFU升级，也是类似的问题，它设计的初衷就是面向专业用户的，而不是小白！所以需要安装DFU软件，按照手册来一步步升级

OTA升级和网络升级，体验好些，可用做到无感升级，但不适合所有场景

而U盘升级，用户学习成本最低，U盘大家都知道，然后拷贝一个Bin文件进去，插入设备，重启设备，就完成升级了，非常简单。类似的变种，比如手机升级，是最先进的，直接将手机模拟成U盘，然后用户拷贝数据到手机，重启就好了，非常简单

在嵌入式系统中，还没这么方便的升级手段，虽然ARM的Mbed有一种类似的固件更新功能，但它是专门为调试器设计的，不能内嵌到用户MCU中

所以，我将手机升级的方案引入到嵌入式系统中，从而为大家提供一个实现稳定，安全，零学习成本的升级方案

经过一段时间的学习研究，有了这个USB MSD Bootloader

Tips：如果您在使用过程中，出现技术问题，请联系我（QQ：819280802）

功能特点

1. 只占用15K FLASH空间
2. 简单易用，直接拖拽文件进行固件升级，无需任何专业知识
3. 采用USB大容量设备类，不用安装任何驱动
4. 支持各种系统（Windows / Linux / Mac / Android）
5. 不用开发任何上位机，提高产品效率
6. 支持各种加密算法（AES256等），轻松安全升级
7. 自动识别Bin，Hex，自定义加密固件（后缀为sec[1]）文件
8. 支持MD5文件校验机制，保证固件升级的完整性
9. 显示设备升级状态信息
10. 支持长文件名升级
11. 多种措施保证系统健壮性，保证Bootloader不会被误擦除，保证APP合法性
12. 支持用户自定义加密算法和完整校验算法，极致安全

[1] 默认禁用SEC加密文件，但留有相应的API接口，用户可以在源码的基础上添加自己的加密，解密算法

系统原理

1. 系统开机上电后，Bootloader接管系统，初始化USB硬件，等待USB连接
2. Bootloader在启动后1秒内，检测USB是否连接PC：如果连接PC，则进入固件升级模式，执行第3步；超时则跳转第8步，尝试执行用户APP
3. Bootloader模拟成MSD设备，构建FAT16虚拟文件系统，U盘名为”Bootloader”，容量为100M，但具体实际可用空间，根据用户MCU来确定，建议不要复制除APP之外的无关文件
4. 当用户复制文件到U盘时，Bootloader会判断文件后缀和判断文件size，如果size大于实际的MCU可用FLASH或者文件后缀不合法，则进入错误状态，更新状态文件，重新枚举USB
5. 文件后缀和size通过检测后，Bootloader会截获PC发送文件数据流，并写入MCU 对应的Flash中
6. 如果写入过程中出错，则终止操作，擦除APP内容，进入错误状态，更新状态文件，重新枚举USB
7. 成功写入后，Bootloader更新状态文件，重新枚举USB，显示升级完成；但不会运行APP，只有拔掉USB后，再次重启，才会进入第8步，尝试运行APP
8. Bootloader检查APP固件的栈和入口函数合法性，只有通过检测后，才开始执行APP。检测判断条件是栈指针必须在RAM地址空间内，入口函数地址必须处于THUMB模式，并LSB为1
9. 停止USB设备，关掉所有的中断，执行APP，APP开始接管系统

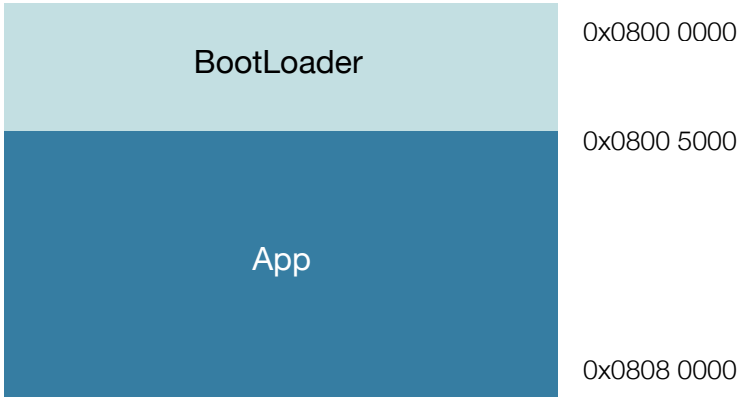
说明：

1: 操作状态和结果，会通过状态文件来显示，详细信息，请参考《状态文件》章节
2: Bootloader可以防止外部攻击，但用户APP不要随意的擦写Bootloader所在的内存区域，从而保证系统稳定

地址空间

以512K的STM32为例，BootLoader占用的Flash空间为前[2]，0x08000000-0x08004FFFF

APP使用余下的全部剩余FLASH，如下所示



具体Bootloader FLASH分配表，如下

芯片	BootLoader开始地址	BootLoader结束地址	APP开始地址
STM32F101/3/5/7	0x0800 0000	0x0800 4FFF	0x0800 5000

[2] Bootloader实际占用空间为15K，但为了便于拓展功能，所以保留20K给Bootloader

使用方法

开发者

1. 将Bootloader.bin程序烧录到芯片中
2. 将App工程的ROM地址修改为0x0800 5000
3. 确保启动向量放在ROM首地址
4. 编译工程，得到APP.bin文件

最终用户

1. 将设备插入PC，等待PC识别出名为”Bootloader”的U盘
2. 将升级固件复制到U盘中
3. 稍等1秒钟，文件名变成“SUCCESS.TXT”，升级成功

状态文件

Bootloader会模拟成一个U盘，里面包含一个名为XXX.txt的文件，XXX用于表示状态信息和操作结果

具体含义，如下表所示

文件名	含义	建议操作
READY.txt	Bootloader就绪，可以复制文件，升级系统	复制固件，开始升级
SUCCESS.txt	升级成功	拔出USB，重启设备
UNKOWN.txt	无法识别文件	重新选择合适的固件，Bootloader支持的后缀为.bin/.hex/.sec
LARGE.txt	固件内容过大，MCU Flash无法满足要求 注意：该错误只针对bin类型的文件有效	选择正确的固件，重新烧录
ERRFLASH.txt	FLASH烧写过程中出错	建议检查FLASH是否被锁定
ERRAPP.txt	无效的APP，栈地址或者入口地址错误	选择正确的固件，重新烧录
ERRKEY.txt	无效的加密文件，无法通过验证	选择正确的加密文件
NOAPP.txt	没有检测到APP，无法运行	重新升级，烧录APP固件

安全性

安全问题非常重要，固件被盗取，会造成极大的经济损失。Bootloader在设计的时候，充分考虑了这个因素，能够保证最大程度的保护APP固件

主要包括

1. 数据加密
   可以采用AES－256加密机制，来加密固件数据部分，保证数据层的安全
2. 高级别的权限机制
   对于sec加密类文件，只有授权用户，才具有升级文件的权利
3. MD5完整性校验
   对于内存比较大的MCU应用，可以设置MCU的FLASH BANK区域，用于放置接收到的固件，然后进行MD5校验，只有校验通过的，才进行解密和更新
4. 读保护
   外界无法通过USB MSD设备类读出任何FLASH内容，保证了绝对的安全
5. APP合法性检测
   通过检测APP固件的栈和入口地址，从而保证执行合法的APP

支持芯片列表

Vender	Chip	Available
ST	STM32 F101xx	Yes*
ST	STM32 F102xx	Yes*
ST	STM32 F103xx	Yes
ST	STM32 F105xx	Yes*
ST	STM32 F107xx	Yes*

注意：理论上支持STM32F101/2/3/5/7带USB接口的MCU，但仅在F103上面做过实际测试

资源占用

Vender	Chip	FLASH
ST	STM32 F101/3/5/7	15K

注意事项

1. Bootloader的安全性非常重要，强烈建议根据自己的应用场景，添加自己的加密key和包头校验机制
2. APP应用级别代码，不要随意擦写Bootloader区域的数据，从而增强系统安全性
3. 芯片在烧写HEX固件时，会首先擦除芯片内所有的APP FLASH空间，如果您希望在升级APP时，保留某些数据FLASH部分，请使用BIN形式的固件来升级，或者调整对应源码来进行修改；建议将数据部分放在APP Flash最后一个扇区
4. APP需要在main函数之前重定向中断向量表，参考代码如下所示

```
void APP_main(void)
{
    NVIC_SetVectorTable(NVIC_VectTab_FLASH, FLASH_START_ADDR-NVIC_VectTab_FLASH);

    delay_init();
    LED_Init();

    while(1)
    {
        LED1=0;
        delay_ms(20);
        LED1=1;
        delay_ms(980);
    }
}
```

更新记录

版本号	修改记录	修改人	时间
V1.0	First Init	cedar	2015-11-28
V1.1	增加地址示意图，修改文字错误	cedar	2015-12-03

联系我们

邮箱：master@iot-studio.com

网站：www.iot-studio.com

QQ技术交流群：386294792