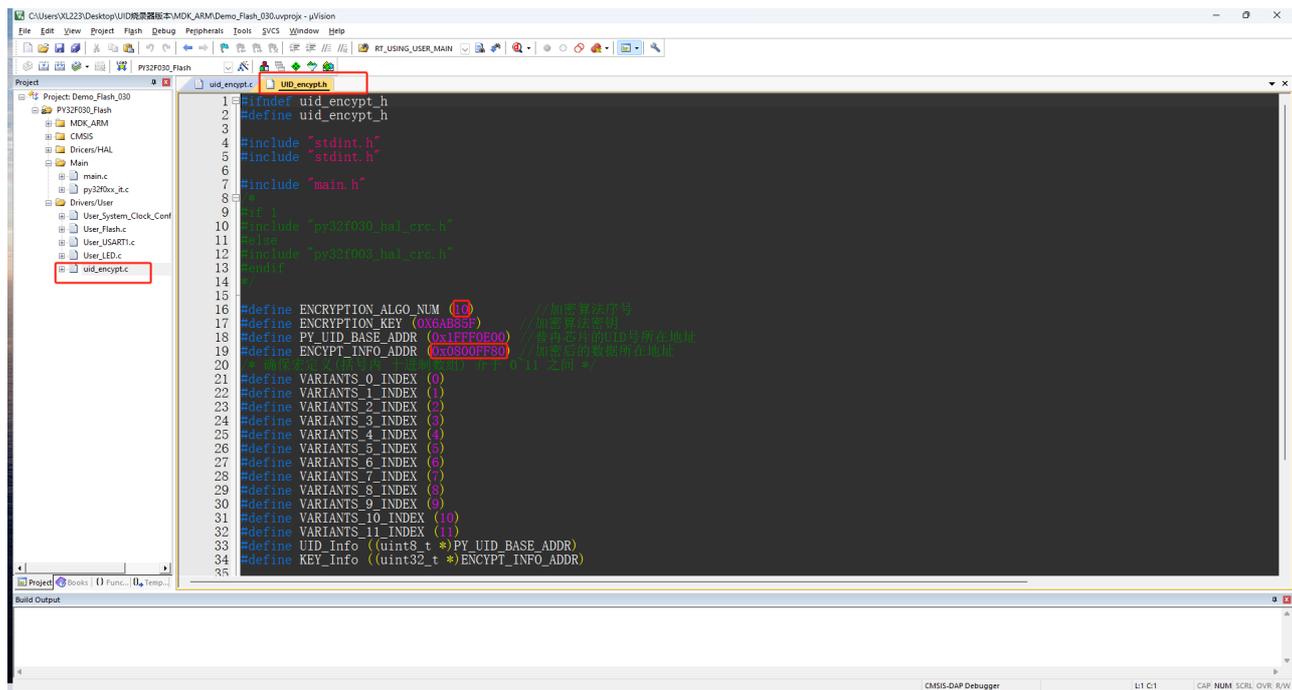


核心文件 大部分不能改动 只能改小部分!!!!!!



只能改圈出来的地方

第一个是加密的算法 可以挑一种

第二个是MCU的Flash地址

重点是第二个 要看使用的MCU Flash空间大小是多少 核心 就是这个地址不能被程序覆盖 要在程序占用的空间的后面 图上示例的是PY32F030 64K的最后128字节的空间

```
7
8 #include "User_Flash.h"
9
10 #include "UID_encrypt.h"
11
12
13 int main(void)
14 {
15
16     HAL_Init();
17     System_Clock_Config_HSI(); // 配置系统时钟为100M, 40MHz
18     LED_Init();
19
20     if(!Get_check_result())
21     {
22         while(1);
23     }
24
25
26
27
28
29     while(1)
30     {
31         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
32         HAL_Delay(300);
33         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
34         HAL_Delay(300);
35
36
37     }
38 }
39
40
41
```

主函数直接 这样子复制过去 就行 拷贝出来的固件 会校验不通过 一直卡在这里

验证方法很简单 用离线烧录器烧一个芯片 然后 用J_Flash去把它的固件读取出来 然后再把读取出来的固件烧写进另外一个芯片 效果就是 不会运行程序

怎么配合离线烧录器使用呢???

芯片选择

PY32F030 X8

设备设置

串口号

COM8 刷新

波特率

1000000 断开

- 限定烧写次数
- 烧写写保护
- 烧写配置项
- 按扇区擦除
- 烧写SDK设置
- 烧写UID加密
- 烧写滚码

烧录速度

高速 中速 低速

烧录次数

预设烧写次数 无限制

电源输出

- 烧录机台电压_5V
- 烧录机台电压_3.3V

电源输出

- 不输出[兼容被烧录端供电电压]
- 输出3.3V[仅烧录阶段供电]
- 输出3.3V[持续供电]

固件加载

C:\Users\XL223\Desktop\Demo.hex 浏览 重载

固件信息: Demo.hex

固件大小: 4800(4.69K) 字节 固件和校验: 0x57C90

填充: 0xFF

START: 0x08000000 END: 0x080012BF

查看烧录配置信息

芯片烧录选项

FLASH_OPTR

- RDP: 0xAA: level 0, 不使能读保护
- nBOOT1: 1: 选择System Memory作为启动区
- NRST_MODE: 0: 仅做复位输入
- WWDG_SW: 1: 软件看门狗
- IWDG_SW: 1: 软件看门狗
- BOR_LEV[[2:0]: 111: BOR上升阈值为3.2V, 下降阈值为3.1V
- BOR_EN: 0: BOR不使能

FLASH_SDKR

SDK_START: 0x800 F800
SDK_END: 0x800 07FF

FLASH_WRR

FLASH_WRR

FLASH_ERASE

Sector_Erase

UID密钥加载

加载加密文件 浏览 重载

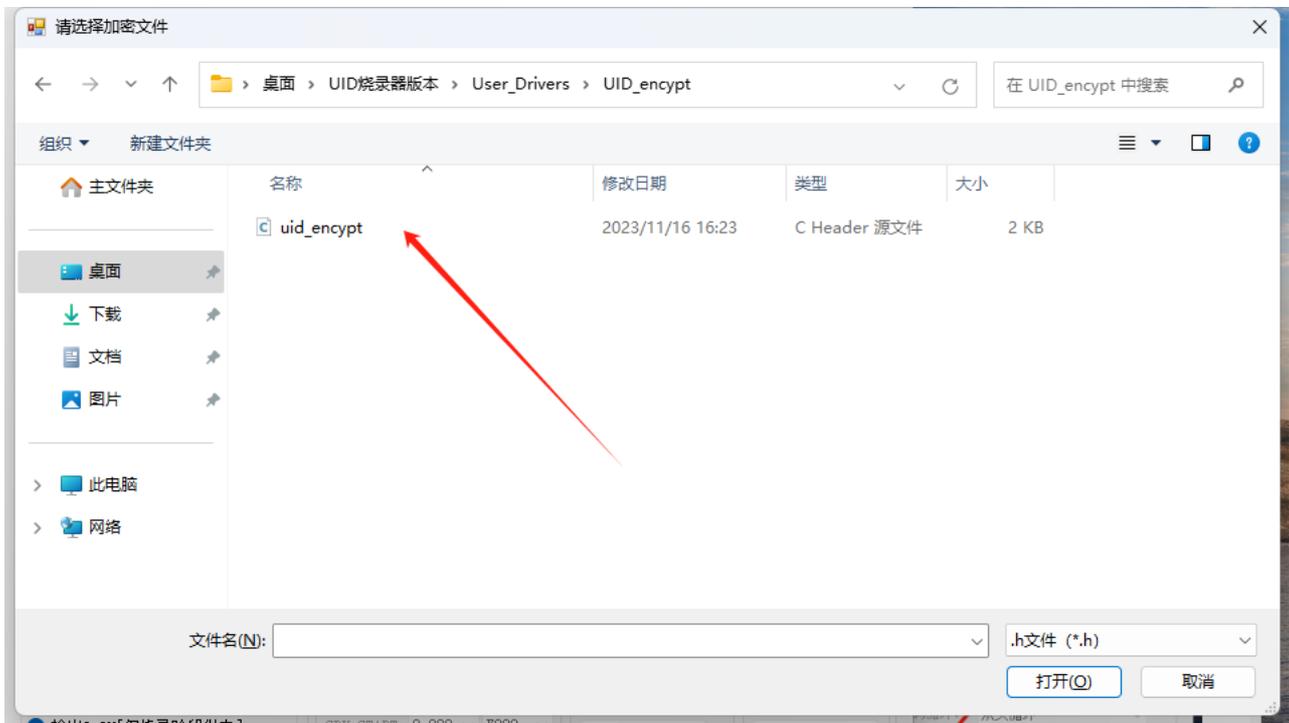
滚码烧录

滚码地址: 800 FF80 滚码步进(DEC): 1 滚码循环: 从头循环
滚码开始(HEX): 00000000 当前滚码(HEX): 00000000 滚码结束(HEX): FFFFFFFF 滚码长度: 1(Word)

信息打印

USB-SERIAL CH340 (COM8)
下位机信息: Wireless:4248F486:2307271702:2307291134
下位机信息: unlimited

载机



选择工程里面那个头文件



看信息是不是和头文件的一样

然后下载就行