第十四天笔记

1. 内部Flash概述
2. 存储器的概念

存储器指的是若干个存储单元的集合，每个存储单元都可以存储若干个二进制数，为了方便的操作存储单元，就为每个存储单元都分配了地址，就可以通过寻址来访问存储单元。由于计算机的处理的数据量较大，并且运算速度都很快，就要求存储器的容量更大，并且存取数据的速度更快。

1. 存储器的分类

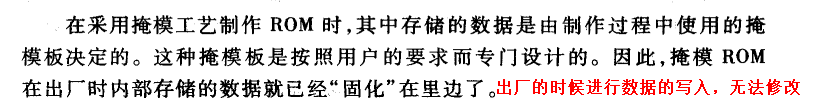
一般存储器按照功能划分可以分为两种： 只读存储器（ROM） 随机存储器（RAM）

* 只读存储器

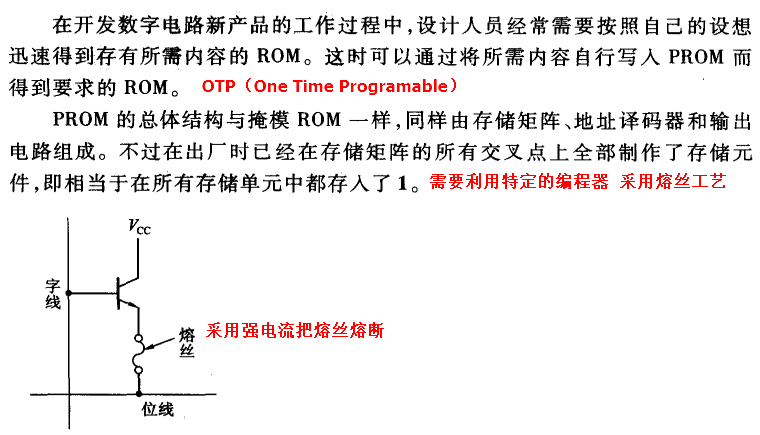
只读存储器也被称为ROM（Read Only Memory），在正常工作时只可以读取数据，不能随意修改ROM中的数据。为了提供工作效率以及降低成本，所以只读存储器也经历几个阶段的发展：ROM ---> PROM--->EPROM--->EEPROM--->Flash

**优点：结构简单，并且掉电不丢失 缺点：只适用于固定数据的存储**

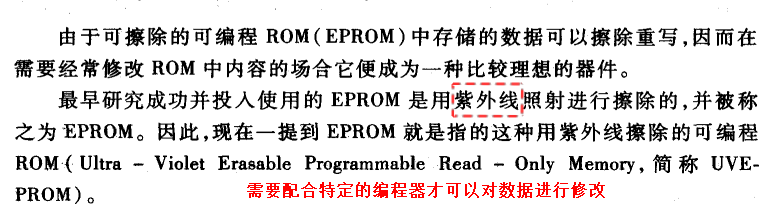
1. 只读存储器



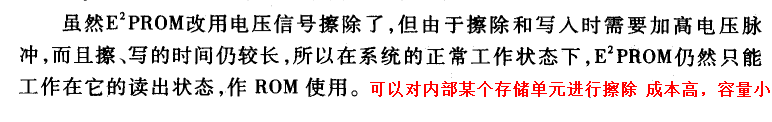
1. 可编程只读存储器



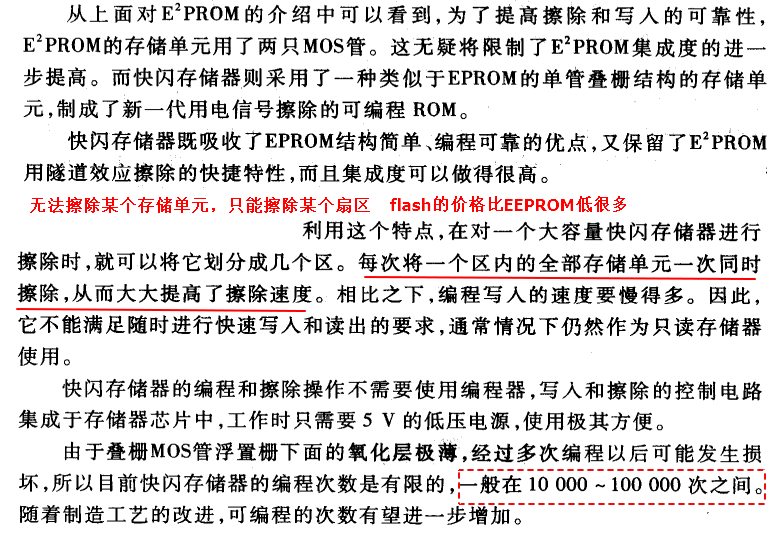
1. 紫外线可擦除只读存储器



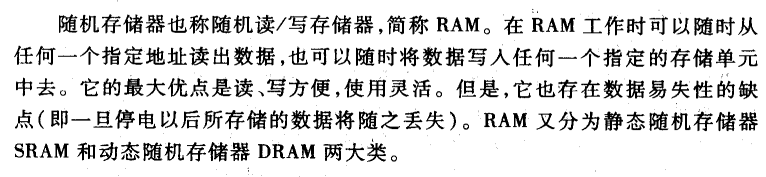
1. 电可擦除只读存储器



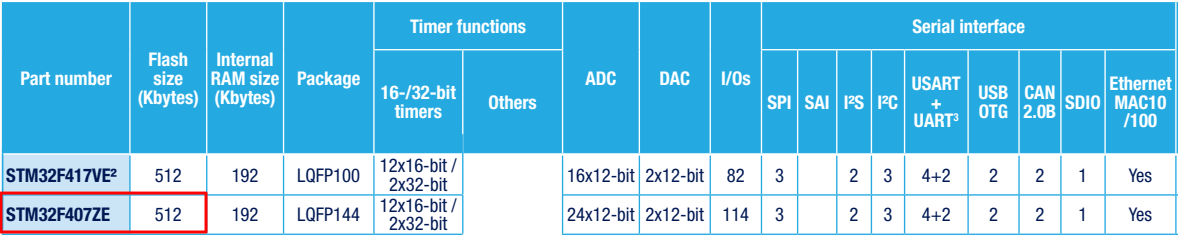
1. 闪存存储器



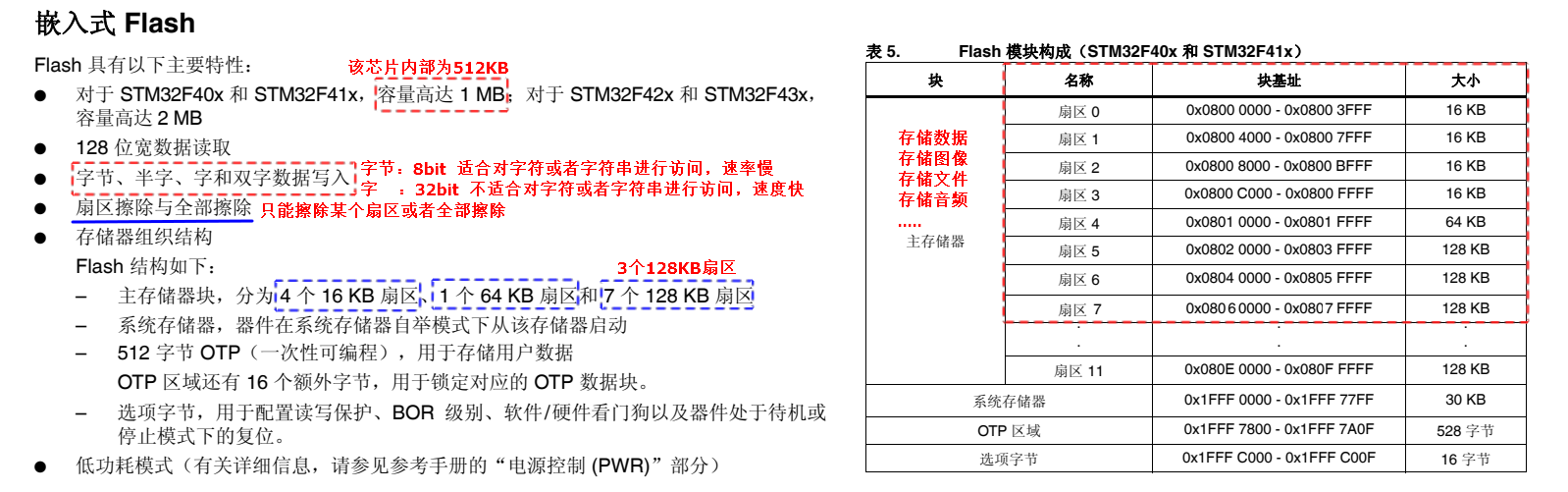
* 随机存储器



1. 内部Flash使用

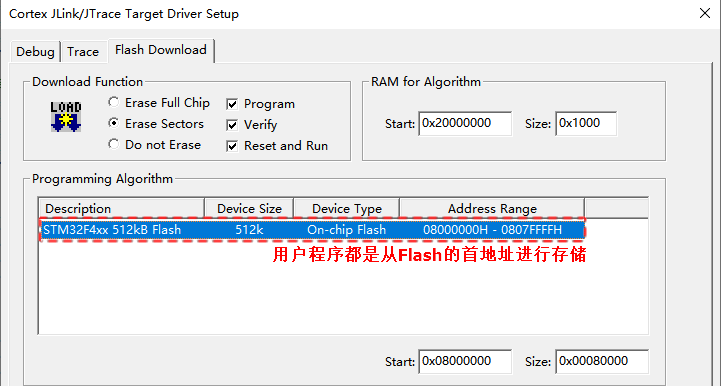


STM32F407芯片内部的Flash容量具有**512KB**，必须要了解内部flash内存分布，如下图所示



需要注意：该芯片只有**3个128KB的扇区**可以使用，对扇区进行访问或者擦除时，需要计算出扇区的起始地址以及结束地址。

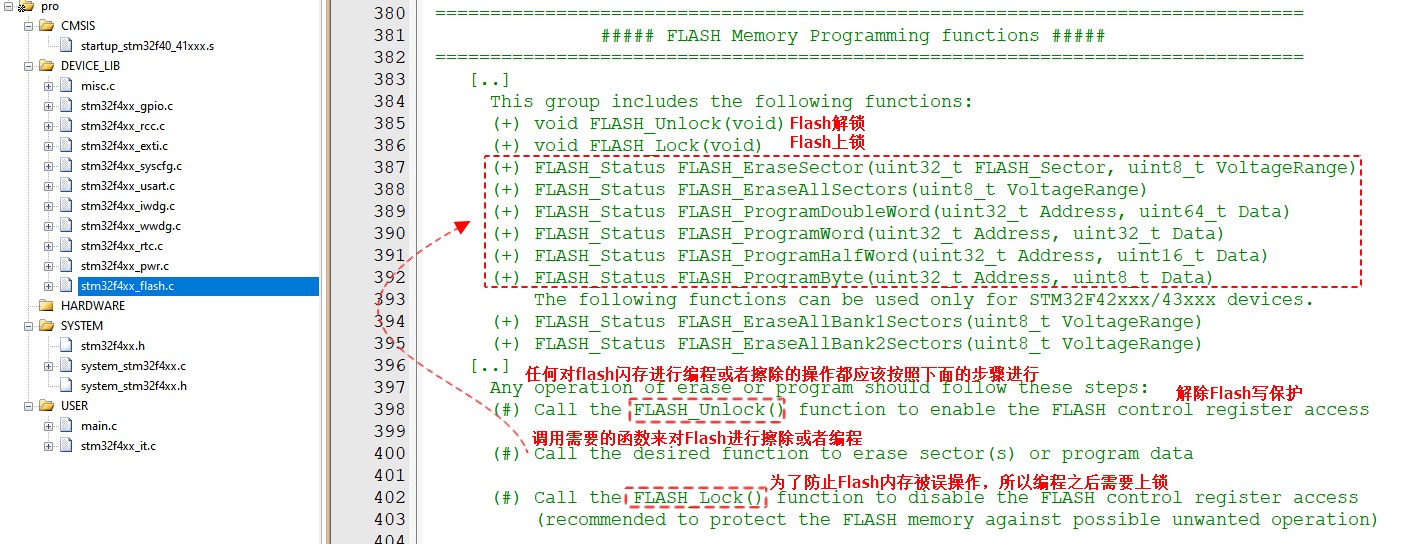
另外，用户程序一般都是下载到Flash闪存空间中的，**一般从Flash闪存的扇区0开始存储**。



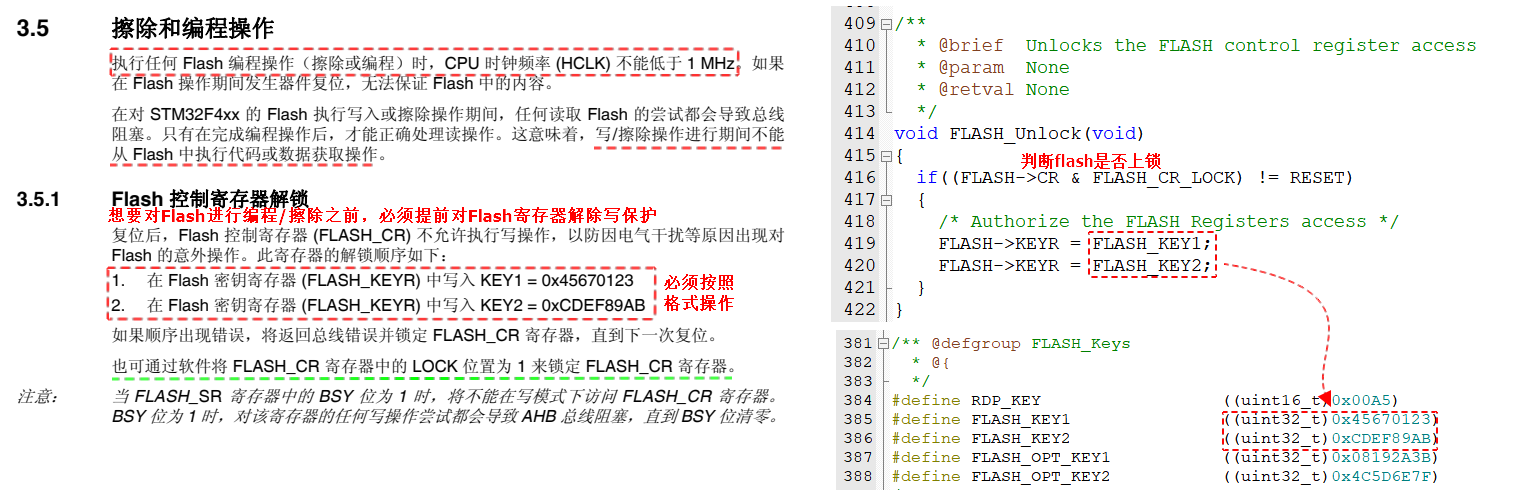
注意，用户在存储自定义数据的时候，尽量先从Flash闪存的后面的扇区进行操作，避免对用户程序造成影响。

1. 内部Flash的流程

如果用户打算对Flash空间进行编程或者擦除，对应的流程可以参考帮助手册以及文件注释

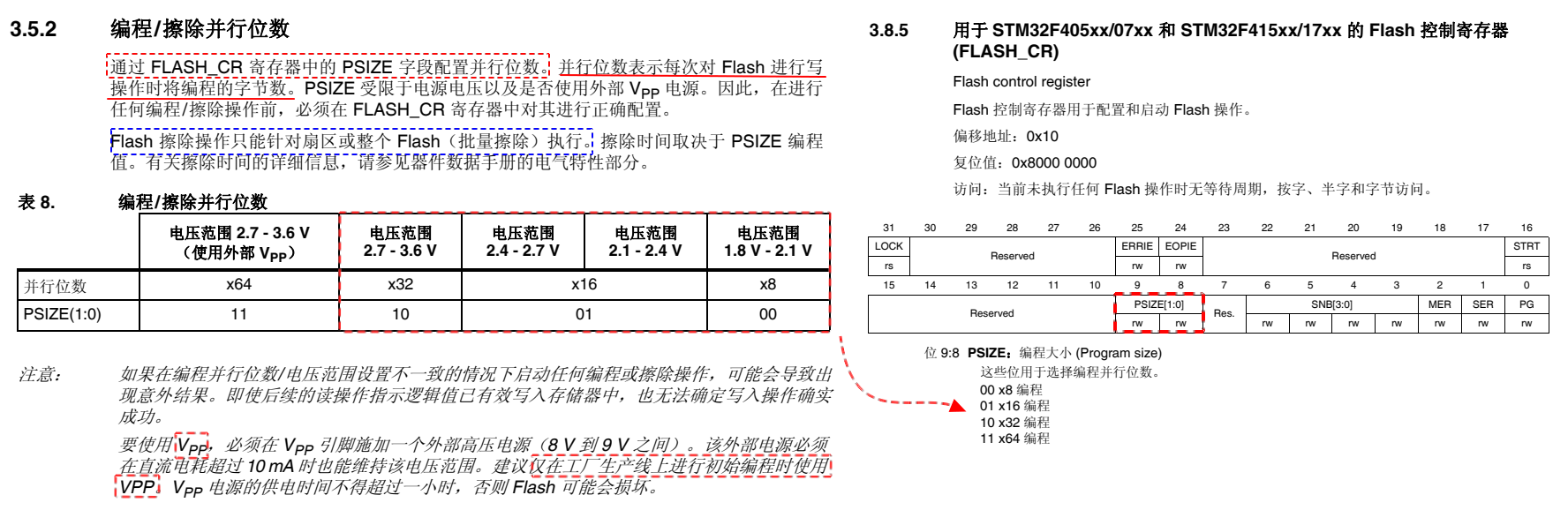


* Flash解锁

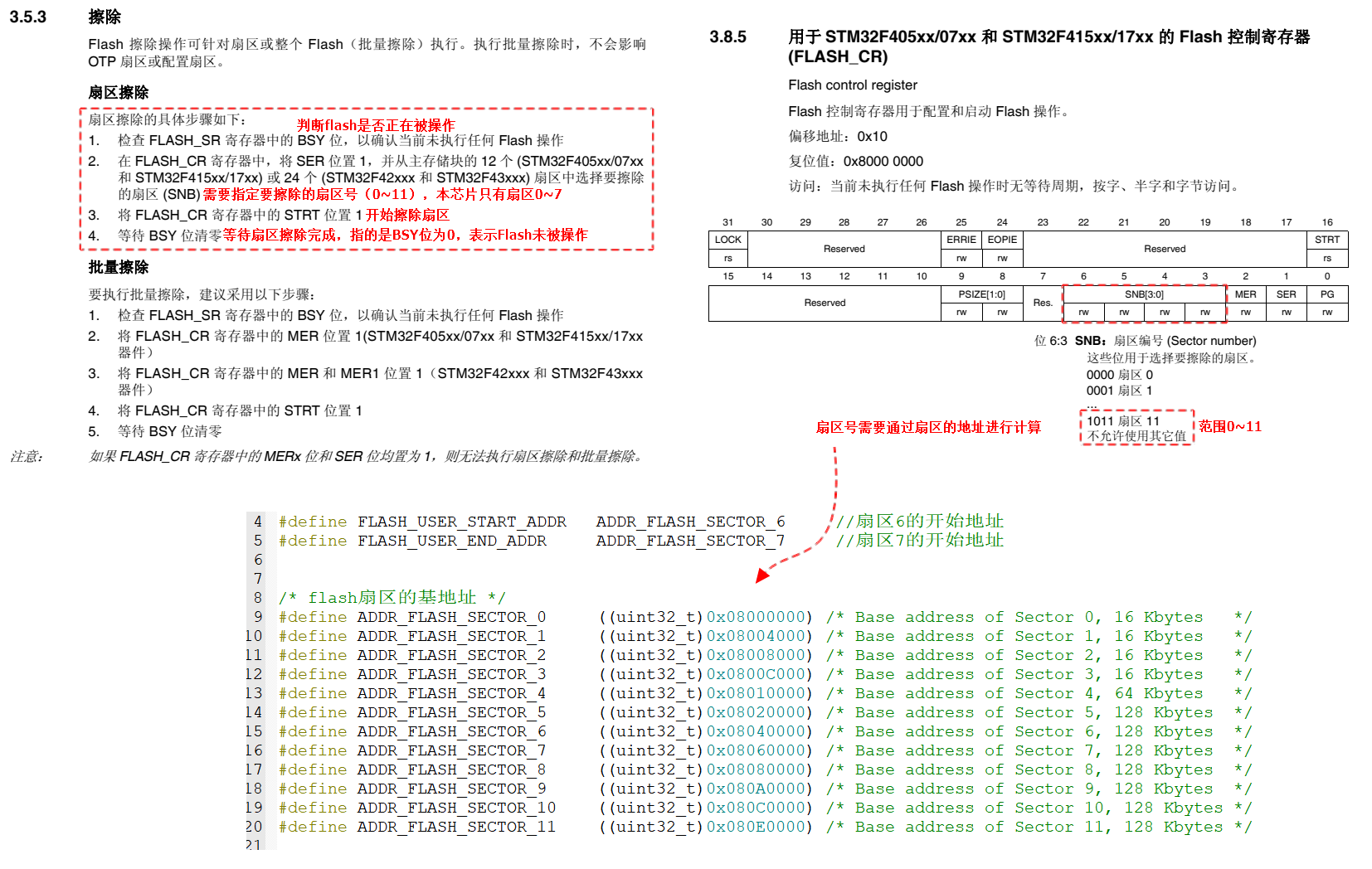


* Flash擦除

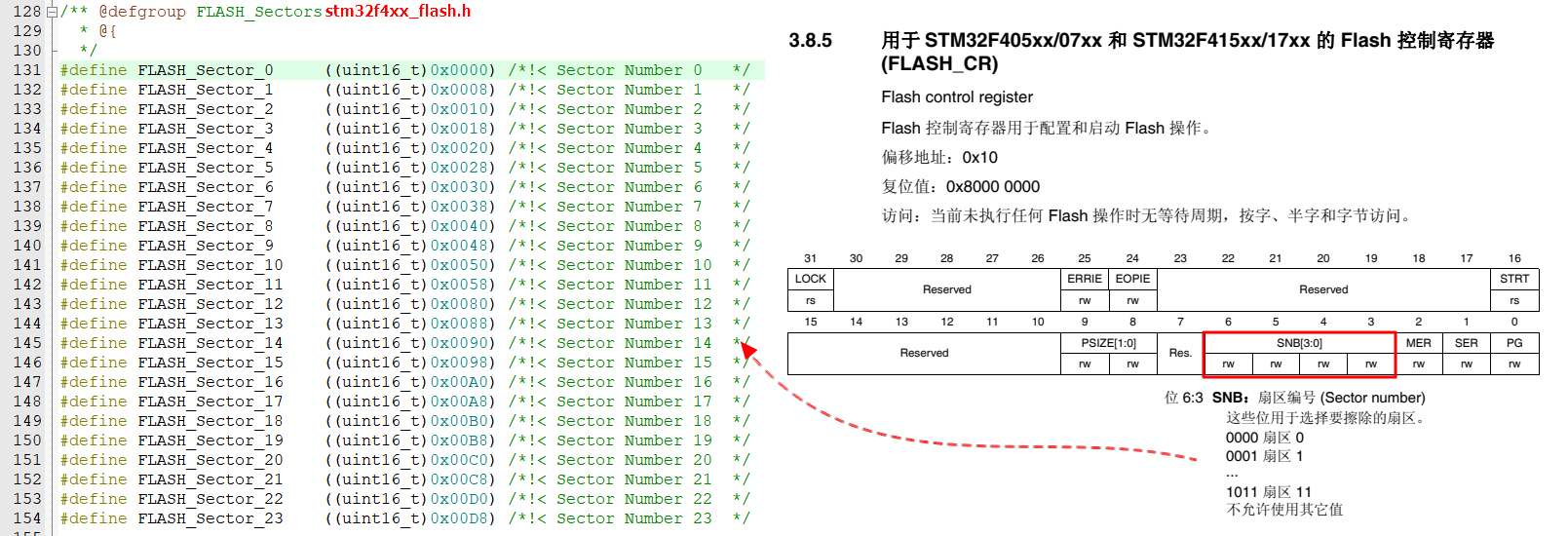
需要注意，在擦除扇区之前，需要设置编程/擦除的并行位数（字节、字、半字、双字），决定了擦除的时间。



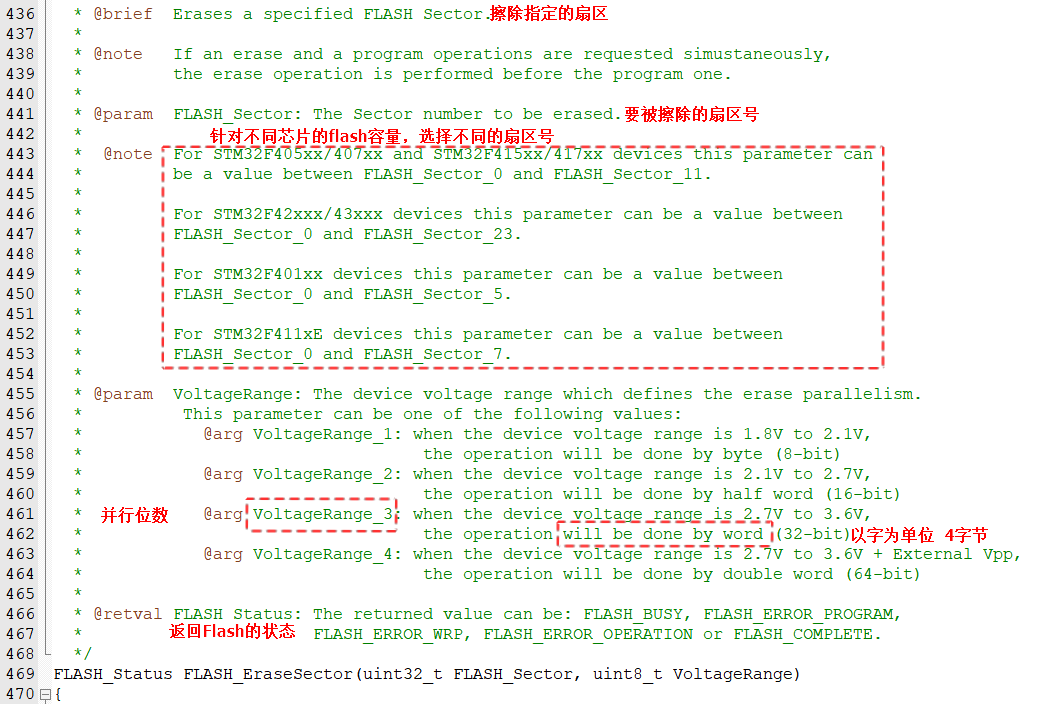
然后，在进行扇区擦除的时候需要指定扇区号（扇区0~扇区7），扇区号可以通过扇区的地址进行计算



**如果在擦除扇区的过程中打算跳转到下一个扇区，需要对扇区号进行计算和偏移（+8）**



打算对扇区进行擦除操作，可以调用函数库中提供的函数接口来选择擦除某个扇区或者擦除所有扇区，如下图所示：



函数原型

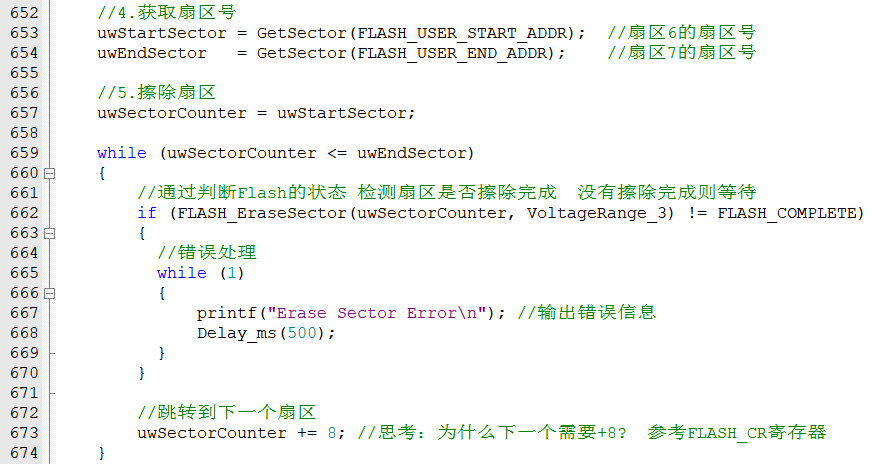
FLASH\_Status FLASH\_EraseSector(uint32\_t FLASH\_Sector, uint8\_t VoltageRange)

函数参数

参数一：FLASH\_Sector 需要擦除的扇区号 如 FLASH\_Sector\_0 ~ FLASH\_Sector\_11

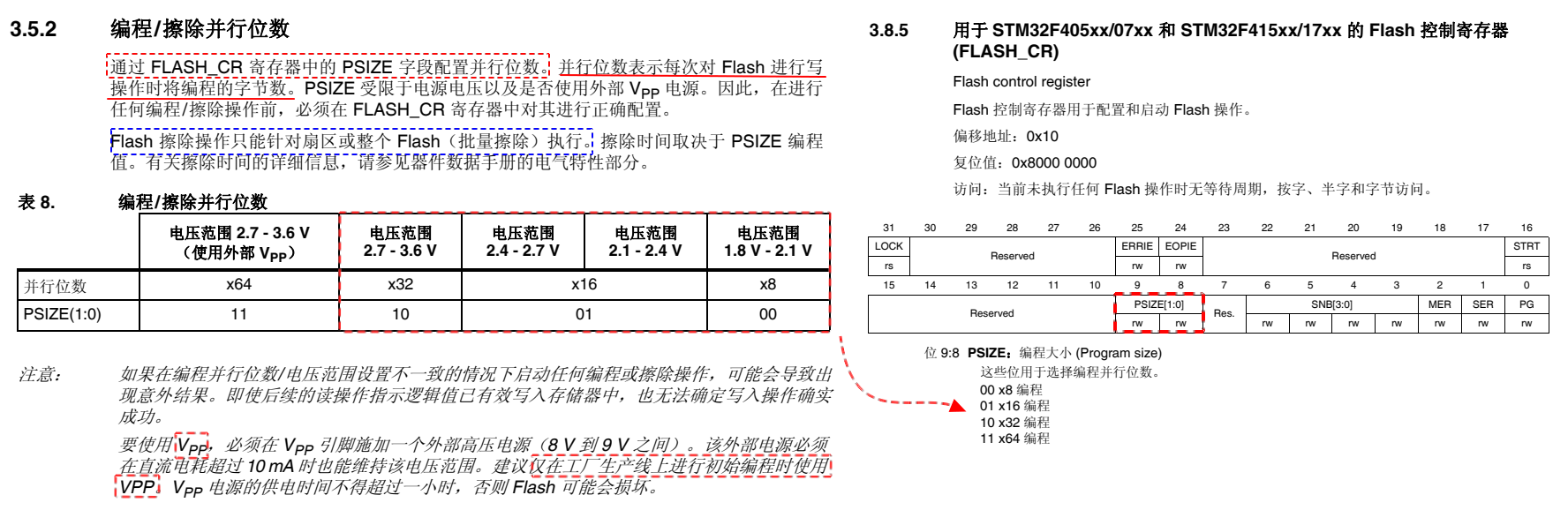
参数二：VoltageRange 需要擦除的并行位数 如 VoltageRange\_3 以字（32bit）为单位

返回值 返回lFlash的操作状态 如 FLASH\_COMPLETE 标志操作完成

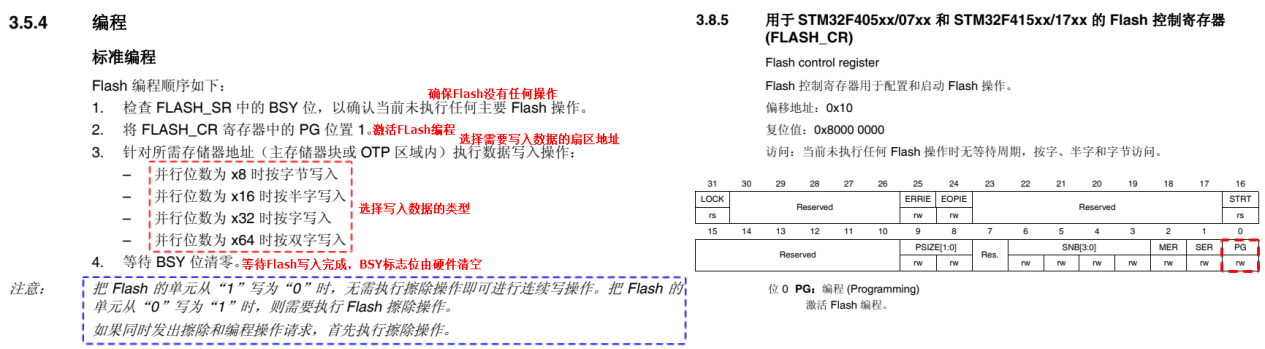


* Flash写入

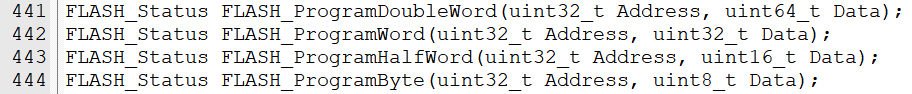
需要注意，在写入扇区之前，需要设置编程/擦除的并行位数（字节、字、半字、双字），决定了写入的时间。

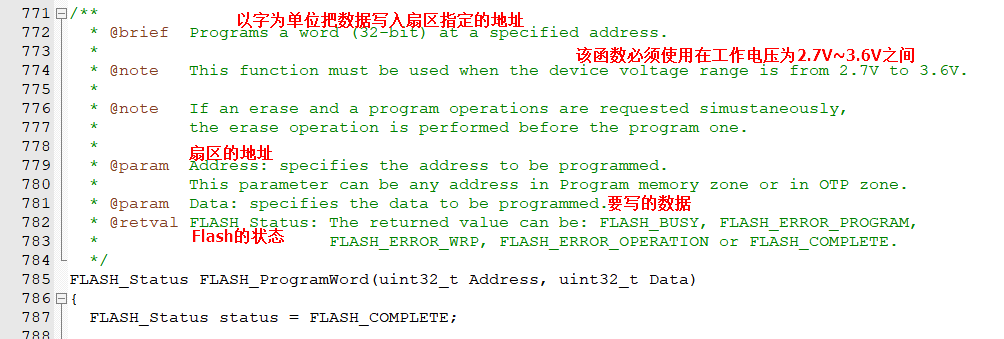


Flash闪存往某个扇区写入数据的操作流程如下



打算对扇区进行写入操作，可以调用函数库中提供的函数接口来选择把数据写入某个扇区





函数原型

FLASH\_Status FLASH\_ProgramWord(uint32\_t Address, uint32\_t Data)

函数参数

参数一： Address 指的是准备写入数据的扇区地址

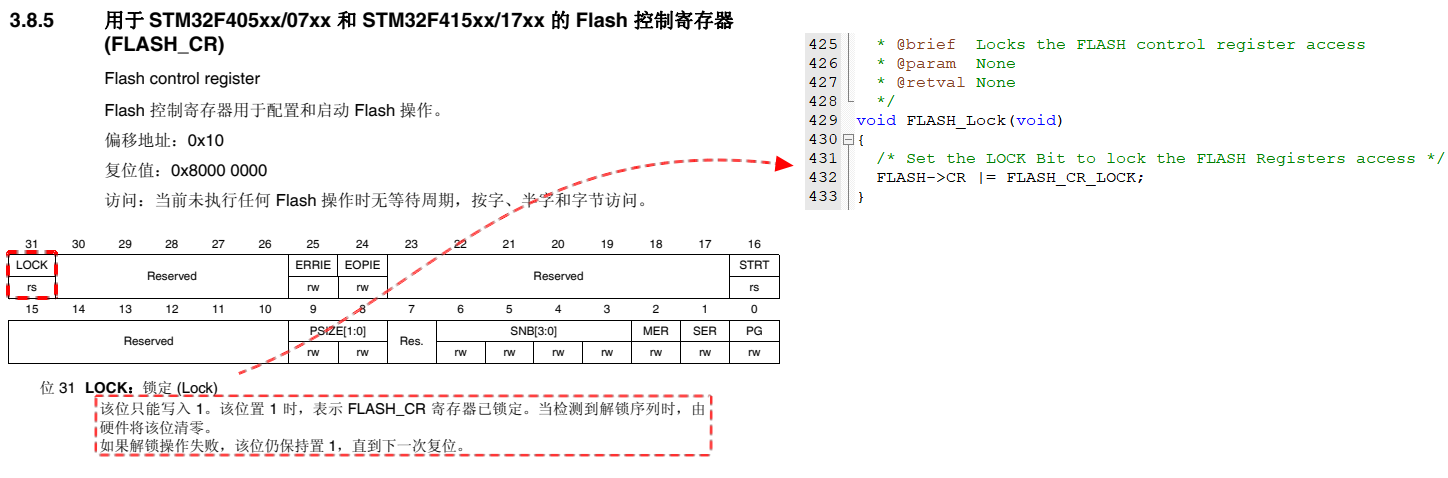
参数二： Data 指的是准备写入的数据内容

返回值 返回lFlash的操作状态 如 FLASH\_COMPLETE 标志操作完成



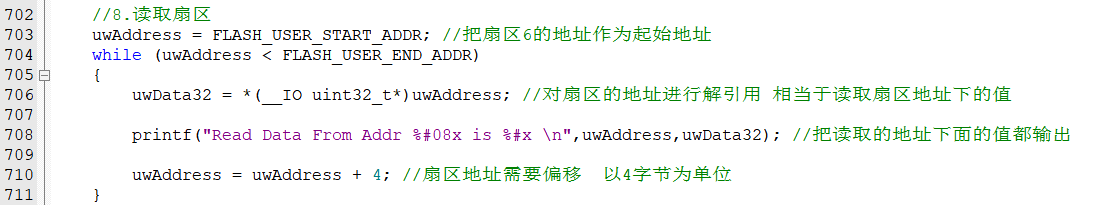
* Flash上锁

如果已经完成Flash扇区的数据写入，为了防止出现意外状况导致扇区数据出现丢失等问题，则需要重新开启Flash寄存器的写保护功能。



* Flash读取

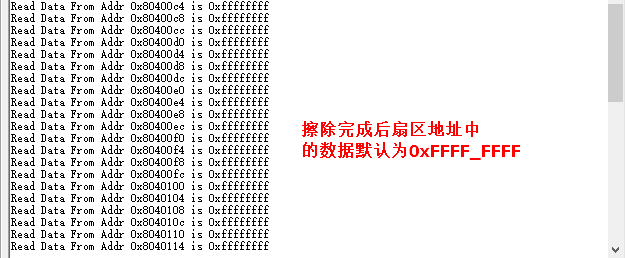
如果已经把数据写入到Flash扇区的某个地址下，则可以通过访问扇区对应的地址空间来读取对应的数据，这里就设计到C语言的地址访问，可以通过指针进行操作。操作如下图所示：



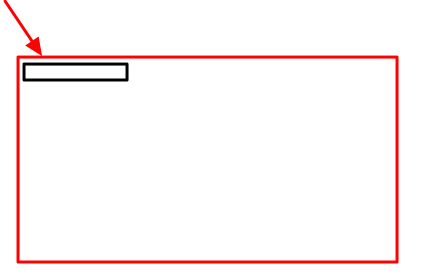
**注意：读取扇区地数据的时候扇区地址需要进行偏移，并且要访问符合扇区地址范围的空间。**



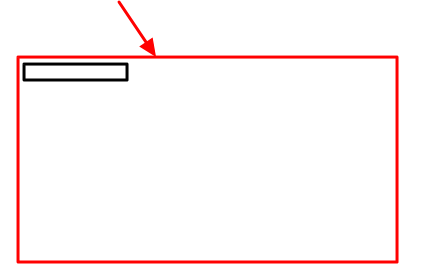
**思考：如果在对flash的某个扇区进行擦除之后，那这个扇区的地址下面的值默认是多少？**



**思考：如果在对flash的某个扇区进行数据写入时，如果扇区的起始地址中已经存在一些数 据，是否需要擦除扇区？？ 答案：需要先擦除扇区再重新写入数据**



**思考：如果在对flash的某个扇区进行数据写入时，如果扇区的起始地址中已经存在一些数 据，那把需要写入的数据写在已经存在的数据的地址之后，是否需要擦除扇区？？ 不需要**





练习：利用RTC时钟来获取当前的日期与时间，格式为”Date=2021-08-27-05,Time=09-32-30\n”

然后把获取的时间与日期写入到Flash空间中，要求可以记录100条数据，应该如何编写代码？ 思考：确保是100条？ 要求把缓存的条数一起输出,存够100条再输出

如 ：”[001]Date=2021-08-27-05,Time=09-32-30\n”