

STM32启动过程浅析

参考资料：

- STM32 MAP文件浅析.pdf
- STM32 启动文件浅析.pdf
- Cortex-M3权威指南(中文).pdf

1, MAP文件浅析（了解）

MDK编译过程文件（11种）.map 连接器生成的列表文件，对分析程序存储占用情况非常有用

MAP文件是MDK编译代码后，产生的集程序、数据及IO空间的一种映射列表文件
简单说就是包括了：各种.c文件、函数、符号等的地址、大小引用关系等信息
分析各.c文件占用FLASH 和 RAM的大小，方便优化代码

MAP文件组成

组成部分	简介
程序段交叉引用关系	描述各文件之间函数调用关系
删除映像未使用的程序段	描述工程中未用到而被删除的冗余程序段(函数/数据)
映像符号表	描述各符号（程序段/数据）在存储器中的地址、类型、大小等
映像内存分布图	描述各个程序段（函数）在存储器中的地址及占用大小
映像组件大小	给出整个映像代码（.o）占用空间汇总信息

MAP文件实操

学会分析：哪个.c占用flash 和ram比较大，以便针对性的优化

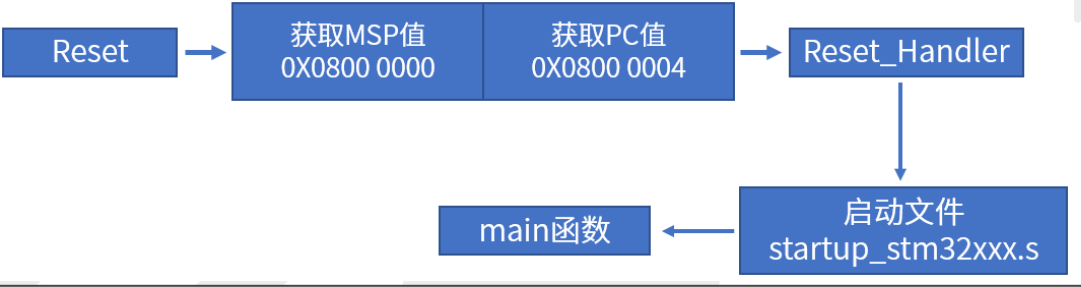
2.1, STM32启动模式（F1/F4/F7/H7）

M3/M4/M7等内核复位后，做的第一件事：
1, 从地址 0x0000 0000 处取出堆栈指针 MSP 的初始值，该值就是栈顶地址
2, 从地址 0x0000 0004 处取出程序计数器指针 PC 的初始值，该值是复位中断服务函数地址

芯片厂商可以把0x0000 0000和0x0000 0004地址映射到其它的地址！！

STM32启动模式（F1/F4/F7/H7）

在系统复位后，SYSCLK的第4个上升沿，BOOT引脚的值将被锁存
我们主要用从主闪存存储器启动的方式



内部FLASH启动为例

参考资料：STM32 启动文件浅析_V1.1.pdf

- 1, 初始化MSP 从0X0800 0000获取
- 2, 初始化PC 从0X0800 0004获取
- 3, 设置堆栈大小 Heap_Size(堆)、Stack_Size(栈)
- 4, 初始化中断向量表 __Vectors定义
- 5, 调用初始化函数 可选的，如调用：SystemInit函数
- 6, 调用__main 标准C库函数，执行一系列设置，最终调用main函数

Reset_Handler函数介绍

```
Reset_Handler PROC
EXPORT Reset_Handler [WEAK]
IMPORT __main
IMPORT SystemInit
LDR R0,=SystemInit
BLX R0
LDR R0,=__main
BX R0
ENDP
```

EXPORT：标明全局属性，可被外部调用
IMPORT：申明来自外部文件，类extern
PROC：定义子程序
ENDP：表示子程序结束
WEAK：弱定义

如果外部没有：SystemInit函数
则会报错！！

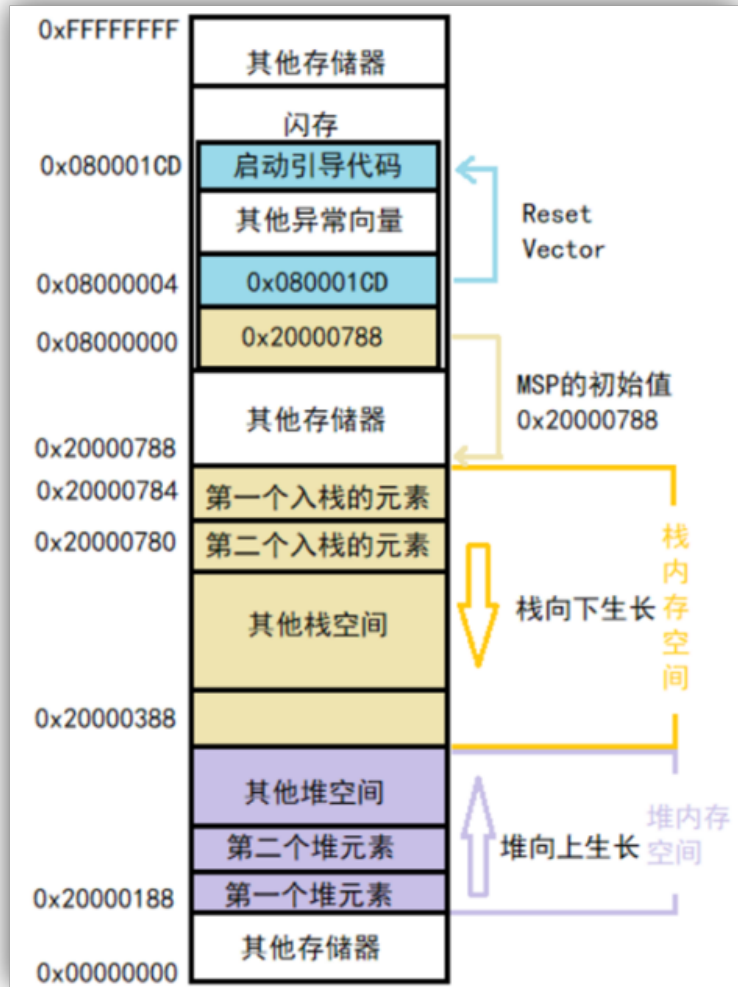
内存	作用
栈(Stack)	编译器自动分配和释放，存放函数参数、局部变量等
堆(Heap)	程序员分配和释放，如malloc、calloc、realloc等

堆栈简介

函数局部变量较多，嵌套关系复杂时，需加大栈大小（Stack_Size）！

2.2, STM32启动过程

STM32启动过程图解



3, 课堂总结（了解）