目录

[STM32笔记 2](#_Toc150452656)

[一、程序的编译 2](#_Toc150452657)

[1、预编译 2](#_Toc150452658)

[2、编译 2](#_Toc150452659)

[3、链接 2](#_Toc150452660)

[4、Elf文件 2](#_Toc150452661)

[5、Hex和bin文件 2](#_Toc150452662)

[二、编译器相关 2](#_Toc150452663)

[1、关键字 2](#_Toc150452664)

[2、变量的定义 2](#_Toc150452665)

[三、芯片的启动过程 2](#_Toc150452666)

[1、上电流程 2](#_Toc150452667)

[2、中断向量表 2](#_Toc150452668)

[1、什么是中断向量表 2](#_Toc150452669)

[2、向量表偏移量寄存器(VTOR)（地址：0xE000\_ED08） 3](#_Toc150452670)

[3、中断发生的过程 3](#_Toc150452671)

[4、Cotex-M0升级 3](#_Toc150452672)

[5、参考资料 3](#_Toc150452673)

[四、Hex文件的分析 3](#_Toc150452674)

# 

# STM32笔记

## 一、程序的编译

### 1、预编译

使用-E选项

### 2、编译

### 3、链接

### 4、Elf文件

### 5、Hex和bin文件

## 二、编译器相关

### 1、关键字

### 2、变量的定义

将变量存放到指定的内存位置

Gcc：

## 三、芯片的启动过程

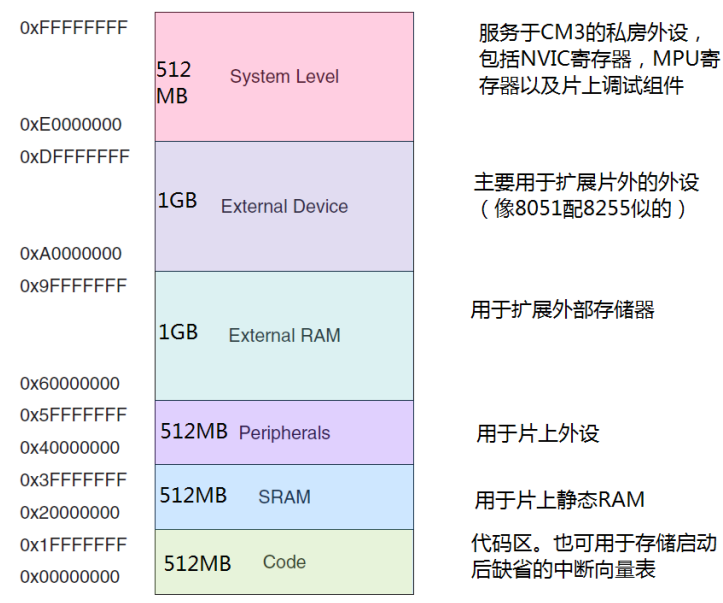
### 1、上电流程

芯片上电复位，程序从0x0000 0000地址开始执行。根据boot脚的不同状态，0x0000 0000地址上的内容来源不同。有三部分来源:

a.用户代码区 0x0800 0000

b.RAM区 0x2000 0000

c.系统引导区 0x



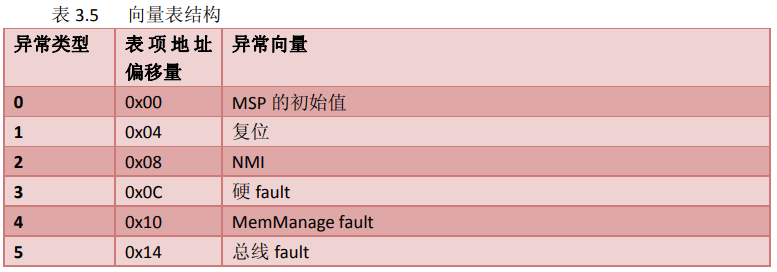
虽然来源不同，但是内容都是一样的，必须是“中断向量表”。1和2内容都是由用户决定。3的内容由芯片厂家确定，出厂就会固化。1、2号区域的内容是程序在下载阶段就写进去的。

芯片上电之后的第一件事情就是从0x0000 0000地址取出栈顶指针，然后从0x0000 0004地址取出指令(复位中断)赋值给PC指针，从而程序开始正常运行。

### 2、中断向量表

#### 1、什么是中断向量表

当一个发生的异常被 CM3 内核接受，对应的异常 handler 就会执行。为了决定 handler 的入 口地址，CM3 使用了“向量表查表机制”。这里使用一张向量表。向量表其实是一个 WORD （32 位整数）数组，每个下标对应一种异常，该下标元素的值则是该异常 handler 的入口地 址。向量表的存储位置是可以设置的，通过 NVIC 中的一个重定位寄存器来指出向量表的地址。在复位后，该寄存器的值为 0。因此，在地址 0 处必须包含一张向量表，用于初始时的异常分配。



(CORTEX-M3权威指南 第三章、第七章)

#### 2、向量表偏移量寄存器(VTOR)（地址：0xE000\_ED08）

该寄存器的作用是为了动态重分发中断，CM3 允许向量表重定位。为了实现这个功能，NVIC 中有一个寄存器，称为“向量表偏移量寄存器”（在地址 0xE000\_ED08 处），通过修改它的值就能定位向量表。

#### 3、中断发生的过程

当中的一旦发生，需要执行三个操作:1、保存现场2、根据中断号查表找中断入口位置3、更新一些寄存器

针对第二步，内核的在什么地方去获取向量表？内核从向量偏移寄存器VTOR读取向量。注意:M0内核没有该寄存器，所以内核会默认从0x0000 0000地址读取中断向量表。

#### 4、Cotex-M0升级

由于M0么有向量偏移寄存器，所有在boot程序中需要1、将中断向量表复制到0x2000 0000地址2、设置系统寄存器，将RAM地址重映射到0x0000 0000地址上。3、跳转到APP程序。(代码可以尝试放在APP代码中，未验证)

memcpy((void\*)0x20000000, (void\*)0x08004000, VECTOR\_SIZE);

SYSCFG\_MemoryRemapConfig(SYSCFG\_MemoryRemap\_SRAM);//SYSCFG->CFGR1 |= 0x03;

#### 5、参考资料

1、[STM32F0的IAP升级](https://blog.csdn.net/weixin_44788542/article/details/114373763)

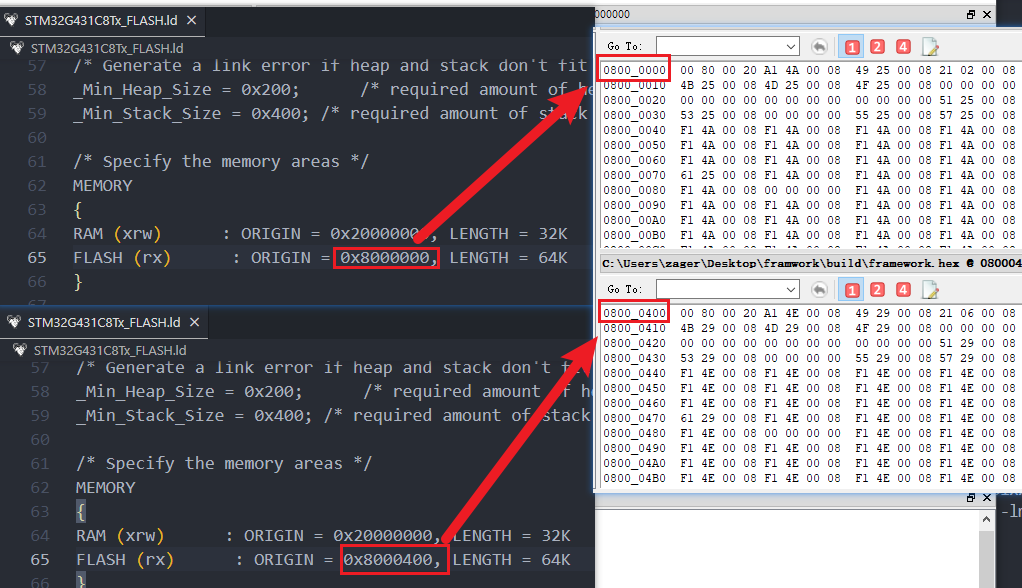
2、CORTEX-M3权威指南

## 四、Hex文件的分析

对于一般的hex文件，通过j-flash软件可以很清楚的看到，在0x0800 0000地址上存放的是栈顶指针0x2000 8000(8000是因为这款mcu的RAM是32K的)。然后依次是各个中断的入口地址(0x800 20xx该地址重复出现，该现象的原因是有些没有用到的中断用该中断来代替【可能是编译造成，也可能是其他原因】)。

Hex文件中携带的0x0800 0000地址如何而来？

该地址用户在.ld文件中指定而来。可以做个实验，修改链接文件中的0x800 0000地址，编译之后，重新打开hex文件可以清楚的看到起始地址被修改了。



### 全局变量和hex文件的关系

a.定义一个全局变量：

unsigned int init\_val\_no\_zero = 0xA5FEEF5A;

b.查看.map文件，获取其地址

c.使用J-Flash软件，打开对应的hex文件，查看0xA5FEEF5A其位置

d.分析hex文件的末尾

需要结合.dis文件来看

e.分析.ld文件

五、链接文件.ld