零死角玩转STM32—M4系列

启动文件讲解

淘宝：fire-stm32.taobao.com

论坛： www.firebbs.cn

扫描进入淘宝店铺

主讲内容

01

启动文件讲解

参考资料:《零死角玩转STM32》

“启动文件详解”章节

启动文件的讲解 — 开始

1-注释的讲解

2-程序的讲解

3-如何查找资料（ARM的汇编指令）

启动文件讲解

启动文件的作用

1-初始化堆栈指针SP

2-初始化PC指针，指向复位程序

3-初始化中断向量表

4-配置系统时钟

5-调用C库函数_main，最终进入C的世界

启动文件讲解

汇编程序如何注释

1-汇编注释用“；”

2-C语言注释用“//”或者“/**/”

启动文件讲解

1-Stack—栈

用于局部变量、函数调用、函数形参的开销

```
Stack_Size      EQU     0x00000400

                AREA    STACK, NOINIT, READWRITE, ALIGN=3
Stack_Mem       SPACE   Stack_Size
__initial_sp
```

启动文件讲解

EQU：宏定义的伪指令，相当于等于，类似与 C 中的 define

AREA：告诉汇编器汇编一个新的代码段或者数据段

SPACE ：用于分配一定大小的内存空间，单位为字节

标号__initial_sp 紧挨着 SPACE 语句放置，表示栈的结束地址，即栈顶地址，栈是由高向低生长的。

启动文件讲解

2-Heap—堆

堆用于动态内存的分配，malloc函数

```
Heap_Size       EQU     0x00000200

                AREA    HEAP, NOINIT, READWRITE, ALIGN=3
__heap_base
Heap_Mem        SPACE   Heap_Size
__heap_limit
```

启动文件讲解

PRESERVE8： 指定当前文件的堆栈按照 8 字节对齐

THUMB： 表示后面指令为 THUMB 指令。 THUBM 是 ARM 以前的指令集， 16bit，现在 Cortex-M 系列的都使用 THUMB-2 指令集， THUMB-2 是 32 位的，兼容 16 位和 32 位的指令，是 THUMB 的超级。

启动文件讲解

EXPORT： 声明一个标号具有全局属性，可被外部的文件使用。如果是 IAR 编译器，则使用的是 GLOBAL 这个指令。

DCD：分配一个或者多个以字为单位的内存，以四字节对齐，并要求初始化这些内存。在向量表中， DCD 分配了一堆内存，并且以 ESR 的入口地址初始化它们。

启动文件讲解

3-向量表

1-向量表实际上是一个32位的整型数组，一个元素对应一个异常（ESR），数组元素存的就是ESR的入口地址。

2-向量表在复位后从FLASH的0地址开始，具体的初始化值请查询参考手册的中断章节。

启动文件讲解

```
 1 __Vectors   DCD    __initial_sp              ;栈顶地址
 2            DCD    Reset_Handler             ;复位程序地址
 3            DCD    NMI_Handler
 4            DCD    HardFault_Handler
 5            DCD    MemManage_Handler
 6            DCD    BusFault_Handler
 7            DCD    UsageFault_Handler
 8            DCD    0                          ; 0 表示保留
 9            DCD    0
10            DCD    0
11            DCD    0
12            DCD    SVC_Handler
13            DCD    DebugMon_Handler
14            DCD    0
15            DCD    PendSV_Handler
16            DCD    SysTick_Handler
17
18
19 ;外部中断开始
20            DCD    WWDG_IRQHandler
21            DCD    PVD_IRQHandler
22            DCD    TAMP_STAMP_IRQHandler
23
24 ;限于篇幅，中间代码省略
25            DCD    LTDC_IRQHandler
26            DCD    LTDC_ER_IRQHandler
27            DCD    DMA2D_IRQHandler
28 __Vectors_End

1 __Vectors_Size EQU __Vectors_End - __Vectors
```

从代码上看，向量表中存放的都是中断服务函数的函数名，可我们知道 C 语言中的函数名就是一个地址。

启动文件讲解

FLASH地址是从0X0800 0000开始，向量表又是从0地址开始存放，这是否矛盾？？

启动文件讲解

4-复位程序

1-复位程序是上电后单片机执行的第一个程序

2-调用SystemInit函数配置系统时钟；调用C库函数_main，并最终进入C的世界

```
Reset_Handler    PROC
                 EXPORT  Reset_Handler             [WEAK]
        IMPORT  SystemInit
        IMPORT  __main

                 LDR     R0, =SystemInit
                 BLX     R0
                 LDR     R0, =__main
                 BX      R0
                 ENDP
```

启动文件讲解

WEAK：表示弱定义，如果外部文件优先定义了该标号则首先引用该标号，如果外部文件没有声明也不会出错。这里表示复位子程序可以由用户在其他文件重新实现，这里并不是唯一的。

IMPORT：表示该标号来自外部文件，跟 C 语言中的 EXTERN 关键字类似。这里表示 SystemInit 和__main 这两个函数均来自外部的文件。

启动文件讲解

Cortex内核的指令

LDR、BLX、BX 是 CM4 内核的指令，可在《CM3 权威指南 CnR2》第四章-指令集里面查询到，具体作用见下表：

指令名称	作用
LDR	从存储器中加载字到一个寄存器中
BL	跳转到由寄存器/标号给出的地址，并把跳转前的下条指令地址保存到 LR
BLX	跳转到由寄存器给出的地址，并根据寄存器的 LSE 确定处理器的状态，还要把跳转前的下条指令地址保存到 LR
BX	跳转到由寄存器/标号给出的地址，不用返回

LDR作为伪指令

LDR：加载一个立即数或者一个地址值到一个寄存器

举例：

LDR Rd, = label

如果label是立即数，那Rd等于立即数

如果label是一个标识符，比如指针，那存到Rd的就是label这个标识符的地址

启动文件讲解

5-中断服务程序

1-启动文件为我们写好了全部的中断服务程序，函数的名称必须与向量表里面初始化的名称一样。

2-这些程序都是空的，需要我们在C文件里面重新实现。如果我们写的中断服务程序的函数名写错了，程序也不会报错，而是会进入一个死循环。

```
SysTick_Handler PROC
                EXPORT  SysTick_Handler                 [WEAK]
                B       .
                ENDP
```

启动文件讲解

6-用户堆栈初始化

由标准的C库函数_main来完成。

IF,ELSE,ENDIF： 汇编的条件分支语句，跟 C 语言的 if ,else 类似。

END：文件结束。

ALIGN：对指令或者数据存放的地址进行对齐，后面会跟一个立即数。缺省表示 4 字节对齐。

ARM汇编指令讲解

启动文件里面涉及到的ARM指令

指令名称	作用
EQU	给数字常量取一个符号名，相当于 C 语言中的 define
AREA	汇编一个新的代码段或者数据段
SPACE	分配内存空间
PRESERVE8	当前文件堆栈需按照 8 字节对齐
EXPORT	声明一个标号具有全局属性，可被外部的文件使用
DCD	以字为单位分配内存，要求 4 字节对齐，并要求初始化这些内存
PROC	定义子程序，与 ENDP 成对使用，表示子程序结束
WEAK	弱定义，如果外部文件声明了一个标号，则优先使用外部文件定义的标号，如果外部文件没有定义也不出错。要注意的是：这个不是 ARM 的指令，是编译器的，这里放在一起只是为了方便。
IMPORT	声明标号来自外部文件，跟 C 语言中的 EXTERN 关键字类似
B	跳转到一个标号
ALIGN	编译器对指令或者数据的存放地址进行对齐，一般需要跟一个立即数，缺省表示 4 字节对齐。要注意的是：这个不是 ARM 的指令，是编译器的，这里放在一起只是为了方便。
END	到达文件的末尾，文件结束

零死角玩转STM32—M4系列

THANKS

论坛： www.firebbs.cn

淘宝：fire-stm32.taobao.com

扫描进入淘宝店铺