

第一讲 泾渭分明

泾河水清，渭河水浑，泾河的水流入渭河时，清浊不混。比喻界限清楚或是非分明。同样，学习 STM32 的过程也如同大河汇聚一样，无数的支流汇聚成大江，聚少成多，积沙成塔。

本讲学习目标：

- 1、理清架构，把握重心。
- 2、审时度势，看清自我。
- 3、适时查找，择优吸收。

A: 写在前面的话

STM32 是 Cortex-M3 系列的，而 Cortex-M3 处理器采用 ARMv7-M 架构。如果不知道，请你记住吧，否则乱说出去让人笑话的。而 M3 的 M，正是指代的 ARM 中的 M，可想而知，M3 只是比较弱的 ARM 一种，在国外并不高端。

为何笔者说的如此廉价的片子，用的好的却是少部分的工程师？为什么买了开发平台却不知如何去有效的学习？究竟错在了哪里使得你止步不前？

原因只有一点：你还没有适应！没有适应 STM32 芯片的编程背景，没有适应大量文档的有效阅读，没有适应 STM32 的开发方式！多数的工程师 51 学的很溜，于是乎闲暇时候转型 STM32。有人说：单片机无非就是配置寄存器。可是正是这句无非，使众多工程师止步不前了。

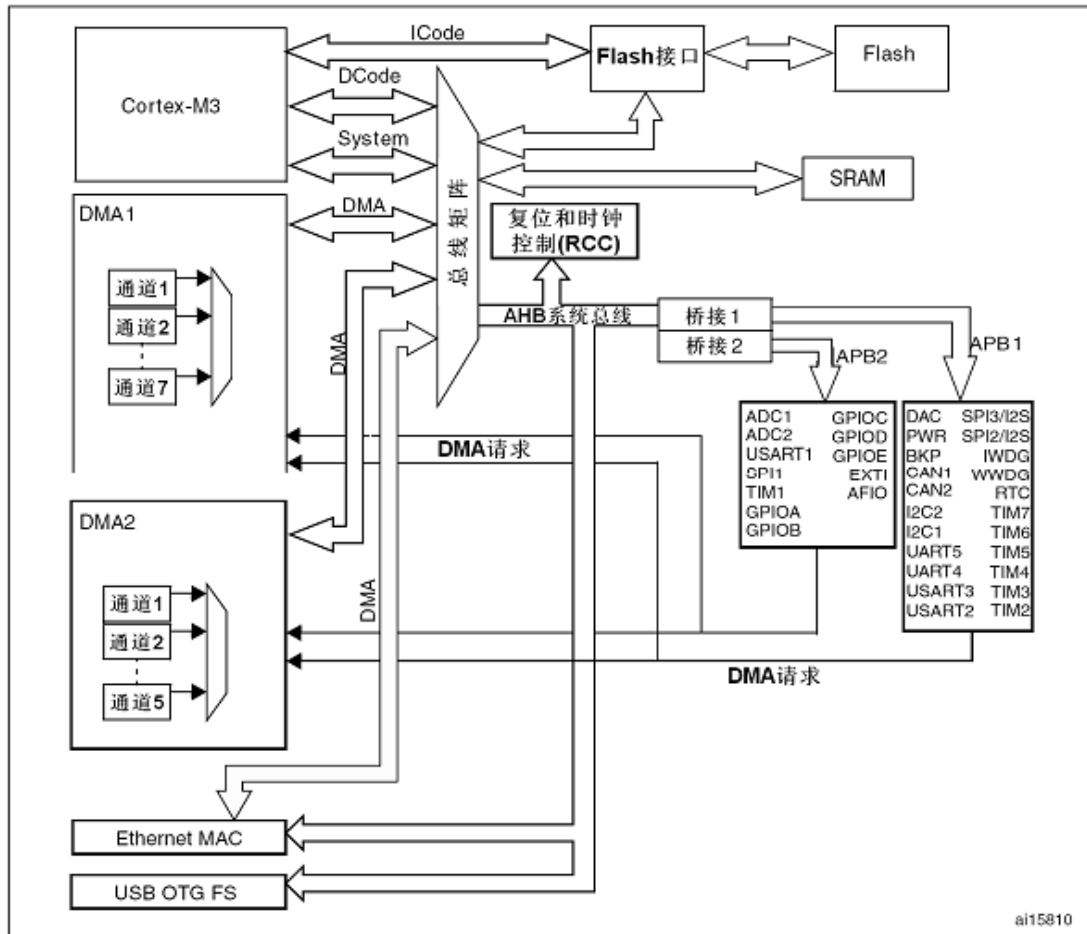
笔者在 10 年 8-9 月之间，入手了 STM32，阅读了 N 多开发板资料，闭门造车看了一周的代码，小有体会，加以实践，果然恍然大悟。玩过一遍之后，就作为技术储备扔下了，并没有太大在意。

E-mail: Poseidonstorm@126.com or 471661781@qq.com

直至近日，为人答疑解惑之时，创立本坛，为避免重复劳动，所以开始着手整理这本“稀里糊涂学 STM32”教程。

B: 揭秘 STM32

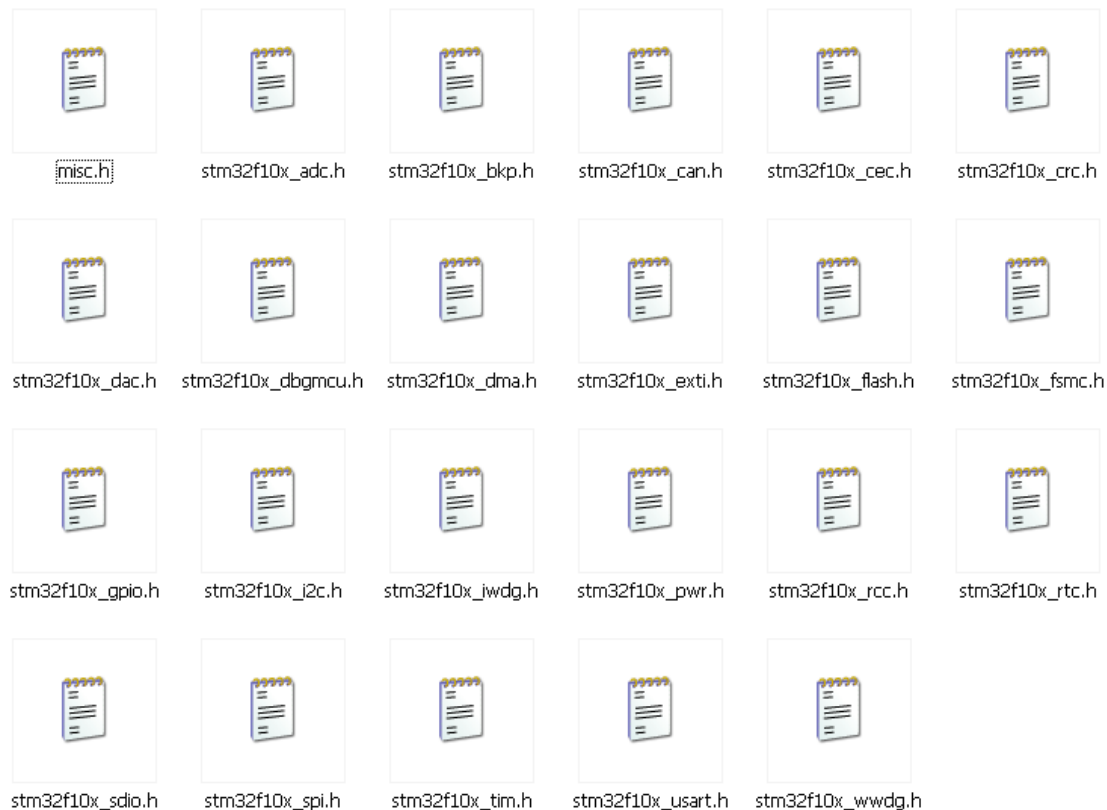
STM32 结构远比 51 复杂，所以有着众多的寄存器网络，整体的芯片架构如下：



DMA 或许很多朋友没有学到，我暂时不想先做介绍，但是希望关注的是 AHB 总线这是贯穿所有外设的一条总线，可以在图中发现，AHB 经过桥接，由 APB1、APB2 控制着几乎所有外设，希望记住一点，2>1，如何理解呢？APB2 属于高速设备，而 APB1 则属于低速设备。

上面的一些介绍，是为了让大家对 STM32 的硬件结构有所理解，如需更加详细的内容可以去参看 26 页 (RM0008 STM32F10xxx 参考手册 中文 V10)。

接下来想说明下 STM32 的软件底层，同样是结构很是复杂，以前玩 51 的时候有些工程师一个 Reg52.h 或者我这种懒人 AT89X52.h 用了很是开心。同样道理，STM32 也有一套完整的，含有致命与知名缺陷的，完整外设的“.h”见下图：



（“STM32F10x_StdPeriph_Lib_V3.4.0”中的 inc）

到这里是一个很重要的节点，常听人说，STM32 有两种软件开发方式，用库和不用库，这时候你该清楚的选择自己要走的路了，什么用库做开发，库在哪里？这点确实值得细细道明。在设计 STM32 时，官方已经发现其架构的复杂，所以请了几个倒霉工程师，做了一些库函数，包括诸如声明、定义等等，开始的几版漏洞百出，直到 2.0-3.0 的这一过渡，库的质量有了明显的改观，并且逐步有了自己的套路。

说下用库开发的优缺点吧，大致是这几点：

- 1、开发效率上高，集成库环境优美~适用于急于求成~

- 2、不用看繁琐的手册，一本固件库函数指南搞定~
- 3、执行及编译效率不高，速度慢，不一定是最优化的代码。
- 4、有手有脚都能做。

（最新的库和相关的 APPLICATION NOTE & DATASHEET 可以在 ST 官网上下载，具体地址是 <http://www.st.com/internet/mcu/class/1734.jsp>）

准备工作：

- 1、下载 V3.4 库文件，找到 `stm32f10x_stdperiph_lib_um.chm` 文件
 - 2、下载中文版 V2.0 固件库手册，因为有可能上面的库文件手册有些 E 文看不懂
 - 3、下载 STM32F10X 参考手册中文版，最近的版本是 V10
 - 4、下载编译环境 KEIL MDK，笔者使用的是 MDK 4.1
 - 5、有条件的可以准备一款开发板，笔者不是卖开发板的，所以任何一块都行，没有统一规定。
 - 6、准备一个 Jlink 或者 ULink 2 仿真器，或者也可以使用 ISP 方式，那只需准备一个 USB 转串口。
 - 7、不要带板凳，因为屁股不能帮助你思考，只需要带脑子即可。
-

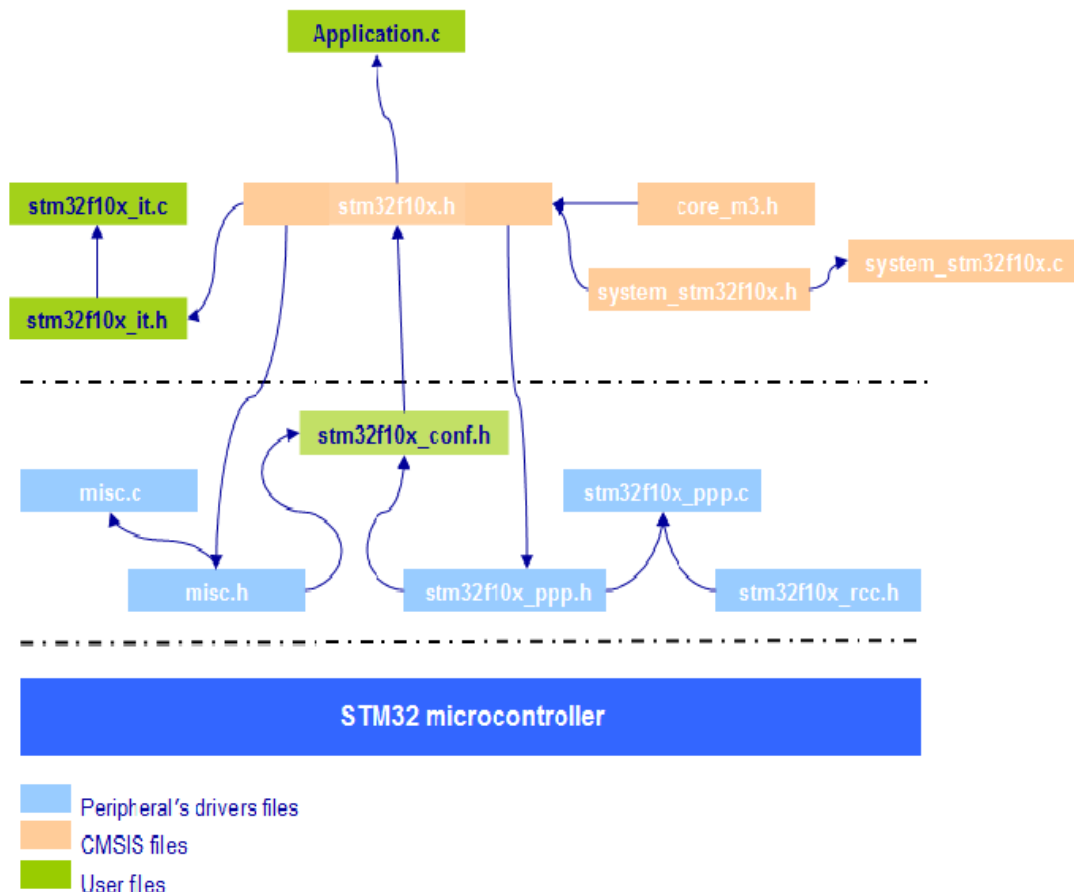
关于奖励：

笔者下学期将有学校实训课教学，故需要时间整理实训指导书，而本群之手册，将会是其蓝本，本群之糊涂课，将为试讲，若有不足还望见谅。凡能完成所有课程，并完成最后大作业真正理解 STM32 开发方式，完成学习心得及笔记者，在分享进步之后，将有机会借到 M3 内核芯片的小板一块，虽然是旧的，而且是借着玩，但是我希望获得的同学能有所应用。并人人相传。

E-mail: Poseidonstorm@126.com or 471661781@qq.com

用库，那么请你下面看这张图，接着听完我的课，要不想用库~哇哈哈，你可以回去看看好友正点原子的<STM32 不完全手册>及其配套的 ALIENTEK MiniSTM32 开发板。其实并不是一定要用库操作，但是初学者，我一直认为，还是需要循序渐进的过程，用库入门，用寄存器提高。

下面是库文件的结构图：



看了以上结构图，是否有所感想？深刻理解了这张图的组成，即可建立自己的工程，完成对应的工作。很多朋友在不同的手册上一步步跟着做，建立了第一个工程，点灯，但是不知其所以然，群内的三位管理员，皆经历过笔者痛骂，试想，如果没有别的技术文档，仅有 ST 的官方文档，您是否能如同 51 那样顺手的完成一个点灯工程呢？

应用层软件是依赖于第二层的定义以及第三层相关的库函数文件完成的，如果你不想使用库，那么你可以利用 stm32f10x.h 中的寄存器和结构体的定义直接对寄存器操作。而用库开发则是依靠第三层中已经定义好的一些函数直接进行函数调用。下面将会解释下几个关键的衔接头文件。

继续看结构图，stm32f10x.h 是如同 Reg52.H 一般重要的头文件，他定义了芯片的类型

E-mail: Poseidonstorm@126.com or 471661781@qq.com

及外设，并且在 0461 行处引入了另外三个头文件（而后会适当做一些介绍）

```
#include "core_cm3.h"           //ARM 公司为了标准化 M3 内核的一些声明
#include "system_stm32f10x.h"    //没啥东西的~
#include <stdint.h>              //编译器及内部参量类型定义
```

在引入头文件过后，定义了一些变量类型，和外设相关的结构体、外设常量。

请注意一下 8729 行及以下部分：

```
#include "stm32f10x_conf.h"      //很关键的哦！稍后会讲！

#define SET_BIT(REG, BIT)        ((REG) |= (BIT)) //一些 ARM 中的位操作定义
#define CLEAR_BIT(REG, BIT)      ((REG) &= ~(BIT))
#define READ_BIT(REG, BIT)       ((REG) & (BIT))
#define CLEAR_REG(REG)           ((REG) = (0x0))
#define WRITE_REG(REG, VAL)      ((REG) = (VAL))
#define READ_REG(REG)            ((REG))

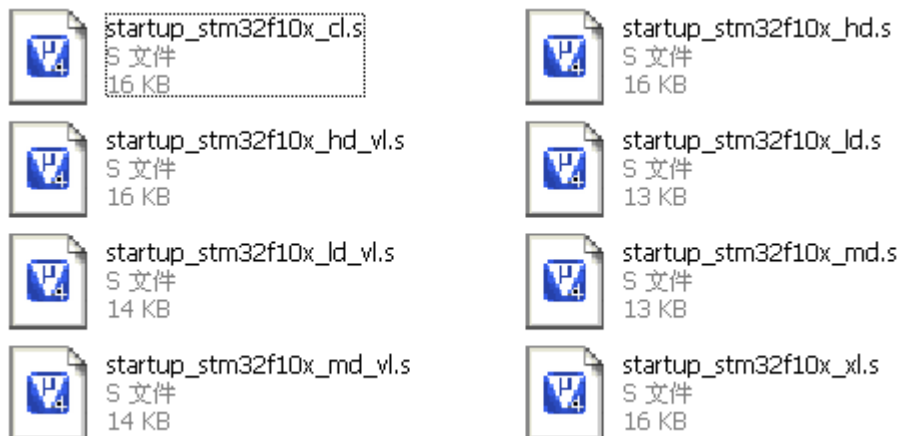
#define MODIFY_REG(REG, CLEARMASK, SETMASK)  WRITE_REG((REG),
                                                    (((READ_REG(REG)) & ~(CLEARMASK)) | (SETMASK)))
```

关于 stm32f10x_conf.h :

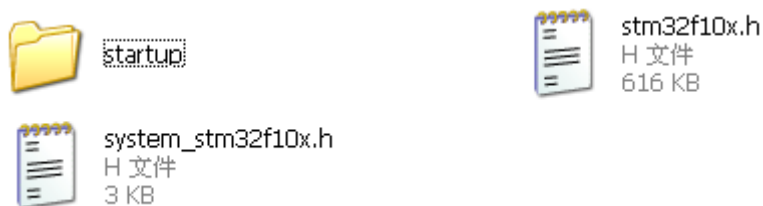
他是一个起到开关作用的头文件，0027 行开始是对前文提到的那堆头文件做开关，有人会问如何做开关，其实只要//注释掉不需使用的头文件即可，而软件层次结构图中的 PPP.X 就是相关的那些.h 和.c 文件。

看了这里，是否对 STM32 的软件结构有所了解了呢？其实只需要从 stm32f10x.h 开始顺藤摸瓜即可了的。还需要提醒的是，在做好这些文件衔接的工程中还需要一个启动文件，不知道你们是否在玩 51 的时候接触过，但是 STM32 的启动文件有些不同，虽说开发时不需要你去了解，因为早已有先人写好了，但是希望有兴趣的可以知道启动过程下是如何运作的。在 inc 里找不到的文件可以在这个路径下试试找，启动文件也在里面：

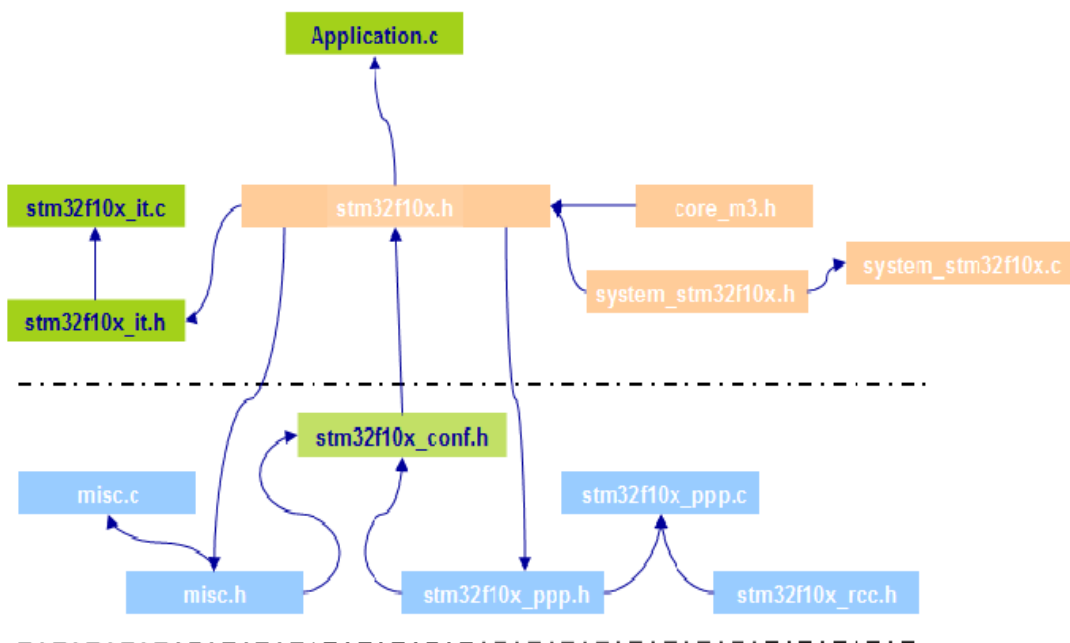
STM32F10x_StdPeriph_Lib_V3.4.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x



在刚才的路径下你还可以找到：



别急，这张图还没彻底讲完，试问说起 `it` 你们会想起什么？



`It` 是 STM32 的中断函数，这个是独立在库文件之外的，暂时还没用到，不过可以在这个路径下找到相关内容。

STM32F10x_StdPeriph_Lib_V3.4.0\Project\STM32F10x_StdPeriph_Template

E-mail: Poseidonstorm@126.com or 471661781@qq.com

以下是我按照上面对软件结构的理解，建立的一套工程模板。原因是如果使用库，那么建立工程时会很繁琐，如果有了一套适宜自己的模板，那么相信开发起来会很顺手。



其中要说明的是“.LIB”文件，这是我自己生成的一个包括所有库函数中.c文件集成库文件，具体生成方法以后会介绍到，需要自行添加“stm32f10x_lib.h”，现在还是希望读者能有针对的添加需要的.c文件和必须的.h文件来建立工程。

另外，需要知道一些外设的缩略词，详情见下图：

缩写	外设/单元	NVIC	嵌套中断向量列表
ADC	模数转换器	PWR	电源/功耗控制
BKP	备份寄存器	RCC	复位与时钟控制器
CAN	控制器局域网模块	RTC	实时时钟
DMA	直接内存存取控制器	SPI	串行外设接口
EXTI	外部中断事件控制器	SysTick	系统滴嗒定时器
FLASH	闪存存储器	TIM	通用定时器
GPIO	通用输入输出	TIM1	高级控制定时器
I2C	内部集成电路	USART	通用同步异步收发
IWDG	独立看门狗	WWDG	窗口看门狗

C: 课后练习

第一讲 泾渭分明的意思现在已经体现出来了，旨在区分一些本来相对模糊地概念，将STM32的层次深入的剖析，希望有所帮助。课后练习是根据自身理解，记住是自身理解，而不是照抄别人的方法，建立一个工程模板。

Ling.Ju