

由于新冠疫情的影响，全球供应链受到了极大的影响，众多芯片价格更是一涨再涨，以出货量最大，应用面最广的stm32f103系列为例，价格由正常价格的7元左右一片涨到了80多元一片，价格整整涨了十多倍。更严重的问题是，你很可能面临着，即使有钱也拿不到货的境地，供货周期长达三个月，这无论是在bom成本上还是交付上都是无法接受的。正是由于这样的困难，国产的芯片迎来了波替换潮，对于一些要求不高的嵌入式应用场合，stc系列51单片机，由于其极低的价格、稳定的货源，成为不错的选择之一。

stm32f103c8与stc8a8k64d4对比主要资源对比

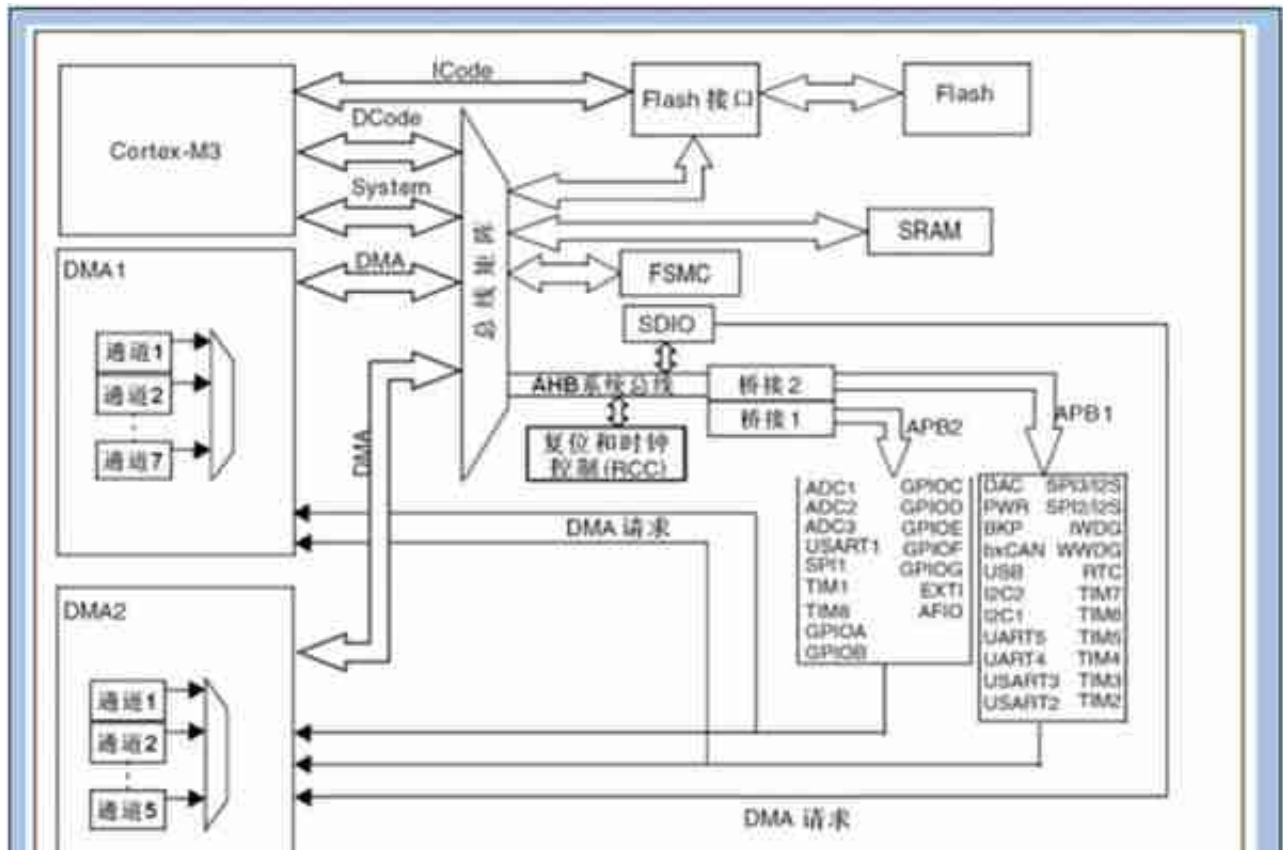
项	STM32F103V8	STC8A8K64D4
GPIO	37pin	43pin
主频	72M	45M
flash	64K	64K
RAM	20K	8K
UART	3	4
ADC	10 ch,10位	15ch, 12位
定时器	1tick + 4个32位	4个
乘除法指令	单周期硬件乘除法	无
价格	64元	4.5元

最重要的是：STC有现货

针对stc8a8k64d4的硬件资源，由于其没有硬件乘除法，导致其在需要进行大量数据计算的应用场景上效率低下，对于一般的应用场景，stc在硬件资源上完全够用，在现有的供应链条件下，STC是一个非常不错的选择。

俗话说有利就有弊，STC带来了不少的硬件成本优势，但是其在编程上却带来了一些不适应，我们在代码编写时，需要特别注意。

由于stc系列单片机是基于51架构，虽然它在内核上进行了加强，但是由于时代的局限性，51架构和现代的MCU架构相比，存在不少的差距。由于现代编译器不断的演进，我们大部人都是使用C语言进行编程，这在一定程度上，帮助我们屏蔽了架构层面的巨大差异。而且由于架构的落后，再加宏晶公司与ST公司本身在IP上面的技术差距，这导致我们在使用STC51的外设时，也有很大的差异。下图是51的架构和stm32的架构，可以看出明显的差距。



编程模型的区别 -- 函数重入

对于C语言的编程模型（请参看我前面的文章《C语言代码组成 - BSS、Data、Stack、Heap、Code、Const》），51和ARM在处理stack上，有巨大的区别。51加架构中，sp指针指向的区域是pdata区域，这个区域最大情况只有256个字节，作为栈来说，是不可能用来分配局部变量的。因此keil编译器对于临时变量，采用了一种叫作变量覆盖的技术，即多个函数使用同一区域作为临时变量区，下一个函数调用运行时，会将上一个函数运行的临时变量覆盖掉。这样就解决了栈空间不足的问题。而且因为没有进出栈的操作，也提高了代码效率。但这带来的最大的问题就是函数重入问题，即51代码全部是不可重入的，如果一个函数在中断中被调用，那就意味着这个函数必须是可重入的，才能正常运行，而且这个中断函数所调用的下级函数也必须是可重入的。如何解决重入问题：

1. keil通过模拟堆栈技术，可以实现函数的重入，通过增加关键字 reentrant 将一个函数定义为可重入函数；
2. 将需要重入的函数定义为原子操作，即进入该函数时，禁止全局中断，执行完成后，开启全局中断；
3. 中断函数中不调用其它函数，只是置标志，进行变量操作，然后再在main的主循环中查询标志，再执行相应的操作；

以上三种方法，各有缺点，方法一简单粗暴，需要消耗更多的资源；方法二会导致中断响应不及时；方法三导致中断响应不及时。所以用户需要根据项目特点，使用合适的策略；

函数传参

C51通过寄存器传递参数，最多只能传递三个参数