

# 如何使用 C 语言来编写 MSP430 的高质量代码

中国计量学院 mirandali 2006-5-27

摘自《MSP430 系列单片机使用 C 语言程序设计》

微处理器一般用于特定环境和特定用途,出于成本、功耗和体积的考虑,一般都要求尽量节省使用资源,并且,由于微处理器硬件一般都不支持有符号数、浮点数的运算,且运算位有限,因此,分配变量时必须仔细。另外要说明的是,速度和存储器的消耗经常是 2 个不可兼顾的目标,在多数情况下,编程者必须根据实际情况作出权衡和取舍。

需要注意的事项如下:

- 1) 通常在满足运算需求的前提下,尽量选择为变量定义字节少的数据类型。

比如最常用的 int 和 char, int 是 16 位的, char 是 8 位的,如果没有必要,不要使用 int,而且使用 char 也最好使用 unsigned char。运行时,可以在变量窗口看到,使用类型为 unsigned char 的变量是 16 进制的格式,而使用 int 的是十进制格式,如果 char 没有定义为 unsigned,会出现负号,如果没有必要的话,在 430 中是不需要负数的。

- 2) 尽量不用过长的数据类型,如 long、long long 和 double

3) MSP430 的 C 编译器不支持位寻址,所以运算中尽量减少位操作,对于只有“是”和“否”的变量,如果 RAM 容量允许,则可分配为 unsigned char 类型,可提高运算速度。如果分配为某字节的某个位,可以减少存储器的消耗,但是会降低运算速度

4) 避免使用浮点数,尽量使用定点数进行小数运算。如果必须使用浮点数,则尽量用 32 位的 float,而不是 64 位的 double

- 5) 尽量将变量分配为无符号数据类型

6) 对于指针变量,如果声明后其值不再改变,则声明为 const 类型,这样编译器编译时能更好的优化生成的代码

7) 尽可能的使用局部变量而非全局变量或者静态变量 (static)。这样有利于编译器编译时更好的优化生成的代码

8) 避免对局部变量使用 &取地址符。因为这样会使编译器无法把此变量放在 CPU 的寄存器中,而是放在 RAM 中,从而失去了优化的机会

- 9) 仅在模块内使用的变量声明为 static,有利于优化代码

- 10) 如果堆栈空间有限,尽量减少函数调用的层次和递归调用

11) 如果传送参数过多，可将参数组成一个数组或者结构体，然后用指针传递

12) 某些变量在中断程序和普通级别程序中都会被用到，所以必须加以保护。

将变量声明为 `volatile` 类型，编译器优化时就不会移动它，对它的访问不会被延迟。

为保证对 `volatile` 的变量不被打断，为此，可以在访问它的部分（即访问它的函数）前面加上 `__monitor` 的声明

