

第三十八节：第二种解决“运算过程中意外溢出”的便捷方法。

【38.1 意外溢出。】

运算过程中的意外溢出，稍不注意，就中招，不信，请看下面的例子：

```
/*---C 语言学习区域的开始。-----*/
unsigned long a=0;
unsigned int x=1000;
unsigned int y=3000;
void main() //主函数
{
    a=x*y;    //猜猜 a 是多大？
    View(a); //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
    while(1)
    {
    }
}
/*---C 语言学习区域的结束。-----*/
```

猜猜 a 是多大？很多人以为理所当然 3000000，但是实际上是 50880！中招了吧。莫名其妙的 50880，就是因为意外溢出所致。怎么办呢？请看下面介绍的两种解决办法。

【38.2 第一种办法：引入中间变量。】

我在前面章节中曾多次说过“为了避免运算过程中的意外溢出，建议大家把所有参与运算的变量都用 unsigned long 类型的变量，如果不是 unsigned long 类型的变量，就引入 unsigned long 类型的中间变量。”这种老方法如下：

```
/*---C 语言学习区域的开始。-----*/
unsigned long a=0;
unsigned int x=1000;
unsigned int y=3000;
unsigned long s; //引入的 unsigned long 中间变量。
unsigned long t; //引入的 unsigned long 中间变量。
void main() //主函数
{
    s=x; //先把变量的数值搬到 unsigned long 中间变量。
    t=y; //先把变量的数值搬到 unsigned long 中间变量。
    a=s*t; //中间变量代表原始变量进行运算。
    View(a); //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
    while(1)
    {
    }
}
/*---C 语言学习区域的结束。-----*/
```

这一次，运算结果是正确的 3000000。

现在反省了一下，这种办法虽然可靠实用，但是显得有点罗嗦，而且引入的中间变量也无形中增加了一点内存。还有没有更好的办法？请看下面介绍的第二种办法。

【38.3 第二种办法：C 语言的类型强制转换。】

前面章节提到，括号在 C 语言中有强制的意思，可以强制改变优先级，在本节也可以临时强制改变运算过程中的变量类型。在运算过程中临时强制改变类型变量，就可以省去额外引入的中间变量，这种方法相比上面第一种老办法确实更便捷灵活。

```
/*---C 语言学习区域的开始。-----*/
unsigned long a=0;
unsigned int x=1000;
unsigned int y=3000;
void main() //主函数
{
    a=(unsigned long)x*(unsigned long)y; //添加的两个括号就是类型的强制转换。
    View(a); //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
    while(1)
    {
    }
}
/*---C 语言学习区域的结束。-----*/
```

这一次，运算结果也是正确的 3000000。

多说一句，除了上述的乘法运算之外，其它的加、减、除法运算适不适用呢？虽然我还没有逐个测试，但是我感觉应该是都适用的。因此，在“加、减、除”等运算中，在必要的时候，也要在相关的变量的前缀加上类型的强制转换。

【38.4 全局变量和局部变量。】

先插入一个知识点，细心的朋友会发现，我上面的例子中，定义的变量都放在了 main 函数之外的上面，这种把变量定义在函数外面的变量叫全局变量，以前例子中定义在函数内的变量叫局部变量。

```
unsigned char a; //这个在函数之外，叫全局变量
void main() //主函数
{
    unsigned char b; //这个在函数之内，叫局部变量
    while(1)
    {
    }
}
```

上面例子中，a 定义在函数之外是全局变量，b 定义在函数之内是局部变量。全局变量与局部变量有什么不一样呢？以后的章节会仔细讲解这方面的知识，现在暂时不讲。之所以在这里提出这个知识点，是因为我今后的例子很多变量可能都会定义成全局变量，因此先在这里给大家打个招呼，知道 C 语言有这样一种语法就可以。

【38.5 例程练习和分析。】

现在编写一个程序来验证刚才讲到的主要内容：

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

unsigned long  a=0;
unsigned long  b=0;
unsigned long  c=0;
unsigned int  x=1000;
unsigned int  y=3000;

unsigned long  s; //中间变量
unsigned long  t;

void main() //主函数
{
    a=x*y; //意外溢出

    s=x;    //引入中间变量
    t=y;
    b=s*t;

    c=(unsigned long)x*(unsigned long)y;    //类型的强制转换

    View(a); //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
    View(b); //把第 2 个数 a 发送到电脑端的串口助手软件上观察。
    View(c); //把第 3 个数 a 发送到电脑端的串口助手软件上观察。

    while(1)
    {
    }
}

/*---C 语言学习区域的结束。-----*/
```

在电脑串口助手软件上观察到的程序执行现象如下：

开始...

第 1 个数

十进制:50880

十六进制:C6C0

二进制:1100011011000000

第 2 个数

十进制:3000000

十六进制:2DC6C0

二进制:1011011100011011000000

第 3 个数

十进制:3000000

十六进制:2DC6C0

二进制:1011011100011011000000

分析:

通过实验结果,发现在单片机上的实验结果和我们的分析是一致的。

【38.6 如何在单片机上练习本章节 C 语言程序?】

直接复制前面章节中第十一节的模板程序,练习代码时只需要更改“C 语言学习区域”的代码就可以了,其它部分的代码不要动。编译后,把程序下载进带串口的 51 学习板,通过电脑端的串口助手软件就可以观察到不同的变量数值,详细方法请看第十一节内容。