

第十九节：加法运算的溢出。

【19.1 什么是加法运算的溢出？】

前面章节介绍的三种数据类型 `unsigned char` , `unsigned int` , `unsigned long`, 它们的数值都是有最大范围的, 分别是 255, 65535, 4294967295。如果运算结果超过了变量本身的最大范围, 会出现什么结果、有什么规律, 这就是本节要讲的溢出。

(1) 什么是溢出? 先看一个例子如下:

```
unsigned char a;  
a=0x8536;
```

分析:

因为 `a` 是 `unsigned char` 变量, 位数是 8 位, 也就是一个字节, 而 `0x8536` 是 16 位, 两个字节, 这种情况下, 把两字节的 `0x8536` 强行赋值给单字节变量 `a`, 变量 `a` 只能接收到最低 8 位的一个字节 `0x36`, 而高 8 位的一个字节 `0x85` 就被丢失了, 这个就是本节所说的溢出。

(2) 再看一个例子如下:

```
unsigned char b=0xff;  
b=b+1;
```

分析:

`b` 默认值是 `0xff`, 再加 1 后, 变成了 `0x0100` 保存在一个隐藏的中间变量, 然后再把这个中间变量赋值给单字节变量 `b`, `b` 只能接收到低 8 位的一个字节 `0x00`, 所以运算后 `b` 的数值由于溢出变成了 `0x00`。

(3) 再看一个例子如下:

```
unsigned char c=0xff;  
c=c+2;
```

分析:

`c` 默认值是 `0xff`, 再加 2 后, 变成了 `0x0101` 保存在一个隐藏中间变量, 然后再把这个中间变量赋值给单字节变量 `c`, `c` 只能接收到低 8 位的一个字节 `0x01`, 所以运算后 `c` 的数值由于溢出变成了 `0x01`。

(4) 再看一个例子如下:

```
unsigned int d=0xfffe;  
d=d+5;
```

分析:

`d` 默认值是 `0xfffe`, 再加 5 后, 变成了 `0x10003` 保存在一个隐藏中间变量, 由于这个隐藏的中间变量是 `unsigned int` 类型, 只能保存两个字节的的数据, 所以在中间变量这个环节就溢出了, 实际上隐藏的中间变量只保存了 `0x0003`, 然后再把这个中间变量赋值给 16 位的两字节变量 `d`, `d` 理所当然就是 `0x0003`。

(5) 再看一个例子如下:

```
unsigned long e=0xfffffffffe;  
e=e+5;
```

分析:

`e` 默认值是 `0xfffffffffe`, 再加 5 后, 变成了 `0x100000003` 保存在一个隐藏中间变量, 由于这个隐藏的中间变量是 `unsigned long` 类型, 只能保存四个字节的的数据, 所以在中间变量这个环节就溢出了, 实际上隐藏的中间变量只保存了 `0x00000003`, 然后再把这个中间变量赋值给 32 位的四字节变量 `e`, `e` 理所当然也是 `0x00000003`。

【19.2 例程练习和分析。】

现在我们编写一个程序来验证上面讲到的例子：
程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

void main() //主函数
{
    unsigned char a;          //一个字节
    unsigned char b=0xff;     //一个字节
    unsigned char c=0xff;     //一个字节
    unsigned int  d=0xfffe;    //两个字节
    unsigned long e=0xffffffff; //四个字节

    a=0x8536;
    b=b+1;
    c=c+2;
    d=d+5;
    e=e+5;

    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 b 发送到电脑端的串口助手软件上观察。
    View(c); //把第 3 个数 c 发送到电脑端的串口助手软件上观察。
    View(d); //把第 4 个数 d 发送到电脑端的串口助手软件上观察。
    View(e); //把第 5 个数 e 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:54

十六进制:36

二进制:110110

第 2 个数

十进制:0

十六进制:0

二进制:0

第 3 个数

十进制:1

十六进制:1

二进制:1

第 4 个数

十进制:3

十六进制:3

二进制:11

第 5 个数

十进制:3

十六进制:3

二进制:11

分析:

通过实验结果，发现在单片机上的计算结果和我们的分析是一致的。

【19.3 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。