

## 手把手教你学单片机的C语言程序设计(八)

## 运算符与表达式(续)

◆ 吕超亚

5. 自增和自减运算符  
与表达式

自增和自减运算符是C语言中特有的一种运算符,它们的作用分别是对运算对象作加1和减1运算,其功能如下:

++ 自增运算符。如:a++,++a

-- 自减运算符。如:a--,--a

看起来 a++ 和 ++a 的作用都是使变量 a 的值加 1,但是由于运算符 ++ 所处的位置不同,使变量 a+1 的运算过程也不同。++a(或 --a)是先执行 a+1(或 a-1)操作,再使用 a 的值,而 a++(或 a--)则是先使用 a 的值,再执行 a+1(或 a-1)操作。

增量运算符 ++ 和减量运算符 -- 只能用于变量,不能用于常数或表达式。

## 实验一

进行自增运算 a++ 和 ++b (范围为 0~9),并将结果在 LED/16\*2 字符液晶试验板上输出显示,其中 a 在个位数码管显示,b 在十位数码管显示。

在我的文档中建立一个文件目录(cs16),然后建立 cs16.uv2 的工程项目,最后建立源程序文件(cs16.c)。

输入下面的程序:

```
#include <REG51.H> // 序号(以下同):1
#define uchar unsigned char //2
#define uint unsigned int //3
uchar code SEG7[10]={0xc0,0xf9,0xa4,
0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,}; //4
//=====5
void delay(uint k) //6
{ //7
uint i,j; //8
for(i=0;i<k;i++){ //9
```

```
for(j=0;j<121;j++) //10
{ //11
} //12
//=====13=====
void main(void) //14
{ //15
uchar a=0,b=0; //16
while(1) //17
{ //18
P0=SEG7[a++]; //19
P1=SEG7[++b]; //20
delay(1000); //21
if((a==9)||(b==9))while(1); //22
} //23
} //24
```

编译通过后,将生成的 cs16.hex 文件烧录到 89S51 芯片中,将芯片插入到 LED/16\*2 字符液晶试验板上,试验板上接通 9V 电源,右边 2 个 LED 数码管显示“10”→“11”...“98”。

为什么两个数字不一样呢?原因就是 ++b 是先执行 b+1 操作,然后再显示。而 a++ 是先将 a 送显,然后再执行 a+1 操作。因此显示结果不一样。

分析程序。

序号 1 (程序解释,以下同): 包含头文件 REG51.H。  
序号 2、3: 数据类型的宏定义。  
序号 4: 数码管 0~9 的字形码。  
序号 5: 程序分隔。  
序号 6~12: 延时子函数。  
序号 13: 程序分隔。  
序号 14: 定义函数名为 main 的主函数。  
序号 15: main 的主函数开始。  
序号 16: 定义无符号字符型变量 a、b 并赋初值 0。  
序号 17: 主循环。  
序号 18: 主循环开始。  
序号 19: a 的值送 P0 口显示,随后 a+1。  
序号 20: b 的值 +1 后送 P1 口显示。  
序号 21: 延时 1 秒便于观察。  
序号 22: 如果 a 或 b 的值为 9,则动态停机。

序号 23: 主循环结束。

序号 24: main 的主函数结束。

## 6. 逗号运算符与表达式

在C语言中,逗号“,”运算符可以将两个(或多个)表达式连接起来,称为逗号表达式。逗号表达式的一般形式为:

表达式 1,表达式 2,...表达式 n  
逗号表达式的运算过程是:先算表达式 1,再算表达式 2,...依次算到表达式 n。

## 7. 条件运算符与表达式

条件运算符是C语言中惟一的一个三目运算符,它要求有三个运算对象,用它可以将三个表达式连接构成一个条件表达式。条件表达式的一般形式如下:

表达式 1?表达式 2:表达式 3

其功能是首先计算表达式 1,当其值为真(非 0 值)时,将表达式 2 的值作为整个条件表达式的值;当逻辑表达式的值为假(0 值)时,将表达式 3 的值作为整个条件表达式的值。

例如: max=(a>b)?a:b

当 a>b 成立时, max=a

否则 a>b 不成立, max=b

## 8. 位运算符与表达式

能对运算对象进行按位操作是C语言的一大特点,正是由于这一特点使C语言具有了汇编语言的一些功能,从而使之能对计算机的硬件直接进行操作。C语言中共有 6 种位运算符。

位运算符的作用是按位对变量进行运算,并不改变参与运算的变量的

值。若希望按位改变运算变量的值,则应利用相应的赋值运算。另外位运算符不能用来对浮点型数据进行操作。

位运算符的优先级从高到低依次是:

按位取反(~)→左移(<<)和右移(>>)→按位与(&)→按位异或(^)→按位或(|)。

**表 1 列出了按位取反、按位与、按位或和按位异或的逻辑真值**

x	y	~x	~y	x&y	x y	x^y
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

## 实验二

在 LED/128\*64 图形液晶试验板上,实现两个变量 x、y 的位运算,其结果输出到 8 个 LED 上显示。

在我的文档中建立一个文件目录(cs17),然后建立 cs17.uv2 的工程项目,最后建立源程序文件(cs17.c)。

输入下面的程序:

```
#include <REG51.H> // 序号(以下同):1
#define uchar unsigned char //2
#define uint unsigned int //3
//=====4
void delay(uint k) //5
{ //6
    uint i,j; //7
    for(i=0;i<k;i++){ //8
        for(j=0;j<121;j++) //9
            ; //10
    } //11
} //12
//=====12
void main(void) //13
{ //14
    uchar x=57,y=136; //15
    P0=~x; //16
    delay(3000); //17
    P0=~y; //18
    delay(3000); //19
    P0=x&y; //20
    delay(3000); //21
    P0=x|y; //22
    delay(3000); //23
    P0=x^y; //24
    delay(3000); //25
```

```
P0=x<<1; //26
delay(3000); //27
P0=y<<2; //28
delay(3000); //29
while(1); //30
} //31
```

编译通过后,将生成的 cs17.hex 文件烧录到 89S51 芯片中,将芯片插入到 LED/128\*64 图形液晶试验板上, SX 双排针上的 8 个短路块全部取下,试验板接通 5V 电源,8 个 LED 开始灯光亮灭变化,间隔为 3 秒。其亮灭的顺序为(0 代表亮、1 代表灭):

```
11000110
↓
01110111
↓
00001000
↓
10111001
↓
10110001
↓
01110010
↓
00100000
```

这些灯光亮灭是否正确,我们在这里分析一下:

x 赋值 57,转化成二进制为 00111001。y 赋值 136,转化成二进制为 10001000。  
 x 取反后为:11000110(正确)。  
 y 取反后为:01110111(正确)。  
 x&y 后为:00001000(正确)。  
 x|y 后为:10111001(正确)。  
 x^y 后为:10110001(正确)。  
 x<<1 后为:01110010(正确)。  
 Y<<2 后为:00100000(正确)。

分析程序。

序号 1 (程序解释,以下同):包含头文件 REG51.H。  
 序号 2、3:数据类型的宏定义。  
 序号 4:程序分隔。  
 序号 5~11:延时子函数。  
 序号 12:程序分隔。  
 序号 13:定义函数名为 main 的主函数。  
 序号 14:main 的主函数开始。  
 序号 15:定义无符号字符型变量 x、y 并赋初值 57,136。  
 序号 16:x 的值取反后送 P0 口。  
 序号 17:延时 3 秒便于观察。  
 序号 18:y 的值取反后送 P0 口。

序号 19:延时 3 秒便于观察。  
 序号 20:x、y 按位与后的值送 P0 口。  
 序号 21:延时 3 秒便于观察。  
 序号 22:x、y 按位或后的值送 P0 口。  
 序号 23:延时 3 秒便于观察。  
 序号 24:x、y 按位异或后的值送 P0 口。  
 序号 25:延时 3 秒便于观察。  
 序号 26:x 的值左移 1 位后送 P0 口。  
 序号 27:延时 3 秒便于观察。  
 序号 28:y 的值左移 2 位后送 P0 口。  
 序号 29:延时 3 秒便于观察。  
 序号 30:动态停机。  
 序号 31:main 主函数结束。

## 9. 强制类型转换运算符与表达式

C 语言中的圆括号“()”也可作为强制类型转换运算符,它的作用是将表达式或变量的类型强制转换为所指定的类型。

在 C 语言程序中进行算术运算时,需要注意数据类型的转换。有两种数据类型转换方式,即隐式转换和显式转换。隐式转换是在对程序进行编译时由编译器自动处理的。隐式转换遵循以下规则:

- (1). 所有 char 型的操作数转换成 int 型。
- (2). 用运算符连接的两个操作数如果具有不同的数据类型,按以下次序进行转换:  
 如果一个操作数是 float 类型,则另一个操作数也转换成 float 类型。  
 如果一个操作数是 long 类型,则另一个操作数也转换成 long 类型。  
 如果一个操作数是 unsigned 类型,则另一个操作数也转换成 unsigned 类型。

(3). 在对变量赋值时发生的隐式转换,将赋值号“=”右边的表达式类型转换成赋值号左边变量的类型。

例如:

把整数赋值给字符型变量,则整数的高 8 位将丧失。

把浮点数赋值给整型变量,则小数部分将丧失。

在 C 语言中只有基本数据类型

## 单片机的基本结构与工作原理(三)

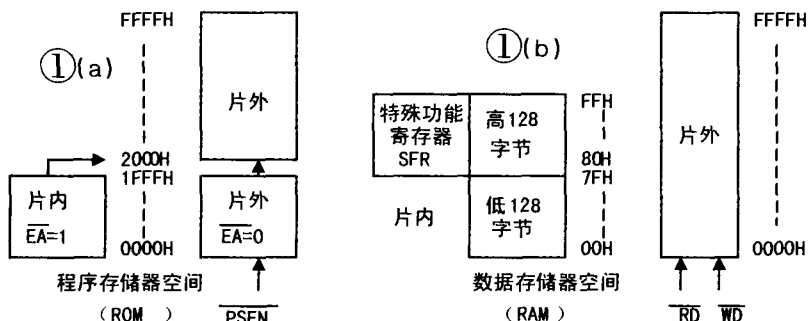
## ——AT89S52 的存储器结构及其操作方式

◆ 陈阳海

程序存储器和数据存储器是单片机最重要的外围单元。本讲我们将介绍 AT89S52 单片机的存储器结构及其操作方式。

## 一、AT89S52 单片机的存储器结构

与 80C51 系列单片机一样, AT89S52 单片机的程序存储器和数据存储器是两个独立的存储器空间, 图 1 是 AT89S52 单片机的存储器结构示意图, 图 1a 是程序存储器结构示意图, 图 1b 是数据存储器的结构示意图。



## 三、程序存储器的结构及其操作方式

## 1. 程序存储器的结构

① 关于片内/片外程序存储器的容量及其选择

AT89S52 是增强型 80C51 单片机, 片内有 8k 字节 Flash ROM, 能够满足大多数应用系统的需要。如果不够

用, 还可以通过并行总线扩展外部程序存储器, 如图 1a 所示。扩展的方法是: 当片内/片外程序存储器选择脚 EA 接高电平(EA=1)时, 程序存储器地址先从片内程序存储器的 0000H 开始执行, 如果有外扩的程序存储器, 当程序超过 8k 字节(1FFFH)时, 会自动转向片外程序存储器去执行; 而当 EA 接地(EA=0)时, 程序直接从片外程序存储器的 0000H 开始执行(参见图 1a 和第五讲《单片机的基本结构与工作原理(二)》一文中的表 1)。AT89S52 程序存储器的寻址范围为 0000H~FFFFH, 也就是说, 其程序存储器可直接扩展至 64k 字节。

② 关于 AT89S52 片内程序存储器的几个特殊地址空间

对于 AT89S52 单片机片内程序存储器的存储空间, 大家需要记住其中几个特殊地址:

·0000H

复位后, 程序计数器 PC 的值为 0000H, 故单片机复位后总是从 0000H 单元开始执行程序。

(即 char、int、long 和 float) 可以进行隐式转换。

其余的数据类型不能进行隐式转换, 例如, 我们不能把一个整数利用隐式转换赋值给一个指针变量, 在这种情况下就必须利用强制类型转换运算符来进行显式转换。强制类型转换运算符的一般使用形式为:

(类型)=表达式

在《手把手教你学单片机的 C 语言程序设计(四)》中, 已刊载了数据类型的隐式转换及强制转换两个实验, 读者朋友可参考。

## 10. sizeof 运算符与表达式

C 语言中提供了一种用于求取数

据类型、变量以及表达式的字节数的运算符 sizeof,

该运算符的一般使用形式为:

sizeof(表达式) 或 sizeof(数据类型)

注意, sizeof 是一种特殊的运算符, 不要认为它是一个函数。实际上, 字节数的计算在编译时就完成了, 而不是在程序执行的过程中才计算出来的。

配文优惠邮购: Keil C51 Windows 集成开发环境(已汉化正式版光盘, 邮购代号: K1): 46 元。TOP851 多功能编程器(邮购代号: B1): 220 元。LED/128\*64 图形液晶试验板(邮购代号: S3): 160 元。LED/16\*2 字符液晶试验板(邮购代号: S2): 140 元。16\*2 字符

型液晶显示模组(邮购代号: L1): 80 元。128\*64 点阵型液晶显示模组(邮购代号: L2): 160 元。5V 高稳定专用稳压电源(邮购代号: D1): 30 元。每次邮费保价费 12 元。开发票另加货款 7% (汇款时注明)。邮购时只需在附言栏中写明邮购代号及数量并附上联系电话即可。邮局汇款邮购: 上海市闵行区莲花路 2151 弄 57 号 201 室, 邮编: 201103, 联系人: 吕超亚, 银行汇款购买(汇款后电话告知): 户名: 上海红棱电子有限公司, 开户行: 上海浦东发展银行闵行区吴中路支行, 帐号: 076499-98530154740000965, 电话(传真): 021-64654216 13774280345, 网址: <http://www.hlelectron.com>, 技术支持 E-mail: zhx2151@sohu.com。