

手把手教你学单片机的C语言程序设计(七)

运算符与表达式

◆ 吕超亚

C语言对数据有很强的表达能力,具有十分丰富的运算符,利用这些运算符可以组成各种表达式及语句。运算符就是完成某种特定运算的符号。表达式则是由运算符及运算对象所组成的具有特定含义的一个式子。由运算符或表达式可以组成C语言程序的各种语句。C语言是一种表达式语言,在任意一个表达式的后面加一个分号“;”就构成了一个表达式语句。

按照运算符在表达式中所起的作用,可分为算术运算符、关系运算符、逻辑运算符、赋值运算符、增量与减量运算符、逗号运算符、条件运算符、位运算符、指针和地址运算符、强制类型转换运算符和sizeof运算符等。运算符按其其在表达式中与运算对象的关系,又可分为单目运算符、双目运算符和三目运算符等。单目运算符只需要有一个运算对象,双目运算符要求有两个运算对象,三目运算符要求有三个运算对象。

1. 算术运算符与表达式

C语言提供的算术运算符有:

- + 加或取正值运算符。如:1+2的结果为3。
- 减或取负值运算符。如:4-3的结果为1。
- * 乘运算符。如:2*3的结果为6。
- / 除运算符。如:6/3的结果为2。
- % 模运算符,或称取余运算符。如:7%3的结果为1。

上面这些运算符中加、减、乘、除为双目运算符,它们要求有两个运算对象。取余运算要求两个运算对象均为整型数据,如果不是整型数据可以采用强制类型转换。例如8%3的结果为2。取正值和取负值为单目运算符,它们的运算对象只有一个,分别是取运算对象的正值和负值。

实验一

进行数学运算并将结果在LED/16*2字符液晶试验板上输出显示。

在我的文档中建立一个文件目录(cs13),然后建立cs13.uv2的工程项目,最后建立源程序文件(cs13.c)。

输入下面的程序:

```
#include <REG51.H> // 序号(以下同):1
```

```
#define uchar unsigned char //2
uchar code SEG7 [10]=
(0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf
8,0x80,0x90,); //3
//=====4==
void main(void) //5
{ //6
uchar a,b,c,out; //7
a=100; //8
b=60; //9
c=9; //10
out=a+3*(b-c)/2; //11
P2= SEG7[ out/100]; //12
P1= SEG7[ (out%100)/10]; //13
P0= SEG7[ out%10]; //14
while(1); //15
} //16
```

编译通过后,将生成的cs13.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3个LED数码管显示“176”。

这个结果正确吗?我们验证一下:由于b-c加了括号,因此优先级最高,其差值为51。接下来乘法的优先级要高于其它运算符,因此51乘3等于153。再下来153除2得76.5,舍去小数部分,得76。76加100,结果为176。完全正确。

分析程序。

```
序号1(程序解释,以下同):包含头文件REG51.H。
```

- 序号2:数据类型的宏定义。
- 序号3:数码管0-9的字形码。
- 序号4:程序分隔。
- 序号5:定义函数名为main的主函数。
- 序号6:main的主函数开始。
- 序号7:定义无符号字符型变量a,b,c,out。
- 序号8:a赋值100。
- 序号9:b赋值60。
- 序号10:c赋值9。
- 序号11:数学运算,其结果放out。
- 序号12:取出out的百位数送P2口显示。说明:out除100,得out的百位数(其十、个位均成为小数而舍去)。
- 序号13:取出out的十位数送P1口显示。说明:out余100,得out的十、个位数,然后再除10,取得out的十位数。
- 序号14:取出out的个位数送P0口显示。说明:out余10,得out的个位数。
- 序号15:动态停机。
- 序号16:main的主函数结束。

2. 关系运算符与表达式

C语言中有以下的关系运算符:

- > 大于。如:x>y。
- < 小于。如:a<4。
- >= 大于等于。如:x>=2。
- <= 小于等于。如:a<=5。
- == 测试等于。如:a==b。
- != 测试不等于。如:x!=5。

前4种关系运算符(>,<,>=,<=)具有相同的优先级,后两种关系运算符

(==, !=)也具有相同的优先级,但前4种的优先级高于后2种。

关系运算符通常用来判别某个条件是否满足,关系运算的结果只有“真”和“假”两种值。当所指定的条件满足时结果为1,条件不满足时结果为0。1表示“真”,0表示“假”。

实验二

在LED/128*64图形液晶试验板上实现:

按下S2键或S1键后,输入两个小于10的整数a、b,其中a在数码管上十位上显示,b在数码管个位上显示。由程序判断a、b的大小:若a>b,则数码管千位上显示“H”;若a=b,则数码管千位上显示“=”;若a<b,则数码管千位上显示“L”。

在我的文档中建立一个文件目录(cs14),然后建立cs14.uv2的工程项目,最后建立源程序文件(cs14.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]={0x3f,0x06,
0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f}; //4
//=====5=====
uchar a=0,b=0,c; //6
//-----7-----
uchar key_s1(void); //8
uchar key_s2(void); //9
void delay(uint k); //10
//-----11-----
void main(void) //12
{
    uchar i; //13
    while(1) //14
    {
        //15
        b=key_s1(); //16
        a=key_s2(); //17
        if(a>b)c=0x76; //18
        if(a<b)c=0x38; //19
        if(a==b)c=0x09; //20
        for(i=0;i<100;i++) //21
        {
            //22
            P0=SEG7[a]; //23
            P2=0xdf; //24
            delay(2); //25
```

```
        P0=SEG7[b]; //26
        P2=0xef; //27
        delay(2); //28
        P0=c; //29
        P2=0x7f; //30
        delay(2); //31
    } //32
} //33
} //34
//-----35-----
void delay(uint k) //36
{
    uint i,j; //37
    for(i=0;i<k;i++){ //38
        for(j=0;j<121;j++){ //39
            //40
            //41
        } //42
    } //43
    uchar key_s1(void) //44
    {static uchar x; //45
    P2=0xff; //46
    if(P2!=0xff) //47
        (delay(10); //48
    if(P2==0xfe) //49
    {
        x=x+1; //50
    } //51
    } //52
    } //53
    if(x>9)x=0; //54
    return x; //55
    } //56
    //-----57-----
    uchar key_s2(void) //58
    {static uchar y; //59
    P2=0xff; //60
    if(P2!=0xff) //61
        (delay(10); //62
    if(P2==0xfd) //63
    {
        y=y+1; //64
    } //65
    } //66
    } //67
    if(y>9)y=0; //68
    return y; //69
    } //70
```

编译通过后,将生成的cs14.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,试验板上接通5V电源,右边2个数码管显示“00”,左边的千位数码管显示“=”。按下S2键,十位数码管上的数字

开始递增,当十位数码管的数字大于个位数码管时,千位数码管显示“H”;按下S1键,个位数码管上的数字开始递增,当个位数码管的数字大于十位数码管时,千位数码管显示“L”;当个位数码管的数字等于十位数码管时,千位数码管显示“=”。程序自己判别出了两个数值的大小,并将结果显示出来。

下面分析程序。

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:定义无符号字符型全局变量a、b、c。

序号7:程序分隔。

序号8~10:函数声明。

序号11:程序分隔。

序号12:定义函数名为main的主函数。

序号13:main的主函数开始。定义无符号字符型局部变量i。

序号14:while循环语句进行无限循环。

序号15:while循环语句开始。

序号16:调用S1键判断子函数,其键值返回至变量b中。

序号17:调用S2键判断子函数,其键值返回至变量a中。

序号18:若a>b,c赋值0x76。

序号19:若a<b,则c赋值0x38。

序号20:若a等于b,c赋值0x09。

序号21:for循环语句,共循环100次。

序号22:for循环语句开始。

序号23:变量a送P0口。

序号24:点亮数码管十位。

序号25:延时2ms。

序号26:变量b送P0口。

序号27:点亮数码管个位。

序号28:延时2ms。

序号29:变量c送P0口。

序号30:点亮数码管千位。

序号31:延时2ms。

序号32:for循环语句结束。

序号33:while循环语句结束。

序号34:main主函数结束。

序号35:程序分隔。

序号36~42:延时子函数。

序号43:程序分隔。

序号44:定义函数名为key_s1的S1键判断子函数。

序号45:key_s1子函数开始。定义静态的局

部变量 x。

序号 46: P2 口置全 1, 以便读取按键状态。

序号 47: 若 P2 口不为 0xff, 说明可能有键按下。

序号 48: 延时 2ms 再判。

序号 49: 若 P2 口等于 0xfe, 说明接下的为 S1 键。

序号 50~52: 变量 x 递增。

序号 53: 结束 if(P2 != 0xff) 语句。

序号 54: 若 x 的值超过 9, 则赋 0。

序号 55: 返回 x 的值。

序号 56: key_s1 子函数结束。

序号 57: 程序分隔。

序号 58~70: 函数名为 key_s2 的 S2 键判断子函数, 具体可参考序号 44~56 分析。

>>= 右移位赋值运算符

<<= 左移位赋值运算符

&= 逻辑与赋值运算符

|= 逻辑或赋值运算符

^= 逻辑异或赋值运算符

~= 逻辑非赋值运算符

复合赋值运算首先对变量进行某种运算, 然后将运算的结果再赋给该变量。复合运算的一般形式为: 变量 复合赋值运算符 表达式

例如: a+=5 等价于 a=a+5;

采用复合赋值运算符, 可以使程序简化, 同时还可以提高程序的编译效率。

```

else out=0; //21
P0= SEG7[a]; //22
P2=0xdf; //23
delay(2); //24
P0= SEG7[b]; //25
P2=0xef; //26
delay(2); //27
if(out==1)P0= 0x73; //28
else P0=0x00; //29
P2=0x7f; //30
delay(2); //31
} //32
} //33
//-----34

```

```

void delay(uint k) //35
{ //36
uint i,j; //37
for(i=0;i<k;i++){ //38
for(j=0;j<121;j++) //39
{;} //40
} //41
//-----42
uchar key_s1(void) //43
{uchar data x; //44
P2=0xff; //45
if(!P2_0)x=5; //46
else x=0; //47
return x; //48
} //49
//-----50
uchar key_s2(void) //51
{uchar data y; //52
P2=0xff; //53
if(!P2_1)y=8; //54
else y=0; //55
return y; //56
} //57

```

编译通过后, 将生成的 cs15.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, 试验板上接通 5V 电源, 右边 2 个数码管显示“00”。按下 S1 键, 个位数数码管上显示“5”; 按下 S2 键, 十位数数码管上显示“8”; 同时按下 S1、S2 键后, 观察到千位数数码管显示“P”。

对程序进行分析。

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: 定义 P2.0 的符号名为 P2_0。
 序号 5: 定义 P2.1 的符号名为 P2_1。

3. 逻辑运算符与表达式

C 语言中提供的逻辑运算符有 3 种:

|| 逻辑或

&& 逻辑与

! 逻辑非

逻辑运算的结果也只有两个: “真”为 1, “假”为 0。

逻辑表达式的一般形式为:

逻辑与: 条件式 1 && 条件式 2

逻辑或: 条件式 1 || 条件式 2

逻辑非: ! 条件式

实验三

在 LED/128*64 图形液晶试验板上实现:

按下 S1 键, 个位数数码管上显示输入值“5”; 按下 S2 键, 十位数数码管上显示输入值“8”; 若同时按下 S1、S2 键, 则除了个位、十位数数码管显示外, 千位数数码管还显示“P”。

在我的文档中建立一个文件目录 (cs15), 然后建立 cs15.uv2 的工程项目, 最后建立源程序文件 (cs15.c)。

```

#include <REG51.H> // 序号(以下同); 1
#define uchar unsigned char //2
#define uint unsigned int //3
sbit P2_0=P2^0; //4
sbit P2_1=P2^1; //5
uchar code SEG7 [10]=
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f}; //6
//=====7=====
uchar data a=0,b=0,out; //8
//-----9
uchar key_s1(void); //10
uchar key_s2(void); //11
void delay(uint k); //12
//-----13
void main(void) //14
{ //15
while(1) //16
{ //17
b= key_s1(); //18
a= key_s2(); //19
if((a==8)&&(b==5))out=1; //20

```

4. 赋值运算符与表达式

1). 简单赋值运算

在 C 语言中, 最常见的赋值运算符为“=”, 赋值运算符的作用是将一个数据的值赋给一个变量, 利用赋值运算符将一个变量与一个表达式连接起来的式子称为赋值表达式, 在赋值表达式的后面加一个分号“;”便构成了赋值语句。例如: x=5;

2). 复合赋值运算符

在赋值运算符“=”的前面加上其他运算符, 就构成了所谓复合赋值运算符。具体如下所示:

+= 加法赋值运算符

-= 减法赋值运算符

*= 乘法赋值运算符

/= 除法赋值运算符

%= 取模(取余)赋值运算符

测试

单片机系统

可靠性的简单方法

◆耿英杰

单片机系统可以分为软件和硬件两个方面,要保证单片机系统可靠性就必须从这两方面入手。

首先在设计单片机系统时,就应该充分考虑到外部的各种各样的可能干扰,尽量利用单片机提供的一切手段去割断或者解决不良外部干扰造成的影响。这里以 HOLTEK 最基本的 I/O 单片机 HT48R05A-1 为例,它内部提供了看门狗定时器 WDT 防止单片机内部程序乱跑出错;提供了低电压复位系统 LVR,当电压低于某个允许值时,单片机会自动 RESET 防止芯片被锁死;HOLTEK 也提供了最佳的外围电路连

接方案,最大可能地避免外部干扰对芯片的影响。

当一个单片机系统设计完成,对于不同的单片机系统产品会有不同的测试项目和方法,但是有一些是必须测试的:测试单片机软件功能的完善性,这是针对所有单片机系统功能的测试,测试软件是否写的正确完整。上电掉电测试,在使用中用户必然会遇到上电和掉电的情况,可以进行多次开关电源,测试单片机系统的可靠性。老化测试,测试长时间工作情况下,单片机系统的可靠性。必要的话可以放置在高温,

高压以及强电磁干扰的环境下测试。ESD 和 EFT 等测试,可以使用各种干扰模拟器来测试单片机系统的可靠性。例如使用静电模拟器测试单片机系统的抗静电 ESD 能力;使用突波杂讯模拟器进行快速脉冲抗干扰 EFT 测试等等。

当然如果没有此类条件,可以模拟人为使用中,可能发生的破坏情况。例如用人体或者衣服织物故意摩擦单片机系统的接触端口,由此测试抗静电的能力。用大功率电钻靠近单片机系统工作,由此测试抗电磁干扰能力等。

序号 6:数码管 0~9 的字形码。

序号 7:程序分隔。

序号 8:在 data 区定义无符号字符型全局变量 a,b,out。

序号 9:程序分隔。

序号 10~12:函数声明。

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:while 循环语句进行无限循环。

序号 17:while 循环语句开始。

序号 18:调用 S1 键判断子函数,其键值返回至变量 b 中。

序号 19:调用 S2 键判断子函数,其键值返回至变量 a 中。

序号 20:如果 a 为 8 且 b 为 5 同时成立(逻辑与),则 out 赋值 1。

序号 21:否则 out 赋值 0。

序号 22:变量 a 送 P0 口。

序号 23:点亮数码管十位。

序号 24:延时 2ms。

序号 25:变量 b 送 P0 口。

序号 26:点亮数码管个位。

序号 27:延时 2ms。

序号 28:如果 out 为 1,则 P0 口赋值 0x73。

序号 29:否则 P0 口赋值 0x00。

序号 30:点亮数码管千位。

序号 31:延时 2ms。

序号 32:while 循环语句结束。

序号 33:main 主函数结束。

序号 34:程序分隔。

序号 35~41:延时子函数。

序号 42:程序分隔。

序号 43:定义函数名为 key_s1 的 S1 键判断子函数。

序号 44:key_s1 子函数开始。定义局部变量 x。

序号 45:P2 口置全 1,以便读取按键状态。

序号 46:如果 P2.0 为低电平(S1 键按下),x 赋值 5。

序号 47:否则 x 赋值 0。

序号 48:返回 x 值。

序号 49:key_s1 子函数结束。

序号 50:程序分隔。

序号 51~57:函数名为 key_s2 的 S2 键判断子函数,具体可参考序号 43~49 分析。

配文优惠邮购:Keil C51 Windows 集成开发环境(已汉化光盘,

邮购代号:K1):46 元。TOP851 多功能编程器(邮购代号:B1):270 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 点阵图型液晶显示模组(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):021-64654216,13044152947,网址:<http://www.hlelectron.com>,技术支持 E-mail:zxh2151@sohu.com。