

学习单片机

//实例 1：用单片机控制第一个灯亮

```
#include<reg51.h> //包含 51 单片机寄存器定义的头文件
void main(void)
{
    P1=0xfe; //P1=1111 1110B, 即 P1.0 输出低电平
}
```

//实例 2：用单片机控制一个灯闪烁：认识单片机的工作频率

```
#include<reg51.h> //包含单片机寄存器的头文件
/
函数功能：延时一段时间
/
void delay(void) //两个 void 意思分别为无需返回值，没有参数传递
{
    unsigned int i; //定义无符号整数，最大取值范围 65535
    for(i=0;i<20000;i++) //做 20000 次空循环
        ; //什么也不做，等待一个机器周期
}
/
函数功能：主函数（C 语言规定必须有也只能有 1 个主函数）
/
void main(void)
{
    while(1) //无限循环
    {
        P1=0xfe; //P1=1111 1110B, P1.0 输出低电平
        delay(); //延时一段时间
        P1=0xff; //P1=1111 1111B, P1.0 输出高电平
        delay(); //延时一段时间
    }
}
```

//实例 3：将 P1 口状态分别送入 P0、P2、P3 口：认识 I/O 口的引脚功能

```
#include<reg51.h> //包含单片机寄存器的头文件
```

/

函数功能：主函数 （C 语言规定必须有也只能有 1 个主函数）

/

```
void main(void)
{
    while(1)          //无限循环
    {
        P1=0xff;      // P1=1111 1111B,熄灭 LED
        P0=P1;        // 将 P1 口状态送入 P0 口
        P2=P1;        // 将 P1 口状态送入 P2 口
        P3=P1;        // 将 P1 口状态送入 P3 口
    }
}
```

//实例 4：使用 P3 口流水点亮 8 位 LED

#include<reg51.h> //包含单片机寄存器的头文件

/

函数功能：延时一段时间

/

```
void delay(void)
{
    unsigned char i,j;
    for(i=0;i<250;i++)
        for(j=0;j<250;j++)
            ;
}
```

/

函数功能：主函数

/

```
void main(void)
{
    while(1)
    {
        P3=0xfe;      //第一个灯亮
        delay();       //调用延时函数
        P3=0xfd;      //第二个灯亮
        delay();       //调用延时函数
        P3=0xfb;      //第三个灯亮
        delay();       //调用延时函数
    }
}
```

```

        P3=0xf7;    //第四个灯亮
        delay();    //调用延时函数
        P3=0xef;    //第五个灯亮
        delay();    //调用延时函数
        P3=0xdf;    //第六个灯亮
        delay();    //调用延时函数
        P3=0xbf;    //第七个灯亮
        delay();    //调用延时函数
        P3=0x7f;    //第八个灯亮
        delay();    //调用延时函数
    }

}

//实例 7：通过对 P3 口地址的操作流水点亮 8 位 LED
#include<reg51.h>    //包含单片机寄存器的头文件
sfr x=0xb0;    //P3 口在存储器中的地址是 b0H， 通过 sfr 可定义 8051 内核单片机

//的所有内部 8 位特殊功能寄存器,对地址 x 的操作也就是对 P1 口的操作
/
函数功能：延时一段时间
/

void delay(void)
{
    unsigned char i,j;
    for(i=0;i<250;i++)
        for(j=0;j<250;j++)
            ;    //利用循环等待若干机器周期，从而延时一段时间
}

/
函数功能：主函数
/

void main(void)
{
    while(1)
    {
        x=0xfe;    //第一个灯亮
        delay();    //调用延时函数
        x=0xfd;    //第二个灯亮
        delay();    //调用延时函数
        x=0xfb;    //第三个灯亮
        delay();    //调用延时函数
        x=0xf7;    //第四个灯亮
    }
}

```

```

        delay(); //调用延时函数
        x=0xef;   //第五个灯亮
        delay(); //调用延时函数
        x=0xdf;   //第六个灯亮
        delay(); //调用延时函数
        x=0xbf;   //第七个灯亮
        delay(); //调用延时函数
        x=0x7f;   //第八个灯亮
        delay(); //调用延时函数
    }

}

//实例 8：用不同数据类型控制灯闪烁时间
#include<reg51.h> //包含单片机寄存器的头文件
/
函数功能：用整形数据延时一段时间
/

void int_delay(void) //延时一段较长的时间
{
    unsigned int m; //定义无符号整形变量，双字节数据，值域为 0~65535
    for(m=0;m<36000;m++)
        ; //空操作
}
/
函数功能：用字符型数据延时一段时间
/

void char_delay(void) //延时一段较短的时间
{
    unsigned char i,j; //定义无符号字符型变量，单字节数据，值域 0~255
    for(i=0;i<200;i++)
        for(j=0;j<180;j++)
            ; //空操作
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    while(1)
    {
        for(i=0;i<3;i++)
        {

```

```

        P1=0xfe;    //P1.0 口的灯点亮
        int_delay(); //延时一段较长的时间
        P1=0xff;    //熄灭
        int_delay(); //延时一段较长的时间
    }
    for(i=0;i<3;i++)
    {
        P1=0xef;    //P1.4 口的灯点亮
        char_delay(); //延时一段较长的时间
        P1=0xff;    //熄灭
        char_delay(); //延时一段较长的时间
    }

}

}

```

//实例 9：用 P0 口、P1 口分别显示加法和减法运算结果

```

#include<reg51.h>
void main(void)
{
    unsigned char m,n;
    m=43;    //即十进制数 2x16+11=43
    n=60;    //即十进制数 3x16+12=60
    P1=m+n;  //P1=103=0110 0111B,结果 P1.3、P1.4、P1.7 口的灯被点亮
    P0=n-m;  //P0=17=0001 0001B,结果 P0.0、P0.4 的灯被熄灭
}

```

//实例 10：用 P0、P1 口显示乘法运算结果

```

#include<reg51.h> //包含单片机寄存器的头文件
void main(void)
{
    unsigned char m,n;
    unsigned int s;
    m=64;

```

```

n=71;
s=m n;          //s=64 71=4544,需要 16 位二进制数表示,高 8 位送 P1 口,低 8
位送 P0 口

//由于 4544=17 256+192=H3 16 16 16+H2 16 16+H1 16+H0
//两边同除以 256,可得 17+192/256=H3 16+H2+ (H1 16+H0)
/256

//因此,高 8 位 16 进制数 H3 16+H2 必然等于 17,即 4544
除以 256 的商

//低 8 位 16 进制数 H1 16+H0 必然等于 192,即 4544 除以 256
的余数

P1=s/256;        //高 8 位送 P1 口 , P1=17=11H=0001 0001B, P1.0 和 P1.4 口灭,
其余亮
P0=s%256;        //低 8 位送 P0 口 , P3=192=c0H=1100 0000B,P3.1,P3.6,P3.7 口
灭,其余亮
}

```

//实例 11：用 P1、P0 口显示除法运算结果

```

#include<reg51.h>  //包含单片机寄存器的头文件
void main(void)
{
    P1=36/5;        //求整数
    P0=((36%5) 10)/5; //求小数
    while(1)
        ;           //无限循环防止程序“跑飞”
}

```

//实例 12：用自增运算控制 P0 口 8 位 LED 流水花样

```

#include<reg51.h>  //包含单片机寄存器的头文件
/
函数功能：延时一段时间
/

void delay(void)
{
    unsigned int i;
    for(i=0;i<20000;i++)

```

```

        ;
    }
/
函数功能   ： 主函数
/

void main(void)
{
    unsigned char i;
    for(i=0;i<255;i++) //注意 i 的值不能超过 255
    {
        P0=i;    //将 i 的值送 P0 口
        delay(); //调用延时函数
    }
}

```

//实例 13：用 P0 口显示逻辑"与"运算结果

```

#include<reg51.h> //包含单片机寄存器的头文件
void main(void)
{
    P0=(4>0)&&(9>0xab); //将逻辑运算结果送 P0 口
    while(1)
        ; //设置无限循环，防止程序“跑飞”
}

```

//实例 14：用 P0 口显示条件运算结果

```

#include<reg51.h> //包含单片机寄存器的头文件
void main(void)
{
    P0=(8>4)?8:4; //将条件运算结果送 P0 口，P0=8=0000 1000B
    while(1)
        ; //设置无限循环，防止程序“跑飞”
}

```

//实例 15：用 P0 口显示按位"异或"运算结果

```
#include<reg51.h>    //包含单片机寄存器的头文件
void main(void)
{
    P0=0xa2^0x3c;//将条件运算结果送 P0 口，P0=8=0000 1000B
    while(1)
        ;    //设置无限循环，防止程序“跑飞”
}
```

//实例 16：用 P0 显示左移运算结果

```
#include<reg51.h>    //包含单片机寄存器的头文件
void main(void)
{
    P0=0x3b<<2;//将左移运算结果送 P0 口，P0=1110 1100B=0xec
    while(1)
        ;    //无限循环，防止程序“跑飞”
}
```

//实例 17：“万能逻辑电路”实验

```
#include<reg51.h>    //包含单片机寄存器的头文件
sbit F=P1^4;          //将 F 位定义为 P1.4
sbit X=P1^5;          //将 X 位定义为 P1.5
sbit Y=P1^6;          //将 Y 位定义为 P1.6
sbit Z=P1^7;          //将 Z 位定义为 P1.7
void main(void)
{
    while(1)
    {
        F=((~X)&Y)|Z; //将逻辑运算结果赋给 F
        ;
    }
}
```

//实例 18：用右移运算流水点亮 P1 口 8 位 LED

```
#include<reg51.h>    //包含单片机寄存器的头文件
/
```


函数功能：延时一段时间

```

/
void delay(void)
{
    unsigned int n;
    for(n=0;n<30000;n++)
        ;
}
/
```

函数功能：主函数

```

/
void main(void)
{
    unsigned char i;
    while(1)
    {
        P1=0xff;
        delay();
        for(i=0;i<8;i++)//设置循环次数为 8
        {
            P1=P1>>1;    //每次循环 P1 的各二进位右移 1 位，高位补 0
            delay();      //调用延时函数
        }
    }
}
}
```

//实例 19：用 if 语句控制 P0 口 8 位 LED 的流水方向

#include<reg51.h> //包含单片机寄存器的头文件

sbit S1=P1^4; //将 S1 位定义为 P1.4

sbit S2=P1^5; //将 S2 位定义为 P1.5

/

函数功能：主函数

```

/
void main(void)
{
    while(1)
    {
        if(S1==0) //如果按键 S1 按下
            P0=0x0f; //P0 口高四位 LED 点亮
        if(S2==0) //如果按键 S2 按下

```

```

        P0=0xf0; //P0 口低四位 LED 点亮
    }
}

```

//实例 20：用 switch 语句的控制 P0 口 8 位 LED 的点亮状态

```

#include<reg51.h> //包含单片机寄存器的头文件
sbit S1=P1^4;     //将 S1 位定义为 P1.4
/
函数功能：延时一段时间
/

void delay(void)
{
    unsigned int n;
    for(n=0;n<10000;n++)
        ;
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    i=0; //将 i 初始化为 0
    while(1)
    {
        if(S1==0) //如果 S1 键按下
        {
            delay(); //延时一段时间
            if(S1==0) //如果再次检测到 S1 键按下
                i++; //i 自增 1
            if(i==9) //如果 i=9，重新将其置为 1
                i=1;
        }
        switch(i) //使用多分支选择语句
        {
            case 1: P0=0xfe; //第一个 LED 亮
                    break;
            case 2: P0=0xfd; //第二个 LED 亮
                    break;

```

```

        case 3:P0=0xfb;    //第三个 LED 亮
            break;
        case 4:P0=0xf7;    //第四个 LED 亮
            break;
        case 5:P0=0xef;    //第五个 LED 亮
            break;
        case 6:P0=0xdf;    //第六个 LED 亮
            break;
        case 7:P0=0xbf;    //第七个 LED 亮
            break;
        case 8:P0=0x7f;    //第八个 LED 亮
            break;
        default:    //缺省值，关闭所有 LED
            P0=0xff;
    }
}
}

```

//实例 21：用 for 语句控制蜂鸣器鸣笛次数

```
#include<reg51.h>    //包含单片机寄存器的头文件
```

```
sbit sound=P3^7;    //将 sound 位定义为 P3.7
```

```
/
```

```
函数功能：延时形成 1600Hz 音频
```

```
/
```

```
void delay1600(void)
```

```
{
```

```
    unsigned char n;
```

```
    for(n=0;n<100;n++)
```

```
        ;
```

```
}
```

```
/
```

```
函数功能：延时形成 800Hz 音频
```

```
/
```

```
void delay800(void)
```

```
{
```

```
    unsigned char n;
```

```
    for(n=0;n<200;n++)
```

```
        ;
```

```
}
```

```
/
```

函数功能：主函数

/

```
void main(void)
{
    unsigned int i;
    while(1)
    {
        for(i=0;i<830;i++)
        {
            sound=0; //P3.7 输出低电平
            delay1600();
            sound=1; //P3.7 输出高电平
            delay1600();
        }
        for(i=0;i<200;i++)
        {
            sound=0; //P3.7 输出低电平
            delay800();
            sound=1; //P3.7 输出高电平
            delay800();
        }
    }
}
```

//实例 22：用 while 语句控制 LED

#include<reg51.h> //包含单片机寄存器的头文件

/

函数功能：延时约 60ms (3 100 200=60000 μ s)

/

```
void delay60ms(void)
{
    unsigned char m,n;
    for(m=0;m<100;m++)
        for(n=0;n<200;n++)
            ;
}
/
```

函数功能：主函数

```

/
void main(void)
{
    unsigned char i;
    while(1)    //无限循环
    {
        i=0;    //将 i 初始化为 0
        while(i<0xff)    //当 i 小于 0xff (255)时执行循环体
        {
            P0=i;        //将 i 送 P0 口显示
            delay60ms(); //延时
            i++;          //i 自增 1
        }
    }
}

```

//实例 23：用 **do-while** 语句控制 P0 口 8 位 LED 流水点亮

#include<reg51.h> //包含单片机寄存器的头文件

/

函数功能：延时约 60ms (3 100 200=60000 μs)

/

```

void delay60ms(void)
{
    unsigned char m,n;
    for(m=0;m<100;m++)
        for(n=0;n<200;n++)
            ;
}

```

/

函数功能：主函数

```

/
void main(void)
{
    do
    {
        P0=0xfe;    //第一个 LED 亮
        delay60ms();
        P0=0xfd;    //第二个 LED 亮
        delay60ms();
        P0=0xfb;    //第三个 LED 亮
        delay60ms();
    }
}

```

```

        P0=0xf7;    //第四个 LED 亮
        delay60ms();
        P0=0xef;    //第五个 LED 亮
        delay60ms();
        P0=0xdf;    //第六个 LED 亮
        delay60ms();
        delay60ms();
        P0=0xbf;    //第七个 LED 亮
        delay60ms();
        P0=0x7f;    //第八个 LED 亮
        delay60ms();
    }while(1);    //无限循环，使 8 位 LED 循环流水点亮
}

```

//实例 24：用字符型数组控制 P0 口 8 位 LED 流水点亮

```

#include<reg51.h> //包含单片机寄存器的头文件
/
函数功能：延时约 60ms (3 100 200=60000 μs)
/

void delay60ms(void)
{
    unsigned char m,n;
    for(m=0;m<100;m++)
        for(n=0;n<200;n++)
            ;
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    unsigned char code Tab[ ]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f}; //定义无符号字符型数组
    while(1)
    {
        for(i=0;i<8;i++)
        {
            P0=Tab[i]; //依次引用数组元素，并将其送 P0 口显示
            delay60ms(); //调用延时函数
        }
    }
}

```

```

    }
}

```

//实例 25： 用 P0 口显示字符串常量

```

#include<reg51.h> //包含单片机寄存器的头文件
/
函数功能： 延时约 150ms (3 200 250=150 000 μ s=150ms
/

void delay150ms(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
/
函数功能： 主函数
/

void main(void)
{
    unsigned char str[]{"Now,Temperature is :"}; //将字符串赋给字符型全部元素
赋值
    unsigned char i;
    while(1)
    {
        i=0; //将 i 初始化为 0，从第一个元素开始显示
        while(str[i]!='\0') //只要没有显示到结束标志'\0'
        {
            P0=str[i]; //将第 i 个字符送到 P0 口显示
            delay150ms(); //调用 150ms 延时函数
            i++; //指向下一个待显字符
        }
    }
}

```

//实例 26： 用 P0 口显示指针运算结果

```

#include<reg51.h>
void main(void)

```

```

{
    unsigned char  p1, p2; //定义无符号字符型指针变量 p1,p2
    unsigned char i,j;    //定义无符号字符型数据
    i=25;                //给 i 赋初值 25
    j=15;
    p1=&i;               //使指针变量指向 i ， 对指针初始化
    p2=&j;               //使指针变量指向 j ， 对指针初始化
    P0= p1+ p2;         // p1+ p2 相当于 i+j,所以 P0=25+15=40=0x28
                        //则 P0=0010 1000B，结果 P0.3、P0.5 引脚 LED 熄灭，其余点亮
    while(1)
        ;                //无限循环，防止程序“跑飞”
}

```

//实例 27：用指针数组控制 P0 口 8 位 LED 流水点亮

```

#include<reg51.h>
/
函数功能：延时约 150ms (3 200 250=150 000 μ s=150ms
/

void delay150ms(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char code Tab[]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
    unsigned char  p[ ]={&Tab[0],&Tab[1],&Tab[2],&Tab[3],&Tab[4],&Tab[5],
                        &Tab[6],&Tab[7]};

    unsigned char i;        //定义无符号字符型数据
    while(1)
    {
        for(i=0;i<8;i++)
        {
            P0= p[i];

```



```

        delay150ms();
    }
}

```

//实例 28：用数组的指针控制 P0 口 8 位 LED 流水点亮

```

#include<reg51.h>
/
函数功能：延时约 150ms (3 200 250=150 000 μ s=150ms
/

void delay150ms(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    unsigned char Tab[ ]={0xFF,0xFE,0xFD,0xFB,0xF7,0xEF,0xDF,0xBF,
                           0x7F,0xBF,0xDF,0xEF,0xF7,0xFB,0xFD,0xFE,
                           0xFE,0xFC,0xFB,0xF0,0xE0,0xC0,0x80,0x00,
                           0xE7,0xDB,0xBD,0x7E,0x3C,0x18,0x00,0x81,
                           0xC3,0xE7,0x7E,0xBD,0xDB,0xE7,0xBD,0xDB};
                           //流水灯控制码
    unsigned char  p; //定义无符号字符型指针
    p=Tab;           //将数组首地址存入指针 p
    while(1)
    {
        for(i=0;i<32;i++) //共 32 个流水灯控制码
        {
            P0= (p+i);    // (p+i)的值等于 a[i]
            delay150ms(); //调用 150ms 延时函数
        }
    }
}

```

//实例 29：用 P0 、 P1 口显示整型函数返回值

```
#include<reg51.h>
/
函数功能： 计算两个无符号整数的和
/

unsigned int sum(int a,int b)
{
    unsigned int s;
    s=a+b;
    return (s);
}
/
函数功能： 主函数
/

void main(void)
{
    unsigned z;
    z=sum(2008,2009);
    P1=z/256;      //取得 z 的高 8 位
    P0=z%256;      //取得 z 的低 8 位
    while(1)
        ;
}
```

//实例 30：用有参函数控制 P0 口 8 位 LED 流水速度

```
#include<reg51.h>
/
函数功能： 延时一段时间
/

void delay(unsigned char x)
{
    unsigned char m,n;
    for(m=0;m<x;m++)
        for(n=0;n<200;n++)
            ;
}
/
```

函数功能：主函数

```
void main(void)
{
    unsigned char i;
    unsigned char code Tab[] = {0xFE, 0xFD, 0xFB, 0xF7, 0xEF, 0xDF, 0xBF, 0x7F};
    //流水灯控制码

    while(1)
    {
        //快速流水点亮 LED
        for(i=0; i<8; i++) //共 8 个流水灯控制码
        {
            P0=Tab[i];
            delay(100); //延时约 60ms, (3 100 200=60 000 μs)
        }
        //慢速流水点亮 LED
        for(i=0; i<8; i++) //共 8 个流水灯控制码
        {
            P0=Tab[i];
            delay(250); //延时约 150ms, (3 250 200=150 000 μs)
        }
    }
}
```

//实例 31：用数组作函数参数控制流水花样

#include<reg51.h>

/

函数功能：延时约 150ms

```
void delay(void)
```

```
{
```

```
    unsigned char m,n;
```

```
    for(m=0;m<200;m++)
```

```
        for(n=0;n<250;n++)
```

```
            ;
```

```
}
```

/

函数功能：流水点亮 P0 口 8 位 LED

```
void led_flow(unsigned char a[8])
```

```
{
```

```

    unsigned char i;
    for(i=0;i<8;i++)
    {
        P0=a[i];
        delay();
    }
}

/
函数功能：主函数
/

void main(void)
{
    unsigned char code Tab[ ]={0xFE,0xFD,0xFB,0xF7,0xEF,0xDF,0xBF,0x7F};
                                //流水灯控制码
    led_flow(Tab);
}

```

//实例 32：用指针作函数参数控制 P0 口 8 位 LED 流水点亮

```

#include<reg51.h>
/
函数功能：延时约 150ms
/

void delay(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
/
函数功能：流水点亮 P0 口 8 位 LED
/

void led_flow(unsigned char *p) //形参为无符号字符型指针
{
    unsigned char i;
    while(1)
    {

```

```

    i=0;    //将 i 置为 0，指向数组第一个元素
    while( (p+i)!='\0') //只要没有指向数组的结束标志
    {
        P0=(p+i); // 取的指针所指变量（数组元素）的值，送 P0 口
        delay();  //调用延时函数
        i++;      //指向下一个数组元素
    }
}

/
函数功能：主函数

void main(void)
{
    unsigned char code Tab[ ]={0xFE,0xFD,0xFB,0xF7,0xEF,0xDF,0xBF,0x7F,
                                0x7F,0xBF,0xDF,0xEF,0xF7,0xFB,0xFD,0xFE,
                                0xFF,0xFE,0xFC,0xFB,0xF0,0xE0,0xC0,0x80,
                                0x00,0xE7,0xDB,0xBD,0x7E,0xFF,0xFF,0x3C,
                                0x18,0x00,0x81,0xC3,0xE7,0xFF,
                                0xFF,0x7E};
                                //流水灯控制码

    unsigned char pointer;
    pointer=Tab;
    led_flow(pointer);
}

```

//实例 33：用函数型指针控制 P1 口灯花样

```

#include<reg51.h> //包含 51 单片机寄存器定义的头文件
unsigned char code Tab[]={0xFE,0xFD,0xFB,0xF7,0xEF,0xDF,0xBF,0x7F};
//流水灯控制码，该数组被定义为全局变量
/
函数功能：延时约 150ms

void delay(void)
{

```

```

        unsigned char m,n;
        for(m=0;m<200;m++)
            for(n=0;n<250;n++)
                ;
    }
/
函数功能：流水灯左移
/

void led_flow(void)
{
    unsigned char i;
    for(i=0;i<8;i++) //8 位控制码
    {
        P0=Tab[i];
        delay();
    }

}
/
函数功能：主函数
/

void main(void)
{
    void ( p)(void); //定义函数型指针，所指函数无参数，无返回值
    p=led_flow; //将函数的入口地址赋给函数型指针 p
    while(1)
        ( p)(); //通过函数的指针 p 调用函数 led_flow ( )
}

```

//实例 34：用指针数组作为函数的参数显示多个字符串

```

#include<reg51.h> //包含 51 单片机寄存器定义的头文件
unsigned char code str1[ ]="Temperature is tested by DS18B20";//C 语言中，字符串
是作为字符数组来处理的
unsigned char code str2[ ]="Now temperature is:"; //所以，字符串的名字就是
字符串的首地址
unsigned char code str3[ ]="The System is designed by Zhang San";
unsigned char code str4[ ]="The date is 2008-9-30";
unsigned char p[ ]={str1,str2,str3,str4}; //定义 p[4]为指向 4 个字符串的字符型指
针数组
/

```

函数功能：延时约 150ms

```
void delay(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
```

函数功能：流水点亮 P0 口 8 位 LED

```
void led_display(unsigned char x[]) //形参必须为指针数组
{
    unsigned char i,j;
    for(i=0;i<4;i++) //有 4 个字符串要显示
    {
        j=0; //指向待显字符串的第 0 号元素
        while( (x[i]+j)!='\0') //只要第 i 个字符串的第 j 号元素不是结束标志
        {
            P0= (x[i]+j); //取得该元素值送到 P0 口显示
            delay(); //调用延时函数
            j++; //指向下一个元素
        }
    }
}
```

函数功能：主函数

```
void main(void)
{
    unsigned char i;
    while(1)
    {
        for(i=0;i<4;i++)
            led_display(p); //将指针数组名作实际参数传递
    }
}
```

//实例 35：字符函数 ctype.h 应用举例

```
#include<reg51.h> //包含 51 单片机寄存器定义的头文件
```

```

#include<ctype.h>
void main(void)
{
    while(1)
    {
        P3=isalpha('_')?0xf0:0x0f;//条件运算，若'_'是英文字母，P3=0xf0
    }
}

```

//实例 36：内部函数 **intrins.h** 应用举例

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
#include<intrins.h>  //包含函数 isalpha（）声明的头文件
/
函数功能：延时约 150ms
/

void delay(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
/
函数功能：主函数
/

void main(void)
{
    P3=0xfe; //P3=1111 1110B
while(1)
{
    P3=_crol_(P3,1);// 将 P3 的二进制位循环左移 1 位后再赋给 P3
    delay(); //调用延时函数
}
}

```

//实例 37：标准函数 **stdlib.h** 应用举例


```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
#include<stdlib.h>    //包含函数 isalpha ( ) 声明的头文件
/
函数功能：延时约 150ms
/

void delay(void)
{
    unsigned char m,n;
    for(m=0;m<200;m++)
        for(n=0;n<250;n++)
            ;
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    while(1)
    {
        for(i=0;i<10;i++) //产生 10 个随机数
        {
            P3=rand()/160; //将产生的随机数缩小 160 倍后送 P3 显示
            delay();
        }
    }
}

```

//实例 38：字符串函数 string.h 应用举例

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
#include<string.h>    //包含函数 isalpha ( ) 声明的头文件
void main(void)
{
    unsigned char str1[ ]="Now, The temperature is :";
    unsigned char str2[ ]="Now, The temperature is 36 Centgrade:";
    unsigned char i;
    i=strcmp(str1,str2); //比较两个字符串，并将结果存入 i
    if(i==0) //str1=str2
        P3=0x00;
    else

```

```

    if(i<0)    //str1<str2
        P3=0xf0;
    else      //str1>str2
        P3=0x0f;
while(1)
    ;    //防止程序“跑飞”
}

```

//实例 39：宏定义应用举例 2

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
#define F(a,b) (a)+(a) (b)/256+(b)    //带参数的宏定义，a 和 b 为形参
void main(void)
{
    unsigned char i,j,k;
    i=40;
    j=30;
    k=20;
    P3=F(i,j+k);    //i 和 j+k 分别为实参，宏展开时，实参将替代宏定义中的形参
    while(1)
        ;
}

```

//实例 40：宏定义应用举例 2

```

#include<AT89X51.h>
#include<ctype.h>
void main(void)
{
    P3_0=0;    //将 P3.0 引脚置低电平，LED 点亮
    P3_1=0;    //将 P3.0 引脚置低电平，LED 点亮
    P3_2=0;    //将 P3.0 引脚置低电平，LED 点亮
    P3_3=0;    //将 P3.0 引脚置低电平，LED 点亮
    P3_4=1;    //将 P3.4 引脚置高电平，LED 熄灭
    P3_5=1;    //将 P3.5 引脚置高电平，LED 熄灭
    P3_6=1;    //将 P3.7 引脚置高电平，LED 熄灭
}

```

```

P3_7=1; //将 P3.7 引脚置高电平，LED 熄灭
while(1)
    ;
}

```

//实例 41：宏定义应用举例 3

```

#include<reg51.h> //包含 51 单片机寄存器定义的头文件
#define MAX 100 //将 MAX 宏定义为字符串 100
void main(void)
{
    #if MAX>80 //如果字符串 100 大于 80
        P3=0xf0; //P3 口低四位 LED 点亮
    #else
        P3=0x0f; //否则，P3 口高四位 LED 点亮
    #endif //结束本次编译
}

```

/

中断、定时器

中断、定时器

中断、定时器

中断、定时器

/

//实例 42：用定时器 T0 查询方式 P2 口 8 位控制 LED 闪烁

```

#include<reg51.h> // 包含 51 单片机寄存器定义的头文件
/
函数功能：主函数
/

void main(void)

```

```

{
    // EA=1;                //开总中断
    // ET0=1;                //定时器 T0 中断允许
    TMOD=0x01;              //使用定时器 T0 的模式 1
    TH0=(65536-46083)/256;   //定时器 T0 的高 8 位赋初值
    TL0=(65536-46083)%256;   //定时器 T0 的高 8 位赋初值
    TR0=1;                  //启动定时器 T0
    TF0=0;
    P2=0xff;
    while(1)//无限循环等待查询
    {
        while(TF0==0)
            ;
        TF0=0;
        P2=~P2;
        TH0=(65536-46083)/256; //定时器 T0 的高 8 位赋初值
        TL0=(65536-46083)%256; //定时器 T0 的高 8 位赋初值
    }
}

```

//实例 43：用定时器 T1 查询方式控制单片机发出 1KHz 音频

```

#include<reg51.h>          // 包含 51 单片机寄存器定义的头文件
sbit sound=P3^7;           //将 sound 位定义为 P3.7 引脚
/
函数功能：主函数

```

```

void main(void)
{
    // EA=1;                //开总中断
    // ET0=1;                //定时器 T0 中断允许
    TMOD=0x10;              //使用定时器 T1 的模式 1
    TH1=(65536-921)/256;    //定时器 T1 的高 8 位赋初值
    TL1=(65536-921)%256;    //定时器 T1 的高 8 位赋初值
    TR1=1;                  //启动定时器 T1
    TF1=0;
    while(1)//无限循环等待查询
    {
        while(TF1==0)
            ;
        TF1=0;
    }
}

```

```

        sound=~sound; //将 P3.7 引脚输出电平取反
        TH1=(65536-921)/256; //定时器 T0 的高 8 位赋初值
        TL1=(65536-921)%256; //定时器 T0 的高 8 位赋初值
    }
}

```

//实例 44：将计数器 T0 计数的结果送 P1 口 8 位 LED 显示

```

#include<reg51.h> // 包含 51 单片机寄存器定义的头文件
sbit S=P3^4; //将 S 位定义为 P3.4 引脚
/
函数功能：主函数
/

void main(void)
{
    // EA=1; //开总中断
    // ET0=1; //定时器 T0 中断允许
    TMOD=0x02; //使用定时器 T0 的模式 2
    TH0=256-156; //定时器 T0 的高 8 位赋初值
    TL0=256-156; //定时器 T0 的高 8 位赋初值
    TR0=1; //启动定时器 T0
    while(1)//无限循环等待查询
    {
        while(TF0==0) //如果未计满就等待
        {
            if(S==0) //按键 S 按下接地，电平为 0
                P1=TL0; //计数器 TL0 加 1 后送 P1 口显示
        }
        TF0=0; //计数器溢出后，将 TF0 清 0
    }
}

```

//实例 45：用定时器 T0 的中断控制 1 位 LED 闪烁

```

#include<reg51.h> // 包含 51 单片机寄存器定义的头文件
sbit D1=P2^0; //将 D1 位定义为 P2.0 引脚
/

```

函数功能：主函数

```
void main(void)
{
    EA=1;                //开总中断
    ET0=1;               //定时器 T0 中断允许
    TMOD=0x01;           //使用定时器 T0 的模式 2
    TH0=(65536-46083)/256; //定时器 T0 的高 8 位赋初值
    TL0=(65536-46083)%256; //定时器 T0 的高 8 位赋初值
    TR0=1;               //启动定时器 T0
    while(1)//无限循环等待中断
        ;
}
```

函数功能：定时器 T0 的中断服务程序

```
void Time0(void) interrupt 1 using 0 // “interrupt” 声明函数为中断服务函数
//其后的 1 为定时器 T0 的中断编号；0 表示使用第 0 组工作寄存器
{
    D1=~D1; //按位取反操作，将 P2.0 引脚输出电平取反
    TH0=(65536-46083)/256; //定时器 T0 的高 8 位重新赋初值
    TL0=(65536-46083)%256; //定时器 T0 的高 8 位重新赋初值
}
```

//实例 46：用定时器 T0 的中断实现长时间定时

```
#include<reg51.h> // 包含 51 单片机寄存器定义的头文件
sbit D1=P2^0; //将 D1 位定义为 P2.0 引脚
unsigned char Countor; //设置全局变量，储存定时器 T0 中断次数
/
```

函数功能：主函数

```
void main(void)
{
    EA=1;                //开总中断
    ET0=1;               //定时器 T0 中断允许
    TMOD=0x01;           //使用定时器 T0 的模式 2
    TH0=(65536-46083)/256; //定时器 T0 的高 8 位赋初值
    TL0=(65536-46083)%256; //定时器 T0 的高 8 位赋初值
    TR0=1;               //启动定时器 T0
```