

```

    Countor=0;           //从 0 开始累计中断次数
    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
{
    Countor++;    //中断次数自加 1
    if(Countor==20) //若累计满 20 次，即计时满 1s
    {
        D1=~D1;    //按位取反操作，将 P2.0 引脚输出电平取反
        Countor=0; //将 Countor 清 0，重新从 0 开始计数
    }
    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 47：用定时器 T1 中断控制两个 LED 以不同周期闪烁

```

#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
sbit D2=P2^1; //将 D2 位定义为 P2.1 引脚
unsigned char Countor1; //设置全局变量，储存定时器 T1 中断次数
unsigned char Countor2; //设置全局变量，储存定时器 T1 中断次数
/
函数功能：主函数
/
void main(void)
{
    EA=1;           //开总中断
    ET1=1;          //定时器 T1 中断允许
    TMOD=0x10;      //使用定时器 T1 的模式 1
    TH1=(65536-46083)/256; //定时器 T1 的高 8 位赋初值
    TL1=(65536-46083)%256; //定时器 T1 的高 8 位赋初值
    TR1=1;          //启动定时器 T1
    Countor1=0;      //从 0 开始累计中断次数
    Countor2=0;      //从 0 开始累计中断次数
}

```

```

        while(1)//无限循环等待中断
        ;
    }
/
函数功能：定时器 T1 的中断服务程序
/
void Time1(void) interrupt 3 using 0 // “interrupt” 声明函数为中断服务函数
    //其后的 3 为定时器 T1 的中断编号；0 表示使用第 0 组工作
寄存器
{
    Countor1++; //Countor1 自加 1
    Countor2++; //Countor2 自加 1
    if(Countor1==2) //若累计满 2 次，即计时满 100ms
    {
        D1=~D1; //按位取反操作，将 P2.0 引脚输出电平取反
        Countor1=0; //将 Countor1 清 0，重新从 0 开始计数
    }
    if(Countor2==8) //若累计满 8 次，即计时满 400ms
    {
        D2=~D2; //按位取反操作，将 P2.1 引脚输出电平取反
        Countor2=0; //将 Countor1 清 0，重新从 0 开始计数
    }
    TH1=(65536-46083)/256; //定时器 T1 的高 8 位重新赋初值
    TL1=(65536-46083)%256; //定时器 T1 的高 8 位重新赋初值
}

```

### **//实例 48：用计数器 T1 的中断控制蜂鸣器发出 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
    ET1=1; //定时器 T1 中断允许
    TMOD=0x10; //TMOD=0001 000B，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
}

```

```

        while(1)//无限循环等待中断
        ;
    }
/
函数功能：定时器 T1 的中断服务程序
/
void Time1(void) interrupt 3 using 0 // “interrupt” 声明函数为中断服务函数
{
    sound=~sound;
    TH1=(65536-921)/256;    //定时器 T1 的高 8 位重新赋初值
    TL1=(65536-921)%256;    //定时器 T1 的高 8 位重新赋初值
}

```

### //实例 49：用定时器 T0 的中断实现"渴望"主题曲的播放

```

#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
unsigned int C;    //储存定时器的定时常数
//以下是 C 调低音的音频宏定义
#define l_dao 262    //将 “l_dao” 宏定义为低音 “1” 的频率 262Hz
#define l_re 286    //将 “l_re” 宏定义为低音 “2” 的频率 286Hz
#define l_mi 311    //将 “l_mi” 宏定义为低音 “3” 的频率 311Hz
#define l_fa 349    //将 “l_fa” 宏定义为低音 “4” 的频率 349Hz
#define l_sao 392    //将 “l_sao” 宏定义为低音 “5” 的频率 392Hz
#define l_la 440    //将 “l_a” 宏定义为低音 “6” 的频率 440Hz
#define l_xi 494    //将 “l_xi” 宏定义为低音 “7” 的频率 494Hz
//以下是 C 调中音的音频宏定义
#define dao 523    //将 “dao” 宏定义为中音 “1” 的频率 523Hz
#define re 587    //将 “re” 宏定义为中音 “2” 的频率 587Hz
#define mi 659    //将 “mi” 宏定义为中音 “3” 的频率 659Hz
#define fa 698    //将 “fa” 宏定义为中音 “4” 的频率 698Hz
#define sao 784    //将 “sao” 宏定义为中音 “5” 的频率 784Hz
#define la 880    //将 “la” 宏定义为中音 “6” 的频率 880Hz
#define xi 987    //将 “xi” 宏定义为中音 “7” 的频率 987Hz
//以下是 C 调高音的音频宏定义
#define h_dao 1046    //将 “h_dao” 宏定义为高音 “1” 的频率 1046Hz
#define h_re 1174    //将 “h_re” 宏定义为高音 “2” 的频率 1174Hz
#define h_mi 1318    //将 “h_mi” 宏定义为高音 “3” 的频率 1318Hz
#define h_fa 1396    //将 “h_fa” 宏定义为高音 “4” 的频率 1396Hz
#define h_sao 1567    //将 “h_sao” 宏定义为高音 “5” 的频率 1567Hz
#define h_la 1760    //将 “h_la” 宏定义为高音 “6” 的频率 1760Hz

```

```

#define h_xi 1975      //将“h_xi”宏定义为高音“7”的频率 1975Hz
/
函数功能：1 个延时单位，延时 200ms
/

void delay()
{
    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)
{
    unsigned char i,j;
//以下是《渴望》片头曲的一段简谱
    unsigned int code f[]={re,mi,re,dao,l_la,dao,l_la,    //每行对应一小节音符
        l_sao,l_mi,l_sao,l_la,dao,
        l_la,dao,sao,la,mi,sao,
        re,
        mi,re,mi,sao,mi,
        l_sao,l_mi,l_sao,l_la,dao,
        l_la,l_la,dao,l_la,l_sao,l_re,l_mi,
        l_sao,
        re,re,sao,la,sao,
        fa,mi,sao,mi,
        la,sao,mi,re,mi,l_la,dao,
        re,
        mi,re,mi,sao,mi,
        l_sao,l_mi,l_sao,l_la,dao,
        l_la,dao,re,l_la,dao,re,mi,
        re,
        l_la,dao,re,l_la,dao,re,mi,
        re,
        0xff}; //以 0xff 作为音符的结束标志

//以下是简谱中每个音符的节拍
//“4”对应 4 个延时单位，“2”对应 2 个延时单位，“1”对应 1 个延时单位
unsigned char code JP[ ]={4,1,1,4,1,1,2,
    2,2,2,2,8,
    4,2,3,1,2,2,
    10,

```

```

        4,2,2,4,4,
        2,2,2,2,4,
        2,2,2,2,2,2,2,
        10,
        4,4,4,2,2,
        4,2,4,4,
        4,2,2,2,2,2,2,
        10,
        4,2,2,4,4,
        2,2,2,2,6,
        4,2,2,4,1,1,4,
        10,
        4,2,2,4,1,1,4,
        10
    };
    EA=1;          //开总中断
    ET0=1;         //定时器 T0 中断允许
    TMOD=0x00;     // 使用定时器 T0 的模式 1（13 位计数器）
    while(1)       //无限循环
    {
        i=0;       //从第 1 个音符 f[0]开始播放
        while(f[i]!=0xff) //只要没有读到结束标志就继续播放
        {
            C=460830/f[i];
            TH0=(8192-C)/32; //可证明这是 13 位计数器 TH0 高 8 位的赋
初值方法
            TL0=(8192-C)%32; //可证明这是 13 位计数器 TL0 低 5 位的赋初
值方法

            TR0=1;          //启动定时器 T0
            for(j=0;j<JP[i];j++) //控制节拍数
                delay();      //延时 1 个节拍单位
            TR0=0;          //关闭定时器 T0
            i++;             //播放下一个音符
        }
    }
}
/
函数功能：定时器 T0 的中断服务子程序，使 P3.7 引脚输出音频的方波
/

void Time0(void ) interrupt 1 using 1
{
    sound=!sound;          //将 P3.7 引脚输出电平取反，形成方波
    TH0=(8192-C)/32;       //可证明这是 13 位计数器 TH0 高 8 位的赋初值方法
    TL0=(8192-C)%32;       //可证明这是 13 位计数器 TL0 低 5 位的赋初值方法

```

```
}
```

### **//实例 50-1：输出 50 个矩形脉冲**

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

```
sbit u=P1^4;    //将 u 位定义为 P1.4
```

```
/
```

```
函数功能：延时约 30ms (3 100 100=30 000  $\mu$  s =30m
```

```
/
```

```
void delay30ms(void)
```

```
{
```

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

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

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

```
            ;
```

```
}
```

```
/
```

```
函数功能：主函数
```

```
/
```

```
void main(void)
```

```
{
```

```
    unsigned char i;
```

```
    u=1;    //初始化输出高电平
```

```
    for(i=0;i<50;i++) //输出 50 个矩形脉冲
```

```
    {
```

```
        u=1;
```

```
        delay30ms();
```

```
        u=0;
```

```
        delay30ms();
```

```
    }
```

```
    while(1)
```

```
        ;//无限循环，防止程序“跑飞”
```

```
}
```

### **//实例 50-2：计数器 T0 统计外部脉冲数**

```
#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)
{
    TMOD=0x06;    // TMOD=0000 0110B,使用计数器 T0 的模式 2
    EA=1;         //开总中断
    ET0=0;        //不使用定时器 T0 的中断
    TR0=1;        //启动 T0
    TH0=0;        //计数器 T0 高 8 位赋初值
    TL0=0;        //计数器 T0 低 8 位赋初值
    while(1)      //无限循环，不停地将 TL0 计数结果送 P1 口
        P1=TL0;
}

```

## //实例 51-2：定时器 T0 的模式 2 测量正脉冲宽度

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
sbit ui=P3^2;    //将 ui 位定义为 P3.0（INT0）引脚，表示输入电压
/
函数功能：主函数
/

void main(void)
{
    TMOD=0x0a;    // TMOD=0000 1010B,使用定时器 T0 的模式 2，GATE 置 1
    EA=1;         //开总中断
    ET0=0;        //不使用定时器 T0 的中断
    TR0=1;        //启动 T0
    TH0=0;        //计数器 T0 高 8 位赋初值
    TL0=0;        //计数器 T0 低 8 位赋初值
    while(1)      //无限循环，不停地将 TL0 计数结果送 P1 口
    {
        while(ui==0)    //INT0 为低电平，T0 不能启动
            ;
        TL0=0;          //INT0 为高电平，启动 T0 计时，所以将 TL0 清 0
        while(ui==1)    //在 INT0 高电平期间，等待，计时
            ;
        P1=TL0;        //将计时结果送 P1 口显示
    }
}

```

## //实例 52：用定时器 T0 控制输出高低宽度不同的矩形波

```
#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
sbit u=P3^0;         //将 u 位定义为 P3.0，从该引脚输出矩形脉冲
unsigned char Countor; //设置全局变量，储存负跳变累计数
```

```
/
函数功能：延时约 30ms (3 100 100=30 000  $\mu$  s =30ms)
```

```
/
```

```
void delay30ms(void)
{
    unsigned char m,n;
    for(m=0;m<100;m++)
        for(n=0;n<100;n++)
            ;
}
```

```
/
函数功能：主函数
```

```
/
```

```
void main(void)
{
    unsigned char i;
    EA=1;    //开放总中断
    EX0=1;   //允许使用外中断
    IT0=1;   //选择负跳变来触发外中断
    Countor=0;
    for(i=0;i<100;i++) //输出 100 个负跳变
    {
        u=1;
        delay30ms();
        u=0;
        delay30ms();
    }
    while(1)
        ; //无限循环，防止程序跑飞
}
```

```
/
函数功能：外中断 T0 的中断服务程序
```

```
/
```

```
void int0(void) interrupt 0 using 0 //外中断 0 的中断编号为 0
{
```



```

    Countor++;
    P1=Countor;

}

```

### //实例 53：用外中断 0 的中断方式进行数据采集

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
sbit S=P3^2;        //将 S 位定义为 P3.2，
/
函数功能：主函数
/

void main(void)
{
    EA=1;    //开放总中断
    EX0=1;    //允许使用外中断
    IT0=1;    //选择负跳变来触发外中断
    P1=0xff;
    while(1)
        ;//无限循环，防止程序跑飞
}
/
函数功能：外中断 T0 的中断服务程序
/

void int0(void) interrupt 0 using 0 //外中断 0 的中断编号为 0
{
    P1=~P1;    //每产生一次中断请求，P1 取反一次。
}

```

### //实例 54-1：输出负脉宽为 200 微秒的方波

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
sbit u=P1^4;        //将 u 位定义为 P1.4
/
函数功能：主函数
/

void main(void)

```

```

{
    TMOD=0x02;          //TMOD=0000 0010B, 使用定时器 T0 的模式 2
    EA=1;               //开总中断
    ET0=1;             //定时器 T0 中断允许
    TH0=256-200;        //定时器 T0 的高 8 位赋初值
    TL0=256-200;        //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
{
    u=~u; //将 P1.4 引脚输出电平取反, 产生方波
}

```

## //实例 54-2: 测量负脉冲宽度

```

#include<reg51.h> //包含 51 单片机寄存器定义的头文件
sbit u=P3^2;      //将 u 位定义为 P3.2
/
函数功能: 主函数
/

void main(void)
{
    TMOD=0x02; //TMOD=0000 0010B,使用定时器 T0 的模式 2
    EA=1;      //开放总中断
    EX0=1;     //允许使用外中断
    IT0=1;     //选择负跳变来触发外中断
    ET0=1;     //允许定时器 T0 中断
    TH0=0;     //定时器 T0 赋初值 0
    TL0=0;     //定时器 T0 赋初值 0
    TR0=0;     //先关闭 T0
    while(1)
        ;//无限循环, 不停检测输入负脉冲宽度
}
/

```

函数功能：外中断 0 的中断服务程序

```
void int0(void) interrupt 0 using 0 //外中断 0 的中断编号为 0
{
    TR0=1;    //外中断一到来，即启动 T0 计时
    TL0=0;    //从 0 开始计时
    while(u==0) //低电平时，等待 T0 计时
        ;
    P1=TL0; //将结果送 P1 口显示
    TR0=0;  //关闭 T0
}
```

## //实例 55：方式 0 控制流水灯循环点亮

```
#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_nop_（）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
```

```
sbit P17=P1^7;
```

/

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：发送一个字节的数据

```
void Send(unsigned char dat)
{
    P17=0;    //P1.7 引脚输出清 0 信号，对 74LS164 清 0
    _nop_();  //延时一个机器周期
    _nop_();  //延时一个机器周期，保证清 0 完成
    P17=1;    //结束对 74LS164 的清 0
    SBUF=dat; //将数据写入发送缓冲器，启动发送
    while(TI==0) //若没有发送完毕，等待
        ;
}
```

```

    TI=0;    //发送完毕，TI 被置“1”，需将其清 0
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    SCON=0x00; //SCON=0000 0000B，使串行口工作于方式 0
    while(1)
    {
        for(i=0;i<8;i++)
        {
            Send(Tab[i]); //发送数据
            delay();      //延时
        }
    }
}

```

### //实例 56-1：数据发送程序

```

#include<reg51.h>    //包含单片机寄存器的头文件
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：向 PC 发送一个字节数据
/

```

```

void Send(unsigned char dat)
{
    SBUF=dat;
    while(TI==0)
    ;
    TI=0;
}
/

```

```

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;
    TMOD=0x20; //TMOD=0010 0000B，定时器 T1 工作于方式 2
    SCON=0x40; //SCON=0100 0000B，串口工作方式 1
    PCON=0x00; //PCON=0000 0000B，波特率 9600
    TH1=0xfd; //根据规定给定时器 T1 赋初值
    TL1=0xfd; //根据规定给定时器 T1 赋初值
    TR1=1; //启动定时器 T1
    while(1)
    {
        for(i=0;i<8;i++) //模拟检测数据
        {
            Send(Tab[i]); //发送数据 i
            delay(); //50ms 发送一次检测数据
        }
    }
}

```

## //实例 56-2：数据接收程序

```

#include<reg51.h> //包含单片机寄存器的头文件
/
函数功能：接收一个字节数据
/

unsigned char Receive(void)
{
    unsigned char dat;
    while(RI==0) //只要接收中断标志位 RI 没有被置“1”
        ; //等待，直至接收完毕（RI=1）
    RI=0; //为了接收下一帧数据，需将 RI 清 0
    dat=SBUF; //将接收缓冲器中的数据存于 dat
    return dat;
}

```

```

/
函数功能：主函数
/

void main(void)
{
    TMOD=0x20; //定时器 T1 工作于方式 2
    SCON=0x50; //SCON=0101 0000B，串口工作方式 1,允许接收（REN=1）
    PCON=0x00; //PCON=0000 0000B，波特率 9600
    TH1=0xfd; //根据规定给定时器 T1 赋初值
    TL1=0xfd; //根据规定给定时器 T1 赋初值
    TR1=1;    //启动定时器 T1
    REN=1;    //允许接收
    while(1)
    {
        P1=Receive(); //将接收到的数据送 P1 口显示
    }
}

```

### //实例 57-1：数据发送程序

```

#include<reg51.h> //包含单片机寄存器的头文件
sbit p=PSW^0;

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：向 PC 发送一个字节数据
/

void Send(unsigned char dat)
{
    ACC=dat;
    TB8=p;
    SBUF=dat;
    while(TI==0)
        ;
    TI=0;
}
/
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;
    TMOD=0x20; //TMOD=0010 0000B，定时器 T1 工作于方式 2
    SCON=0xc0; //SCON=1100 0000B，串口工作方式 3,
                //SM2 置 0，不使用多机通信，TB8 置 0
    PCON=0x00; //PCON=0000 0000B，波特率 9600
    TH1=0xfd; //根据规定给定时器 T1 赋初值
    TL1=0xfd; //根据规定给定时器 T1 赋初值
    TR1=1;    //启动定时器 T1
    while(1)
    {
        for(i=0;i<8;i++) //模拟检测数据
        {
            Send(Tab[i]); //发送数据 i
            delay();      //50ms 发送一次检测数据
        }
    }
}

```

## //实例 57-2：数据接收程序

```

#include<reg51.h> //包含单片机寄存器的头文件
sbit p=PSW^0;
/
函数功能：接收一个字节数据
/

unsigned char Receive(void)
{
    unsigned char dat;
    while(RI==0) //只要接收中断标志位 RI 没有被置"1"
        ;      //等待，直至接收完毕（RI=1）
}

```

```

        RI=0;          //为了接收下一帧数据，需将 RI 清 0
    ACC=SBUF; //将接收缓冲器中的数据存于 dat
    if(RB8==p)
    {
        dat=ACC;
        return dat;
    }
}
/
函数功能：主函数

void main(void)
{
    TMOD=0x20; //定时器 T1 工作于方式 2
    SCON=0xd0; //SCON=1101 0000B，串口工作方式 1,允许接收（REN=1）
    PCON=0x00; //PCON=0000 0000B，波特率 9600
    TH1=0xfd;  //根据规定给定时器 T1 赋初值
    TL1=0xfd;  //根据规定给定时器 T1 赋初值
    TR1=1;     //启动定时器 T1
    REN=1;     //允许接收
    while(1)
    {
        P1=Receive(); //将接收到的数据送 P1 口显示
    }
}

```

### //实例 58：单片机向 PC 发送数据

```

#include<reg51.h>          //包含单片机寄存器的头文件
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：向 PC 发送一个字节数据

void Send(unsigned char dat)
{
    SBUF=dat;
    while(TI==0)
    ;
    TI=0;
}

```



```

/
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;
    TMOD=0x20; //TMOD=0010 0000B，定时器 T1 工作于方式 2
    SCON=0x40; //SCON=0100 0000B，串口工作方式 1
    PCON=0x00; //PCON=0000 0000B，波特率 9600
    TH1=0xfd; //根据规定给定时器 T1 赋初值
    TL1=0xfd; //根据规定给定时器 T1 赋初值
    TR1=1; //启动定时器 T1
    while(1)
    {
        for(i=0;i<8;i++) //模拟检测数据
        {
            Send(Tab[i]); //发送数据 i
            delay(); //150ms 发送一次数据
        }
    }
}

```

## //实例 59：单片机接收 PC 发出的数据

```

#include<reg51.h> //包含单片机寄存器的头文件
/
函数功能：接收一个字节数据

unsigned char Receive(void)
{
    unsigned char dat;

```

```

while(RI==0) //只要接收中断标志位 RI 没有被置“1”
    ; //等待，直至接收完毕（RI=1）
    RI=0; //为了接收下一帧数据，需将 RI 清 0
    dat=SBUF; //将接收缓冲器中的数据存于 dat
    return dat;
}
/

```

函数功能：主函数

```

void main(void)
{
    TMOD=0x20; //定时器 T1 工作于方式 2
    SCON=0x50; //SCON=0101 0000B，串口工作方式 1,允许接收（REN=1）
    PCON=0x00; //PCON=0000 0000B，波特率 9600
    TH1=0xfd; //根据规定给定时器 T1 赋初值
    TL1=0xfd; //根据规定给定时器 T1 赋初值
    TR1=1; //启动定时器 T1
    REN=1; //允许接收
    while(1)
    {
        P1=Receive(); //将接收到的数据送 P1 口显示
    }
}
/

```

数码管显示          数码管显示          数

码管显示                  数码管显示

/

## //实例 60：用 LED 数码显示数字 5

```

#include<reg51.h> // 包含 51 单片机寄存器定义的头文件
void main(void)
{

```

```

P2=0xfe;    //P2.0 引脚输出低电平，数码显示器接通电源准备点亮
P0=0x92;    //让 P0 口输出数字"5"的段码 92H
}

```

## //实例 61：用 LED 数码显示器循环显示数字 0~9

```

#include<reg51.h>    // 包含 51 单片机寄存器定义的头文件
/
函数功能：延时函数，延时一段时间
/

void delay(void)
{
    unsigned char i,j;
    for(i=0;i<255;i++)
        for(j=0;j<255;j++)
            ;
}
/
函数功能：主函数
/

void main(void)
{
    unsigned char i;
    unsigned char code
    Tab[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
    //数码管显示 0~9 的段码表，程序运行中当数组值不发生变化
    时，
    //前面加关键字 code ，可以大大节约单片机的存储空间
    P2=0xfe;    //P2.0 引脚输出低电平，数码显示器 DS0 接通电源工作
    while(1)    //无限循环
    {
        for(i=0;i<10;i++)
        {
            P0=Tab[i]; //让 P0 口输出数字的段码 92H
            delay();    //调用延时函数
        }
    }
}

```

## //实例 62：用数码管慢速动态扫描显示数字"1234"

```
#include<reg51.h>          // 包含 51 单片机寄存器定义的头文件
void delay(void)           //延时函数，延时一段时间
{
    unsigned char i,j;
    for(i=0;i<250;i++)
        for(j=0;j<250;j++)
            ;
}
void main(void)
{
    while(1)    //无限循环
    {
        P2=0xfe;    //P2.0 引脚输出低电平，DS0 点亮
        P0=0xf9;    //数字 1 的段码
        delay();
        P2=0xfd;    //P2.1 引脚输出低电平，DS1 点亮
        P0=0xa4;    //数字 2 的段码
        delay();
        P2=0xfb;    //P2.2 引脚输出低电平，DS2 点亮
        P0=0xb0;    //数字 3 的段码
        delay();
        P2=0xf7;    //P2.3 引脚输出低电平，DS3 点亮
        P0=0x99;    //数字 4 的段码
        delay();
        P2=0xff;
    }
}
```

## //实例 63：用 LED 数码显示器伪静态显示数字 1234

```
#include<reg51.h>          // 包含 51 单片机寄存器定义的头文件
void delay(void)           //延时函数，延时约 0.6 毫秒
{
    unsigned char i;
    for(i=0;i<200;i++)
        ;
}

void main(void)
```

```

{
    while(1)    //无限循环
    {
        P2=0xfe;    //P2.0 引脚输出低电平，DS0 点亮
        P0=0xf9;    //数字 1 的段码
        delay();
        P2=0xfd;    //P2.1 引脚输出低电平，DS1 点亮
        P0=0xa4;    //数字 2 的段码
        delay();
        P2=0xfb;    //P2.2 引脚输出低电平，DS2 点亮
        P0=0xb0;    //数字 3 的段码
        delay();
        P2=0xf7;    //P2.3 引脚输出低电平，DS3 点亮
        P0=0x99;    //数字 4 的段码
        delay();
        P2=0xff;
    }
}

```

## //实例 64：用数码管显示动态检测结果

```

#include<reg51.h>    // 包含 51 单片机寄存器定义的头文件
#include<stdlib.h>    //包含随机函数 rand()的定义文件
unsigned char i;    //记录中断次数
unsigned int x;    //随机检测的数据
unsigned char code Tab[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
//数码管显示 0~9 的段码表

```

/

函数功能：快速动态扫描延时，延时约 0.9 毫秒

```

/
void delay(void)
{
    unsigned int i;
    for(i=0;i<300;i++)
        ;
}

```

/

函数功能：4 位数的数码显示器显示

入口参数：k

出口参数：无

```
/
void display(unsigned int k)
{
    P2=0xfe;    //即 P2=1111 1110B, P2.0 引脚输出低电平, 数码显示器 DS0 接
通电源
    P0=Tab[k/1000];    //显示千位
    delay();
    P2=0xfd;    //即 P2=1111 1101B, P2.1 引脚输出低电平, 数码显示器 DS1 接通
电源
    P0=Tab[(k%1000)/100];    //显示百位
    delay();
    P2=0xfb;    //即 P2=1111 1011B, P2.2 引脚输出低电平, 数码显示器 DS2 接通
电源
    P0=Tab[(k%100)/10];    //显示十位
    delay();
    P2=0xf7;    //即 P2=1111 0111B , P2.3 引脚输出低电平, 数码显示器 DS3 接
通电源
    P0=Tab[k%10];    //显示个位
    delay();
    P2=0xff;    //关闭所有显示器
}

void main(void)    //主函数
{
    TMOD=0x01;    //使用定时器 T0
    TH0=(65536-46083)/256;    //将定时器计时时间设定为 46083×
1.085 微秒=50000 微秒=50 毫秒
    TL0=(65536-46083)%256;
    EA=1;    //开启总中断
    ET0=1;    //定时器 T0 中断允许
    TR0=1;    //启动定时器 T0 开始运行

    while(1)
    {
        display(x);    //调用检测结果的显示程序
    }
}
/

    函数功能：定时器 T0 的中断服务程序
```

/

```

void Time0(void) interrupt 1 using 1
{
    TR0=0;    //关闭定时器 T0
    i++;      //每来一次中断，i 自加 1
    if(i==20) //够 20 次中断，即 1 秒钟进行一次检测结果采样
    {
        x=rand()/10;    //随机产生一个从 0 到 32767 的整数，再将其除以 10，
        获得一个随机 4 位数，模拟检测结果
        i=0;            //将 i 清 0，重新统计中断次数
    }
    TH0=(65536-46083)/256; //重新给计数器 T0 赋初值
    TL0=(65536-46083)%256;
    TR0=1;    //启动定时器 T0
}

```

## //实例 65：数码秒表设计

```

#include<reg51.h> // 包含 51 单片机寄存器定义的头文件
unsigned char code Tab[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
//数码管显示 0~9 的段码表
unsigned char int_time;    //记录中断次数
unsigned char second;      //储存秒
/
函数功能：快速动态扫描延时,延时约 0.6 毫秒

/
void delay(void)
{
    unsigned char i;
    for(i=0;i<200;i++)
        ;
}
/
函数功能：显示秒
入口参数：k
出口参数：无

/
void DisplaySecond(unsigned char k)

```

```

{

    P2=0xfb;          //P2.6 引脚输出低电平， DS6 点亮
    P0=Tab[k/10];      //显示十位
    delay();

    P2=0xf7;          //P2.7 引脚输出低电平， DS7 点亮
    P0=Tab[k%10];      //显示个位
    delay();
    P2=0xff;          //关闭所有数码管

}

void main(void)      //主函数
{
    TMOD=0x01;        //使用定时器 T0
    TH0=(65536-46083)/256; //将定时器计时时间设定为  $46083 \times 1.085$  微
秒
    //50000 微秒=50 毫秒
    TL0=(65536-46083)%256;
    EA=1;              //开启总中断
    ET0=1;              //定时器 T0 中断允许
    TR0=1;              //启动定时器 T0 开始运行
    int_time=0;         //中断次数初始化
    second=0;           //秒初始化
    while(1)
    {
        DisplaySecond(second); //调用秒的显示子程序
    }
}

//
//函数功能：定时器 T0 的中断服务程序
//
void interserve(void ) interrupt 1 using 1
{
    TR0=0;             //关闭定时器 T0
    int_time++;         //每来一次中断,中断次数 int_time 自加 1
    if(int_time==20)    //够 20 次中断,即 1 秒钟进行一次检测结果采样
    {
        int_time=0;     //中断次数清 0
        second++;       //秒加 1
        if(second==60)
            second =0; //秒等于 60 就返回 0
    }
}

```



```

    }
    TH0=(65536-46083)/256;    //重新给计数器 T0 赋初值
    TL0=(65536-46083)%256;
    TR0=1;    //启动定时器 T0
}

```

## //实例 66：数码时钟设计

```

#include<reg51.h>    // 包含 51 单片机寄存器定义的头文件
unsigned char Tab[ ]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
//control  shape
unsigned char port[8]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
unsigned char int_time ; //中断次数计数变量
unsigned char second;    //秒计数变量
unsigned char minute;    //分钟计数变量
unsigned char hour;    //小时计数变量

```

```

////////////////////////////////////

```

```

void delay(void)    //延时函数，延时约 0.6ms
{
    unsigned char j;
    for(j=0;j<200;j++)
        ;
}

```

```

/
函数功能：显示秒的子程序
入口参数：s

```

/

```

void DisplaySecond(unsigned char s)
{
    P2=0xbf;    //P2.6 引脚输出低电平， DS6 点亮
    P0=Tab[s/10];    //显示十位
    delay();

    P2=0x7f;    //P2.7 引脚输出低电平， DS7 点亮
    P0=Tab[s%10];    //显示个位
    delay();
    P2=0xff;    //关闭所有数码管
}

```

```
}
```

```
/
```

函数功能：显示分钟的子程序

入口参数：m

```
/
```

```
void DisplayMinute(unsigned char m)
```

```
{
```

```
    P2=0xf7;    // P2.3 引脚输出低电平， DS3 点亮
```

```
    P0=Tab[m/10]; //显示个位
```

```
    delay();
```

```
    P2=0xef;    // P2.4 引脚输出低电平， DS4 点亮
```

```
    P0=Tab[m%10];
```

```
    delay();
```

```
    P2=0xdf;    //P2.5 引脚输出低电平， DS5 点亮
```

```
    P0=0xbf;    //分隔符“-”的段码
```

```
    delay();
```

```
    P2=0xff;    //关闭所有数码管
```

```
}
```

```
/
```

函数功能：显示小时的子程序

入口参数：h

```
/
```

```
void DisplayHour(unsigned char h)
```

```
{
```

```
    P2=0xfe;        //P2.0 引脚输出低电平， DS0 点亮
```

```
    P0=Tab[h/10];    //显示十位
```

```
    delay();
```

```
    P2=0xfd;        //P2.1 引脚输出低电平， DS1 点亮
```

```
    P0=Tab[h%10];    //显示个位
```

```
    delay();
```

```
    P2=0xfb;        //P2.2 引脚输出低电平， DS2 点亮
```

```
    P0=0xbf;        //分隔符“-”的段码
```

```
    delay();
```

```
    P2=0xff;        //关闭所有数码管
```

```
}
```

```
/
```

函数功能：主函数

/

```
void main(void)
{
    TMOD=0x01;           //使用定时器 T0
    EA=1;                 //开中断总允许
    ET0=1;                //允许 T0 中断
    TH0=(65536-46083)/256; //定时器高八位赋初值
    TL0=(65536-46083)%256; //定时器低八位赋初值
    TR0=1;
    int_time=0;           //中断计数变量初始化
    second=0;             //秒计数变量初始化
    minute=0;            //分钟计数变量初始化
    hour=0;               //小时计数变量初始化

    while(1)
    {
        DisplaySecond(second); //调用秒显示子程序
        delay();
        DisplayMinute(minute); //调用分钟显示子程序
        delay();
        DisplayHour(hour);
        delay();
    }
}
```

/

函数功能：定时器 T0 的中断服务子程序

/

```
void interserve(void ) interrupt 1 using 1 //using Time0
{
    int_time++;
    if(int_time==20)
    {
        int_time=0; //中断计数变量清 0
        second++;   //秒计数变量加 1
    }
    if(second==60)
    {
        second=0; //如果秒计满 60，将秒计数变量清 0
        minute++; //分钟计数变量加 1
    }
    if(minute==60)
```

```

        {
            minute=0;    //如果分钟计满 60，将分钟计数变量
清 0
            hour++;      //小时计数变量加 1
        }
        if(hour==24)
        {
            hour=0;      //如果小时计满 24，将小时计数变量
清 0
        }

        TH0=(65536-46083)/256;    //定时器重新赋初值
        TL0=(65536-46083)%256;

    }

```

## //实例 67：用 LED 数码管显示计数器 T0 的计数值

```

#include<reg51.h>    //包含 51 单片机寄存器定义的头文件
sbit S=P3^2;    //将 S 位定义为 P3.2 引脚
unsigned char Tab[ ]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};    //段
码表
unsigned char x;
/
函数功能： 延时约 0.6ms
/

void delay(void)
{
    unsigned char j;
    for(j=0;j<200;j++)
        ;
}

/
函数功能： 显示计数次数的子程序
入口参数： x
/

void Display(unsigned char x)
{

```

```

    P2=0xf7;          //P2.6 引脚输出低电平，DS6 点亮
    P0=Tab[x/10];     //显示十位
    delay();
    P2=0xfb;          //P2.7 引脚输出低电平，DS7 点亮
    P0=Tab[x%10];     //显示个位
    delay();

}

/
函数功能：主函数
/

void main(void)
{
    EA=1;    //开放总中断
    EX0=1;   //允许使用外中断
    IT0=1;   //选择负跳变来触发外中断
    x=0;

    while(1)
        Display(x);

}

/
函数功能：外中断 T0 的中断服务程序
/

void int0(void) interrupt 0 using 0 //外中断 0 的中断编号为 0
{
    x++;
    if(x==100)
        x=0;

}

```

## //实例 68：静态显示数字“59”

```

#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)

```

```

{
    P0=0x92;    //将数字 5 的段码送 P0 口
    P1=0x90;    //将数字 9 的段码送 P1 口
    while(1)    //无限循环，防止程序跑飞
        ;
}

```

### //实例 69：无软件消抖的独立式键盘输入实验

```

#include<reg51.h>    // 包含 51 单片机寄存器定义的头文件
sbit S1=P1^4;        //将 S1 位定义为 P1.4 引脚
sbit LED0=P3^0;      //将 LED0 位定义为 P3.0 引脚
void main(void)      //主函数
{
    LED0=0;          //P3.0 引脚输出低电平
    while(1)
    {
        if(S1==0)    //P1.4 引脚输出低电平，按键 S1 被按下
            LED0=!LED0; //P3.0 引脚取反
    }
}

```

### //实例 70：软件消抖的独立式键盘输入实验

```

#include<reg51.h>    // 包含 51 单片机寄存器定义的头文件
sbit S1=P1^4;        //将 S1 位定义为 P1.4 引脚
sbit LED0=P3^0;      //将 LED0 位定义为 P3.0 引脚
/
函数功能：延时约 30ms
/

void delay(void)
{
    unsigned char i,j;
    for(i=0;i<100;i++)
        for(j=0;j<100;j++)
            ;
}
/

```

函数功能：主函数

```
void main(void) //主函数
{
    LED0=0;      //P3.0 引脚输出低电平
    while(1)
    {
        if(S1==0) //P1.4 引脚输出低电平，按键 S1 被按下
        {
            delay(); //延时一段时间再次检测
            if(S1==0) // 按键 S1 的确被按下
                LED0=!LED0; //P3.0 引脚取反
        }
    }
}
```

## //实例 71：CPU 控制的独立式键盘扫描实验

```
#include<reg51.h> //包含 51 单片机寄存器定义的头文件
sbit S1=P1^4;      //将 S1 位定义为 P1.4 引脚
sbit S2=P1^5;      //将 S2 位定义为 P1.5 引脚
sbit S3=P1^6;      //将 S3 位定义为 P1.6 引脚
sbit S4=P1^7;      //将 S4 位定义为 P1.7 引脚
unsigned char keyval; //储存按键值
```

/

函数功能：流水灯延时

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

/

函数功能：软件消抖延时

```
void delay30ms(void)
{
    unsigned char i,j;
```