

## 单片机设计题——数码管

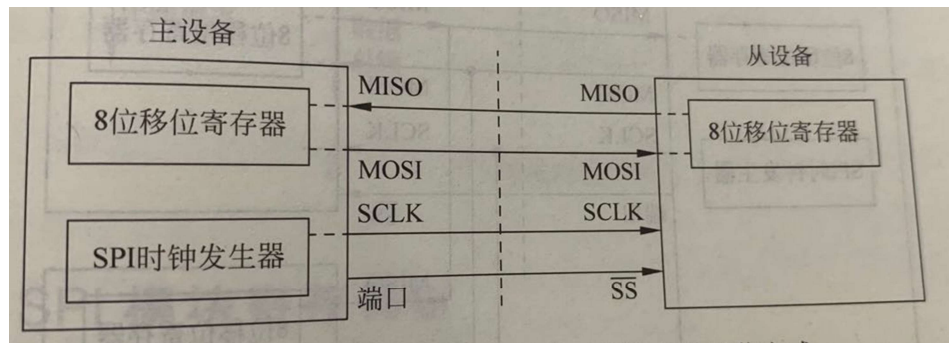
**【实验要求】** 该次作业，用于熟悉 SPI 接口和 PWM。

- 1) 使用 STC 单片机上的七段数码管，在竖向方向上实现，从上向下“下雨”的效果，填满所有 7 段数码管，然后清空，再重新填充七段数码管。（80 分）
- 2) 在 7 段数码管上实现“贪吃蛇”演示效果（20 分）

### 1. 设计思路

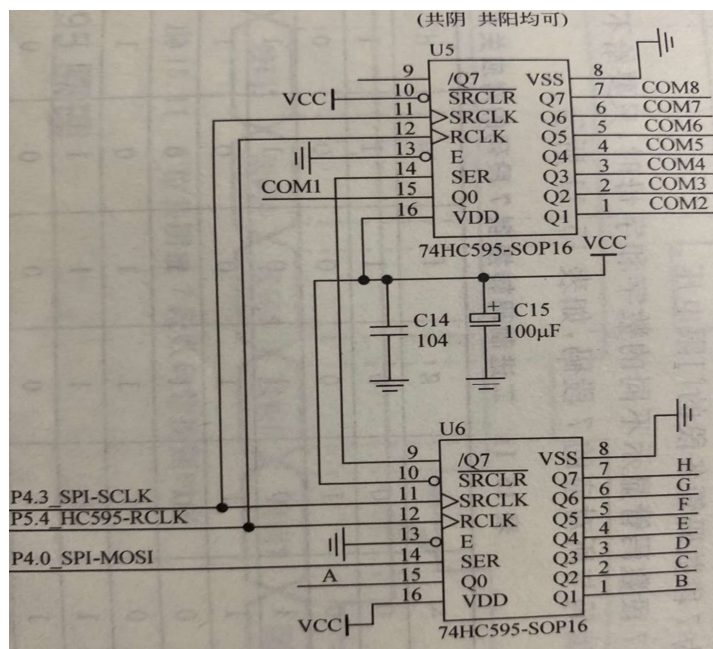
本程序参照书 P430 例程搭建代码程序结构

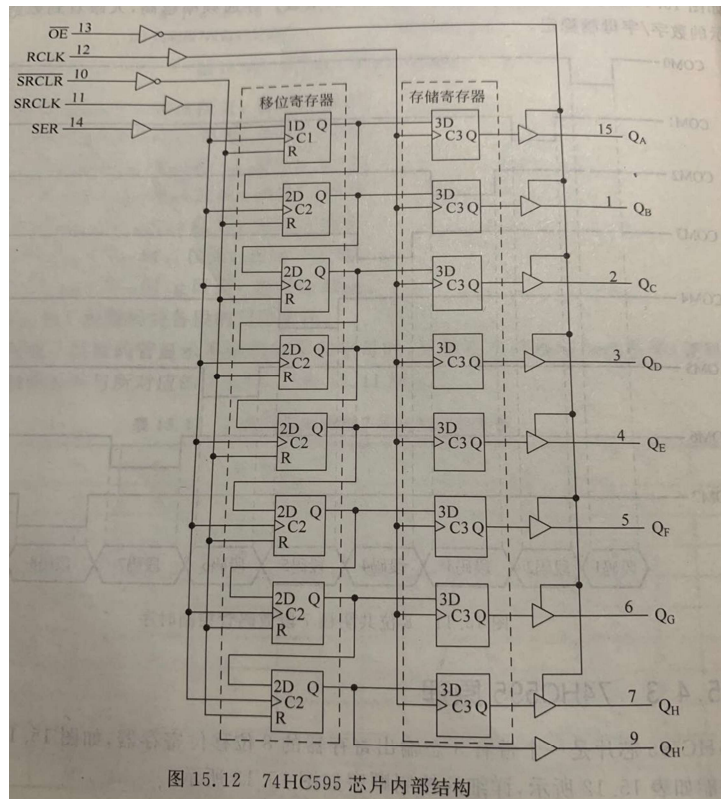
- (1) **明确 SPI 原理：**基于 SPI 的通信方式是典型的高速同步双向数据传输方式。其主要实现方式是主设备的 SPI 接口和从设备的 SPI 的 8 位移位寄存器构成 1 个循环的 16 位移位寄存器，且主设备发出同步时钟信号 SCLK 和 SS 信号。



- (2) **明确 74HC595 控制结构：**U5 输出管选码，U6 输出段码。

数据从标记为 U6 的 74HC595 器件的 SER 引脚输入，在每个 SRCLK 的上升沿，将 SER 引脚上的数据移入到移位寄存器，在 SRCLK 的第 9 个上升沿，数据开始从 Q7 移出。由于将 U6 的 Q7 和 U5 的 SER 引脚连接在一起，数据开始移入标记为 U5 的 74HC595 器件内的移位寄存器，如此下去。当第 16 个上升沿的时候，16 位数据充满移位寄存器。





- (3) **数码管扫描：**利用定时器 0 的定时中断对数码管进行扫描，使数码管正常显示。利用定时器 1 设定 1Hz 的中断频率，来控制整个动画过程的显示进度。
- (4) **动画显示：**模仿视频制作帧动画的方式，将下雨动画分为 6 步完成，将贪吃蛇演示动画分为 11 步完成。并且为方便代码书写，提前定义图形显示的段码数组。

## 2. 调试过程及遇到问题的解决办法

### 变量重定义问题。

错误情况：在程序最开始定义了全局变量，`unsigned int j`；但是在主函数开始错误的再次定义了 `unsigned char j`；编译并没有出现错误，但是在实际的调试过程中 `j` 失去了全局变量的作用，从而在中断函数中的 `j++` 操作并没生效，导致动画没有形成。

解决方法：将主函数中的二次定义删掉。

## 3. 实验结果

详细介绍请看压缩包内视频

按中断 0 按键能切换下雨和贪吃蛇显示模式

#### 4. 代码及注释

##### spi.h

```
#define TMS 65500          //定义定时器 0 的计数初值
#define TMS1 3036         //定义定时器 1 的计数初值
#define SSIG      1       //定义 SPCTL 寄存器 SSIG 位的值，主模式
#define SPEN      1       //定义 SPCTL 寄存器 SPEN 位的值，使能 SPI
#define DORD      0       //定义 SPCTL 寄存器 DORD 位的值，先送 MSB
#define MSTR      1       //定义 SPCTL 寄存器 MSTR 位的值，SPI 为主机
#define CPOL      1       //定义 SPCTL 寄存器 CPOL 位的值，空闲时为高电平
#define CPHA      1       //定义 SPCTL 寄存器 CPHA 位的值，前沿驱动数据
#define SPR1      0       //与 SPECTL 寄存器 SPRO 一起确定 SPI 的时钟频率
#define SPRO      0       //SPI 时钟频率为 CPU 时钟的 1/4
#define SPEED_4   0       //SPI 时钟频率为 CPU 时钟的 1/4
#define SPEED_8   1       //SPI 时钟频率为 CPU 时钟的 1/8
#define SPEED_16  2       //SPI 时钟频率为 CPU 时钟的 1/16
#define SPEED_32  3       //SPI 时钟频率为 CPU 时钟的 1/32

#define SPIF      0x80     //定义 SPSTAT 寄存器 SPIF 标志的值
#define WCOL      0x40     //定义 SPSTAT 寄存器 WCOL 标志的值

sfr SPSTAT =0xCD;         //定义 SPSTAT 寄存器的地址 0xCD
sfr SPCTL   =0xCE;         //定义 SPCTL 寄存器的地址 0xCE
sfr SPDAT   =0xCF;         //定义 SPDAT 寄存器的地址 0xCF

sfr AUXR    =0x8E;         //定义 AUXR 寄存器的地址 0x8E
sfr AUXR1   =0xA2;         //定义 AUXR1 寄存器的地址 0xA2
sfr CLK_DIV =0x97;         //定义 CLK_DIV 寄存器的地址 0x97
sfr P5      =0xC8;         //定义 P5 寄存器的地址 0xC8
sfr P4      =0xC0;         //定义 P4 寄存器的地址 0xC0
sbit P4_6=P4^6;           //定义 P4.6 管脚
sbit HC595_RCLK=P5^4;     //定义 P5.4 管脚
```

##### main.c

```
#include "reg51.h"
#include "spi.h"
//定义下雨显示图形库
unsigned char code t_display[6]={0x00,0x01,0x23,0x63,0x77,0x7F};
//定义贪吃蛇显示图形库
unsigned char code t_display1[5]={0x00,0x01,0x02,0x40,0x42};
//定义管选码的反码，在一个时刻只有一个管选信号为低，其余为高
unsigned char code T_COM[8]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

```

bit flag=0; //定义全局变量 flag
unsigned int m=0; //定义全局变量 m
unsigned int j=0; //定义全局变量 j
unsigned int state=0; //定义全局变量 state，控制显示的模式
//定义 SPI 数据发送函数
void SPI_SendByte(unsigned char dat)
{
    SPSTAT=SPIF+WCOL; //软件写 1 清零 SPSTAT 寄存器
    SPDAT=dat; //将 dat 写入 SPDAT SPI 数据寄存器
    while((SPSTAT & SPIF)==0); //判断发送是否完成
    SPSTAT=SPIF+WCOL; //软件写 1 清零 SPSTAT 寄存器
}
//定义用于写 7 段数码管的子函数 seg7scan（下雨显示）
void seg7scan(unsigned char index1,unsigned char index2)
{
    SPI_SendByte(~T_COM[index1]); //向 74HC595（U5）写入管选信号
    SPI_SendByte(t_display[index2]); //向 74HC595（U6）写入段码数据
    HC595_RCLK=1; //上升沿有效
    HC595_RCLK=0;
}
//定义用于写 7 段数码管的子函数 seg7scan（贪吃蛇显示）
void seg7scan1(unsigned char index1,unsigned char index2)
{
    SPI_SendByte(~T_COM[index1]); //向 74HC595（U5）写入管选信号
    SPI_SendByte(t_display1[index2]); //向 74HC595（U6）写入段码数据
    HC595_RCLK=1; //上升沿有效
    HC595_RCLK=0;
}
//声明定时器 0 的中断服务程序
void timer_0() interrupt 1
{
    flag=1; //将 flag 标志置 1
}
//声明定时器 1 的中断服务程序
void timer_1() interrupt 3
{
    P46=!P46; //进入一次中断，控制 LED 改变一次
    if(state==0) //如果下雨显示模式，m++
    {
        m++;
        if(m==6) m=0;
    }
    else //如果贪吃蛇显示模式，j++
    {

```

```

        j++;
        if(j==11) j=0;
    }
}
//声明外部中断 0 的中断服务程序
void interrupt_0() interrupt 0
{
    state^=1;                //显示模式改变一次
    m=0;
    j=0;
}
//主函数
void main()
{
    unsigned char i=0;        //定义本地字符型变量 char
    SPCTL=(SSIG<<7)+(SPEN<<6)+(DORD<<5)+(MSTR<<4)
        +(CPOL<<3)+(CPHA<<2)+SPEED_4;
    //设置为主模式，使能 SPI，先送 MSB，SPI 为主机
    //空闲时为高电平，前沿驱动数据，SPI 时钟频率为 CPU 四分频
    AUXR1=0X08;              //将 SPI 接口信号切换到第三组引脚上

    CLK_DIV=0x03;            //主时钟 8 分频作为 SYSclk 频率
    TL0=TIMS;                //TIMS 写入定时器 0 低 8 位寄存器 TL0
    TH0=TIMS>>8;             //TIMS 写入定时器 0 高 8 位寄存器 TH0
    TL1=TIMS1;               //TIMS1 写入定时器 1 低 8 位寄存器 TL1
    TH1=TIMS1>8;             //TIMS1 写入定时器 1 高 8 位寄存器 TH1
    AUXR&=0x3F;              //定时器 0 和 1 是 12 分频
    TMOD=0x00;               //定时器 0/1，16 位重加载定时器模式
    TR0=1;                   //启动定时器 0
    TR1=1;                   //启动定时器 1
    ET0=1;                   //允许定时器 0 溢出中断
    ET1=1;                   //允许定时器 1 溢出中断
    EX0=1;                   //开启外部中断 0
    IT0=1;                   //设置外部中断 0 为下降沿触发
    EA=1;                    //CPU 允许响应中断请求

    while(1)                 //无限循环
    {
        while(state==0)      //当下雨显示模式时
        {
            if(flag==1)       //如果定时器 0 中断
            {
                flag=0;        //将 flag 清 0
                for(i=0;i<8;i++) //轮流导通七段数码管，需要 8 次

```

```

        {
            seg7scan(i,m);          //显示内容，定时器 1 中断触发，内容才改变
        }
    }
}
while(state==1)                    //当贪吃蛇显示模式
{
    if(flag==1)                    //如果定时器 0 中断
    {
        flag=0;                    //将 flag 清 0
        //将动画内容分 11 步进行显示，定时器 1 中断 1 次，前进一步
        switch(j)
        {
            case 0:{
                seg7scan1(0,1);
                seg7scan1(1,0);
                seg7scan1(2,0);
                seg7scan1(3,3);
                for(i=4;i<8;i++)
                    seg7scan1(i,0);

                };break;

            case 1:{
                seg7scan1(0,0);
                seg7scan1(1,1);
                seg7scan1(2,0);
                seg7scan1(3,3);
                for(i=4;i<8;i++)
                    seg7scan1(i,0);

                };break;

            case 2:{
                seg7scan1(0,0);
                seg7scan1(1,2);
                seg7scan1(2,0);
                seg7scan1(3,3);
                for(i=4;i<8;i++)
                    seg7scan1(i,0);

                };break;

            case 3:{
                seg7scan1(0,0);
                seg7scan1(1,0);
                seg7scan1(2,3);
            }
        }
    }
}

```

```

        seg7scan1(3,3);
        for(i=4;i<8;i++)
            seg7scan1(i,0);

    };break;

case 4:{

        for(i=0;i<3;i++)
            seg7scan1(i,0);
        seg7scan1(3,3);
        seg7scan1(4,3);
        for(i=5;i<8;i++)
            seg7scan1(i,0);

    };break;

case 5:{

        for(i=0;i<4;i++)
            seg7scan1(i,0);
        seg7scan1(4,3);
        seg7scan1(5,3);
        seg7scan1(6,0);
        seg7scan1(7,0);
    };break;

case 6:{

        for(i=0;i<5;i++)
            seg7scan1(i,0);
        seg7scan1(5,4);
        seg7scan1(6,0);
        seg7scan1(7,0);
    };break;

case 7:{

        for(i=0;i<5;i++)
            seg7scan1(i,0);
        seg7scan1(5,2);
        seg7scan1(6,1);
        seg7scan1(7,0);
    };break;

case 8:{

        for(i=0;i<6;i++)
            seg7scan1(i,0);
        seg7scan1(6,1);
        seg7scan1(7,1);
    };break;

case 9:{

        for(i=0;i<7;i++)

```

```
        seg7scan1(i,0);
        seg7scan1(7,1);
    };break;
    case 10:{
        for(i=0;i<8;i++)
            seg7scan1(i,0);

        };break;
    default;;
    }
}
}
```