《单片机原理及应用》第五次作业

【设计目标】

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）

【设计思路】

1.通过外部中断 0 控制下雨模式与贪吃蛇模式的切换，每进入一个模式，都将从该模式的初始状态开始演示。

2.下雨模式中，利用正方形表示雨滴，在竖向上，雨滴从两边一次落下，并在底部累积，直到将全部数码管点亮，将重新开始下雨模式的演示。

3.贪吃蛇模式中，初始状态为蛇在左上角，只有一段，运动状态为向右，单片机随机产生食物。

4.通过扫描键盘上的 1，4，5，6 分别代表上，左，下，右来改变贪吃蛇的运动方向，考虑在数码管的特殊性，贪吃蛇的运动方向不能够直接转 180 度。如果不按键改变方向，贪吃蛇会一直往一个方向运动。

5.考虑到数码管显示空间小的问题，贪吃蛇碰到四周会从另一端出来，同时若是考虑贪吃蛇头部的碰撞问题，贪吃蛇运动的空间将限制的极小，所以仅仅演示到四段游戏就将结束，也不考虑头部碰到身体的问题。

6.利用 x，y 数组存储贪吃蛇的位置坐标，x 表示数码管的标号，范围为 0~7，y 表示数码管显示的 led 标号，范围为 0~6；用 xnext，ynext 存储下一状态贪吃蛇第一段的位置坐标。

7.利用 dir，dir0 分别表示当前位置状态的运动方向和前一位置状态的运动方向，用 0，1，2，3 分别表示下，右，上，左。注：状态指位置的状态，非运动方向的状态，例如：一直向右运动时，dir 和 dir0 都为 1，若按下向下按键，dir0 仍为 1，dir 变为 0，等到下一次位置更新之后，dir0 也变为 0.

8.利用 goal 表示食物是否已经被吃掉，若 goal=1，则表示食物已经被吃，需要重新随机产生一个食物，产生后将 goal 置 0；当贪吃蛇下一位置将与食物重叠，则将 goal 置 1，表示已将食物吃掉。

9.利用 length 表示当前贪吃蛇的长度，没吃掉一个食物，length 加 1，直到 length=4，游戏结束，数码管显示 good，利用外部中断 0 切换至下雨模式后在切回贪吃蛇模式可重新进行游戏。

【设计代码】

头文件部分

spi.h

```
#define TIMS 131000 //定义定时器 0 的计数初值
#define TIMS1 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
#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 SPR0 一起确定 SPI 的时钟频率
```

```
#define SPR0 0 //SPI 时钟频率为 CPU 时钟的 1/4
#define SPEED_4 0
#define SPEED_16 1
#define SPEED_64 2
#define SPEED_128 3
#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 端口寄存器的地址 0xC8
sbit HC595_RCLK=P5^4; //定义 P5.4 引脚
```

主程序部分

```
#include "reg51.h"
#include "stdlib.h"
#include "spi.h" //包含自定义头文件


//t_display 数组保存着下雨模式要显示的段码
unsigned char code t_display[4]={0x63,0x5C,0x00,0x7F};
//snake 数组保存贪吃蛇模式要显示的段码
unsigned char code snake[7]={0x01,0x02,0x04,0x08,0x10,0x20,0x40};
//T-COM 数组保存着管选码的反码，在一个时刻只有一个管选信号为低
unsigned char code T_COM[8]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
//gg 数组保存着游戏结束时要显示的段码
unsigned char code gg[8]={0x6F,0x5C,0x5C,0x5E};
unsigned char x[4]={0,0,0,0},xnext;                    //存储贪吃蛇位置坐标
unsigned char y[4]={0,0,0,0},ynext;                    //xnext，ynext 存储下一步的位置坐标
unsigned char a,b;                                                    //存储食物的位置坐
标
bit flag=0,change=0;              //定义全局位变量 flag
unsigned m=0;                              //定义全局无符号变量 m
unsigned mode=0;                           //声明控制遥控模式的位
unsigned dir=1,dir0=1;                     //声明判断贪吃蛇运动方向的变量
unsigned length=1;                    //声明贪吃蛇长度变量
unsigned goal=1;                           //声明食物是否需要生成的变量


void delay(int i)                          //声明延时函数
{
    while(--i);
```

```
}

void SPI_SendByte(unsigned char dat)          //定义 SPI 数据发送函数
{
    SPSTAT=SPIF+WCOL;                         //写“1”清零 SPSTAT 寄存器内容
    SPDAT=dat;                                    //dat 写入 SPDATSPI 数据寄存器
    while((SPSTAT & SPIF)==0); //判断发送是否完成，没有则等待
    SPSTAT=SPIF+WCOL;                             //写“1”清零 SPSTAT 寄存器内容
}


//定义 7 段数码管的函数 seg7scan，index1 参数控制管选，Index2 控制段码
void seg7scan(unsigned char index1,unsigned char index2)
{ SPI_SendByte(~T_COM[index1]);                   //向 74HCT595（U5）写入管选信号
    switch(mode)                                                  //向 74HCT595（U6）
写入段码数据
    {
        case 0:                 //下雨模式
            SPI_SendByte(t_display[index2]);break;
        case 1:                 //贪吃蛇模式
            SPI_SendByte(snake[index2]);break;
    }
    HC595_RCLK=1;                                                 //通过 P5.4 端口向两
片 595 发数据锁存
    HC595_RCLK=0;                                                 //上升沿有效
}
//定义显示游戏结束提示函数
void good()                 //显示“good”字符
{
    unsigned char j;
    for(j=0;j<4;j++)
    {
        SPI_SendByte(~T_COM[j]);              //向 74HCT595（U5）写入管选信号
        SPI_SendByte(gg[j]);                      //向 74HCT595（U6）写入段码数据
        HC595_RCLK=1;                                             //通过 P5.4 端口
向两片 595 发数据锁存
        HC595_RCLK=0;                                             //上升沿有效
    }
}

void down()                           //下一步运动方向为向下时，判断下一步的位置
{
    xnext=x[0];
    ynext=y[0];
```

```
    if(dir0==3)                         //方向由左往下变化
    {
        if(y[0]==0||y[0]==3) ynext=5;
        else if(y[0]==6) ynext=4;
    }else if(dir0==1)                   //方向由右往下变化
    {
        if(y[0]==0||y[0]==3) ynext=1;
        else if(y[0]==6) ynext=2;
    }else if(dir0==0)                   //方向一直为向下
    {
        if(y[0]==4||y[0]==1) ynext=y[0]+1;
        else ynext=y[0]-1;
    }
}

void right()                            //下一步运动方向为向右时，判断下一步的位置
{
    xnext=x[0];
    ynext=y[0];
    if(dir0==0)                         //方向由下往右变化
    {
        if(y[0]==5) ynext=6;
        else if(y[0]==4) ynext=3;
        else
        {
            xnext=(x[0]+1)%8;
            if(y[0]==1) ynext=6;
            else if(y[0]==2) ynext=3;
        }
    }else if(dir0==2)                   //方向由上往右变化
    {
        if(y[0]==5) ynext=0;
        else if(y[0]==4) ynext=6;
        else
        {
            xnext=(x[0]+1)%8;
            if(y[0]==1) ynext=0;
            else if(y[0]==2) ynext=6;
        }
    }else if(dir0==1)                   //方向一直为向右
    {
        xnext=(x[0]+1)%8;
    }
}
```

```
void up()                          //下一步运动方向为向上时，判断下一步的位置
{
    xnext=x[0];
    ynext=y[0];
    if(dir0==3)                    //方向由左往上变化
    {
        if(y[0]==0||y[0]==3) ynext=4;
        else if(y[0]==6) ynext=5;
    }else if(dir0==1)              //方向由右往上变化
    {
        if(y[0]==0||y[0]==3) ynext=2;
        else if(y[0]==6) ynext=1;
    }else if(dir0==2)              //方向一直向上
    {
        if(y[0]==4||y[0]==1) ynext=y[0]+1;
        else ynext=y[0]-1;
    }
}

void left()                        //下一步运动方向为向左时，判断下一步的位置
{
    xnext=x[0];
    ynext=y[0];
    if(dir0==0)                    //方向由下往左变化
    {
        if(y[0]==1) ynext=6;
        else if(y[0]==2) ynext=3;
        else
        {
            xnext=(x[0]-1)%8;
            if(y[0]==5) ynext=6;
            else if(y[0]==4) ynext=3;
        }
    }else if(dir0==2)              //方向由上往左变化
    {
        if(y[0]==1) ynext=0;
        else if(y[0]==2) ynext=6;
        else
        {
            xnext=(x[0]-1)%8;
            if(y[0]==5) ynext=0;
            else if(y[0]==4) ynext=6;
        }
```

```
    }else if(dir0==3)                    //方向一直向左
    {
        xnext=(x[0]-1)%8;
    }
}

void update()                    //更新贪吃蛇的坐标位置
{
    unsigned char j;
    if(xnext==a&&ynext==b)       //判断下一步是否吃到食物
    {
        length++;
        goal=1;                                  //将生成食物标志位置一

    }
    for(j=length-1;j>0;j--)      //位置坐标更新
    {                                                    //用前一位置坐标代替现
位置的坐标
        x[j]=x[j-1];
        y[j]=y[j-1];
    }
    x[0]=xnext;                                  //用 xnext，ynext 更新贪吃蛇头部的位置
坐标
    y[0]=ynext;
    dir0=dir;                                    //更新上一状态的运动方向
}


void timer_0() interrupt 1 //声明定时器 0 的中断服务程序
{
    flag=1;        //置 flag 标志为 1
}


void timer_1() interrupt 3 //声明定时器 1 的中断服务程序
{
    P46=!P46;              //P4.6 引脚取反
    m++;                           //全局变量 m 递增
    if(m==16) m=0;   //如果 m 等于 16，则 m 置为 0
    change=1;              //置 change 标志为 1
}

void control() interrupt 0    //声明外部中断 0 中断服务程序
{
```

```
    if(mode==0)                          //每次进入贪吃蛇模式，都从头开始玩起
    {
        mode=1;
        //初始化贪吃蛇模式参数
        x[0]=0;
        y[0]=0;
        dir=1;
        dir0=1;
        length=1;
        goal=1;
        change=1;
    }
    else
    {
        mode=0;
        m=0;
    }          //控制遥控是否进入贪吃蛇模式
}

void main()
{
    unsigned char i=0;                  //定义本地字符型变量 char
    unsigned char c1_new,c1_old=1;          //声明字符型变量
    SPCTL=(SSIG<<7)+(SPEN<<6)+(DORD<<5)+(MSTR<<4)
                +(CPOL<<3)+(CPHA<<2)+SPEED_4;
    //给寄存器 SPCTL 赋值
    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
    AUXR1=0x08;                             //将 SPI 接口信号线切换到第 3 组引脚上
    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;                              //使能外部中断 0
    IT0=1;                              //设置外部中断 0 为低电平触发
    EA=1;                                       //CPU 允许响应中断请求
    P47=1;
    while(1)  //无限循环
    {
```

```
        if(flag==1)          //如果 flag 为 1，表示定时器 0 中断
        {
            flag=0;          //将 flag 标志清零
            switch(mode)
            {
                case 0:                 //演示下雨模式
                    for(i=0;i<8;i++)          //轮流导通 7 段数码管，需要 8 次
                    {
                        //控制其中一个数码管，送管选和段码
                        if(m<7-i)
                        {
                            seg7scan(i,2);
                        }else if(m<i+8)
                        {
                            seg7scan(i,(m-7+i)%2);
                        }else
                        {
                            seg7scan(i,3);
                        }
                    }
                    break;
                case 1:
                    if(length==4)                   //如果贪吃蛇已经达到四段，则游戏结束
                    {
                        good();
                    }else
                    {
                        //扫描按键
                        P0=0xF0;                    //将 P0.0~P0.3 拉低，在读 P0.4~P0.7
前，发 F
                        delay(60);                  //延迟读
                        c1_new=P0&0xF0;             //得到矩阵按键的信息
                        if(c1_new!=c1_old)
                        {
                            c1_old=c1_new;                //将新按键的状态变量保存作
为旧的按键
                            if(c1_new!=0xF0)
                            {
                                P0=0xFE;
                                delay(60);
                                c1_new=P0;
                                if(c1_new==0xde)                    //如果扫描到按了 4 号
按键，改变方向为向左
                                {
```

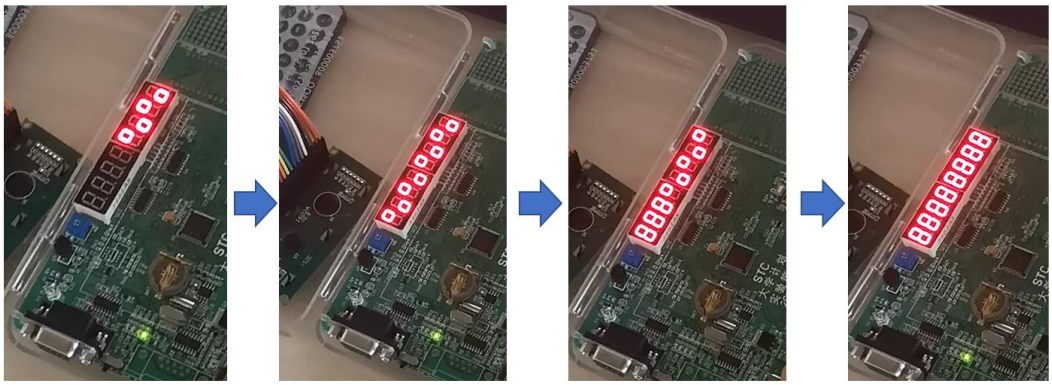
```
                    if(dir!=1)
                    {
                        dir0=dir;
                        dir=3;    //改变方向
                    }
                }
                P0=0xFD;
                delay(60);
                c1_new=P0;
                switch(c1_new)
                {
                    case 0xed:                    //如果扫描到按了 5 号按键，改变方向为向下
                        if(dir!=0)
                        {
                            dir0=dir;
                            dir=2;    //改变方向
                        }
                        break;
                    case 0xdd:                    //如果扫描到按了 1 号按键，改变方向为向上
                        if(dir!=2)
                        {
                            dir0=dir;
                            dir=0;    //改变方向
                        }
                        break;
                    default:break;
                }
                P0=0xFB;
                delay(60);
                c1_new=P0;
                if(c1_new==0xdb)              //如果扫描到按了 6 号按键，改变方向为向左
                {
                    if(dir!=3)
                    {
                        dir0=dir;
                        dir=1;    //改变方向
                    }
                }
            }
        }
        if(goal==1)                           //若需要生成新
```

食物，则产生食物的坐标

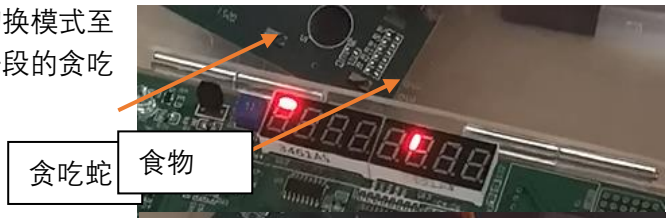
```
                    {
                        a=rand()%8;
                        b=rand()%7;
                        for(i=0;i<length;i++)          //保证新食物与贪吃蛇不
重叠
                        {
                            if(a==x[i]&&b==y[i])
                            {
                                a=(a+1)%8;
                                b=(b+1)%7;
                                break;
                            }
                        }
                        goal=0;
                    }
                    seg7scan(a,b);

                    switch(dir)                    //根据前后状态判断下一步的位置
坐标 xnext，ynext
                    {
                        case 0:down();break;
                        case 1:right();break;
                        case 2:up();break;
                        case 3:left();break;
                        default:break;
                    }
                    if(change==1)                  //每触发一次定时器 0 中断更新一次
贪吃蛇位置
                    {
                        update();
                        change=0;                  //更改状态更新标志位
                    }

                    for(i=0;i<length;i++)
                    {
                        seg7scan(x[i],y[i]);           //显示贪吃蛇当前位置
                    }
                }break;
            }
        }
    }
}
```

【实验结果】

下载了程序之后，单片机初始状态 mode=0，此时为下雨演示模式，重复演示图示竖向的下雨。

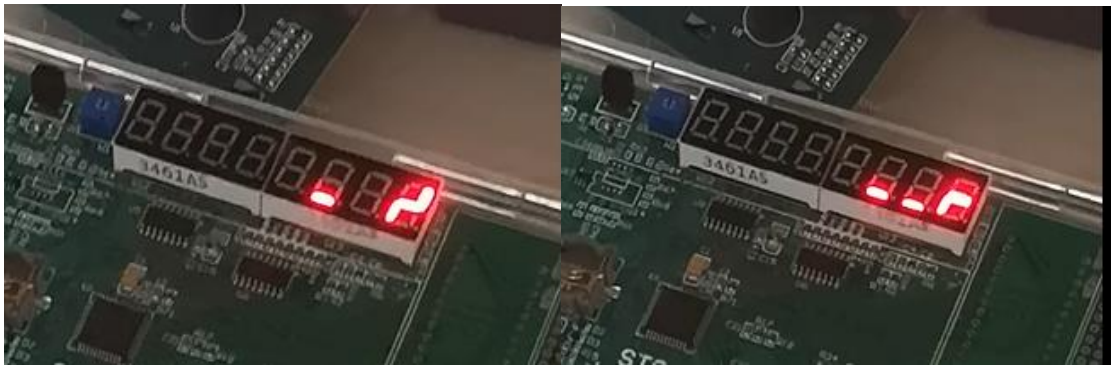


按下 SW17 通过触发外部中断 0，切换模式至贪吃蛇模式，数码管初始状态显示一段的贪吃蛇以及生成的食物，如图：



通过扫描键盘上的按键控制蛇运动的方向直到蛇吃掉食物，贪吃蛇长度会变成两段，同时产生位置随机的新食物。

继续控制贪吃蛇吃食物，直到贪吃蛇的长度变为 4 段，贪吃蛇游戏结束并显示 good。



此时如果想要重新开始，或是回到下雨模式，可以通过触发外部中断 0 来切换模式。
#good game！#