

西安科技大学计算机学院
计算机科学系

《计算机微机原理与接口技术》
课程设计报告

题 目：抢答器

学生姓名：李 琦

学 号：17408070220

班 级：计科 1701

完成日期：2019 年 7 月 4 日

指导教师：桑亚群

课程设计小组成员名单及分工

姓名	学号	主要完成内容	备注
李琦	17408070220	独立按键和矩阵键盘功能设计、数码管显示、LCD 液晶屏显示、LED 点阵动态倒计时显示	

具体内容

一、课程设计的内容及其要求

功能描述：实现八选手抢答，抢答时间可以使用按键调整。采用 4 位数码管显示选手编号和抢答倒计时时间，设置主持人按键三个，功能依次为：复位、暂停、开始。

基础要求：

- ① 用独立按键作为主持人按键；
- ② 用数码管显示选手编号和倒计时时间；
- ③ LCD 显示选手姓名及其他信息。
- ④ 用 4*4 键盘作为抢答者按键和抢答时间设定按键；

发挥要求：

- ① 增加抢答人数，如 16 路抢答等；
- ② 用 LCD 屏按照多个抢答者同时抢答时按键的先后顺序，循环显示多个抢答者姓名等信息；
- ③ 用 LED 点阵显示动态倒计时。

二、设计方案

总体电路设计方案：本次课程设计的题目为抢答器。①、4x4 矩阵键盘为选手抢答时的按键，由 P1 工作组控制。②、4 个独立按键为主持人的控制键，分别为开始、暂停、复位、设置倒计时时间，由 P3⁰-P3³ 控制。③、数码管的显示部分，采用 74LS138 译码器控制位选，输入端为 P2²-P2⁴，74HC245 锁存器控制段选，输入端为 P0 工作组。④、LCD1602 液晶屏的显示，数据\指令输入端为 P0 工作组，控制端为 P2⁵-P2⁷。⑤、LED8x8 点阵，由 74HC595 锁存器来控制点阵的行选，输入端为 P3⁴-P3⁶，P0 工作组控制点阵的列选。

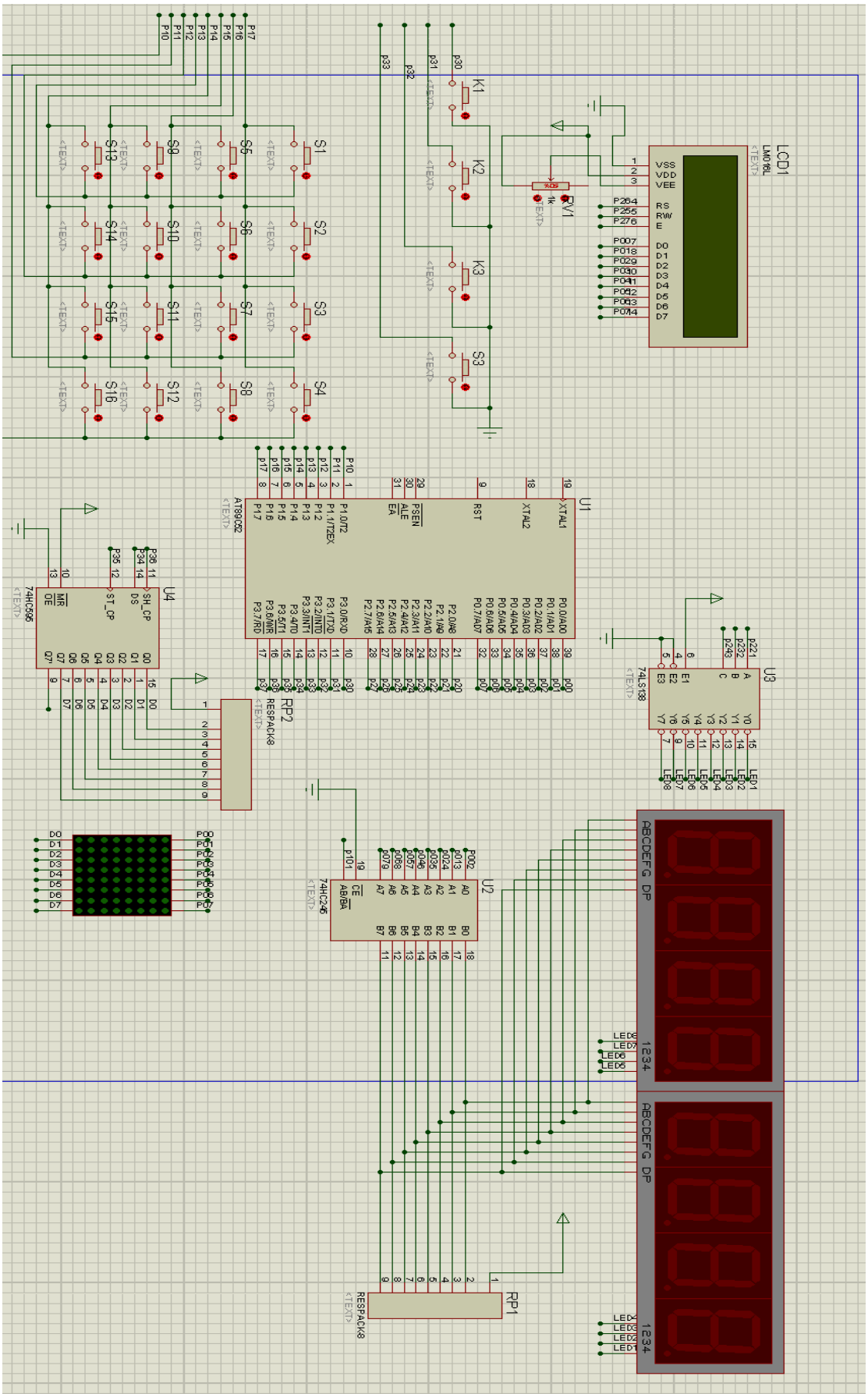


图 1 总电路图

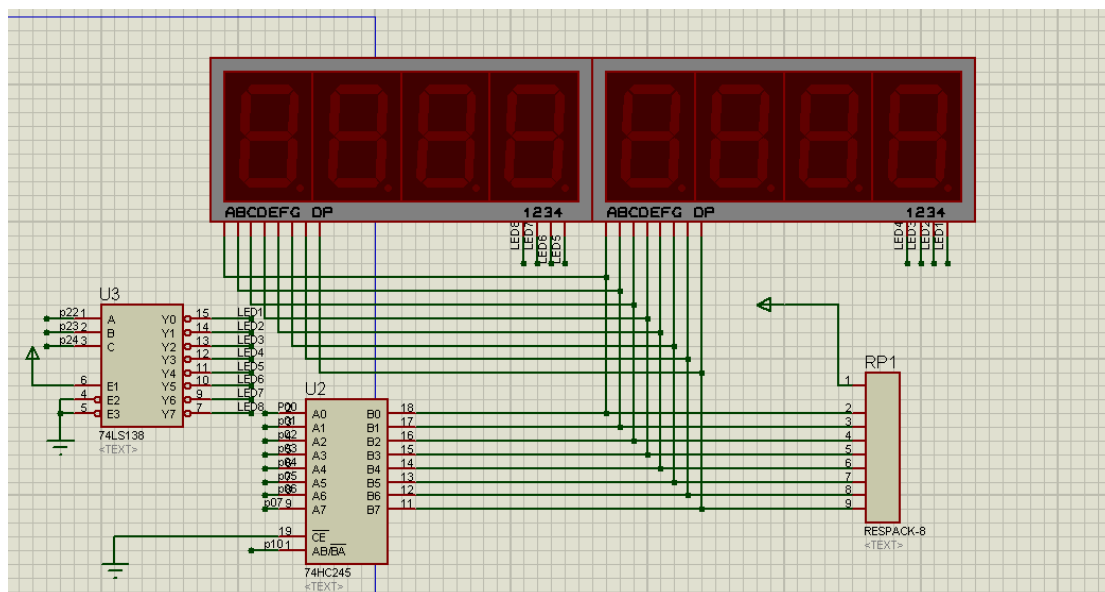


图 2 数码管显示部分电路

数码管显示功能模块：通过 74HC245 锁存器来实现段选，本次的数码管为共阴极，所以 P0 工作组为高电平时，数码管就实现了段选。通过 74LS138 译码器来实现位选，当 $P2^2$ - $P2^4$ 的电平发生改变时，数码管就实现了位选。

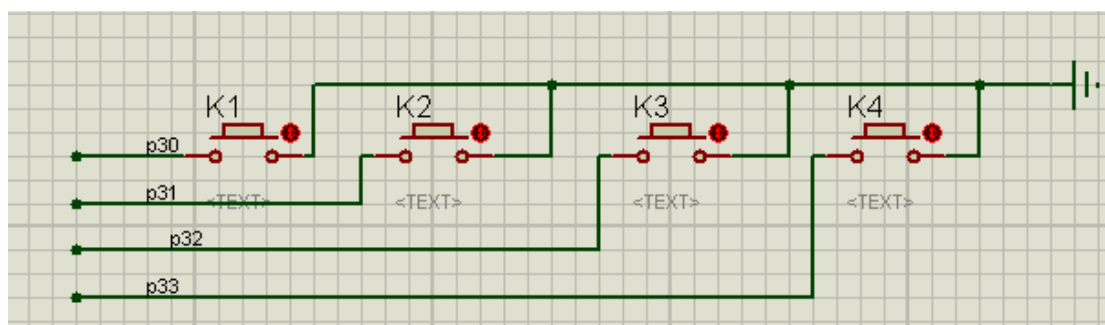


图 3 独立按键电路

独立按键模块：主持人通过 4 个独立按键，来实现倒计时的暂停（K1）、开始（K2）、复位（K3）以及时间设置（K4）（需要配合矩阵键盘的输入）。

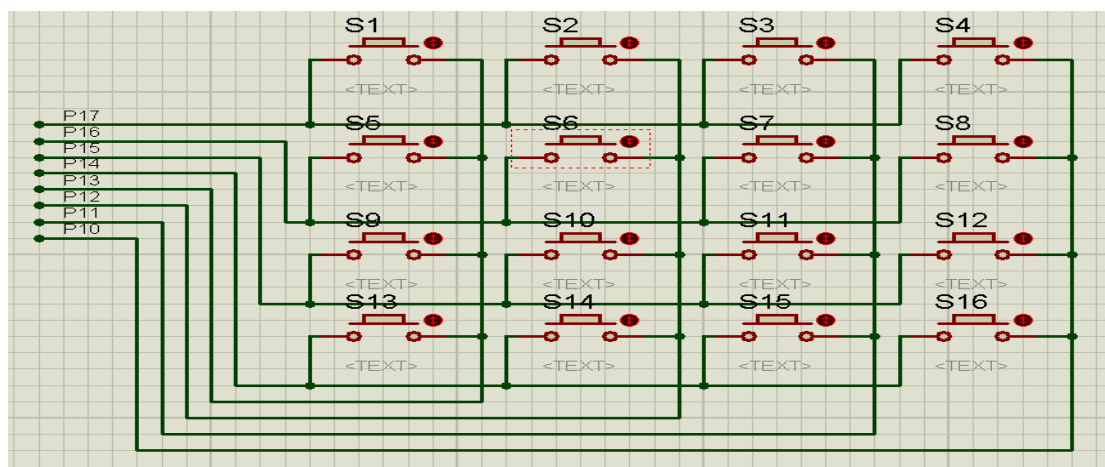


图 4 矩阵键盘电路

4x4 矩阵键盘模块：开关的左端由 $P1^7$ - $P1^4$ 控制，右端由 $P1^3$ - $P1^0$ 控制。键盘扫描函数中，我采用了行扫描法，来检测按键是否被按下，以及对应的键值是多少。

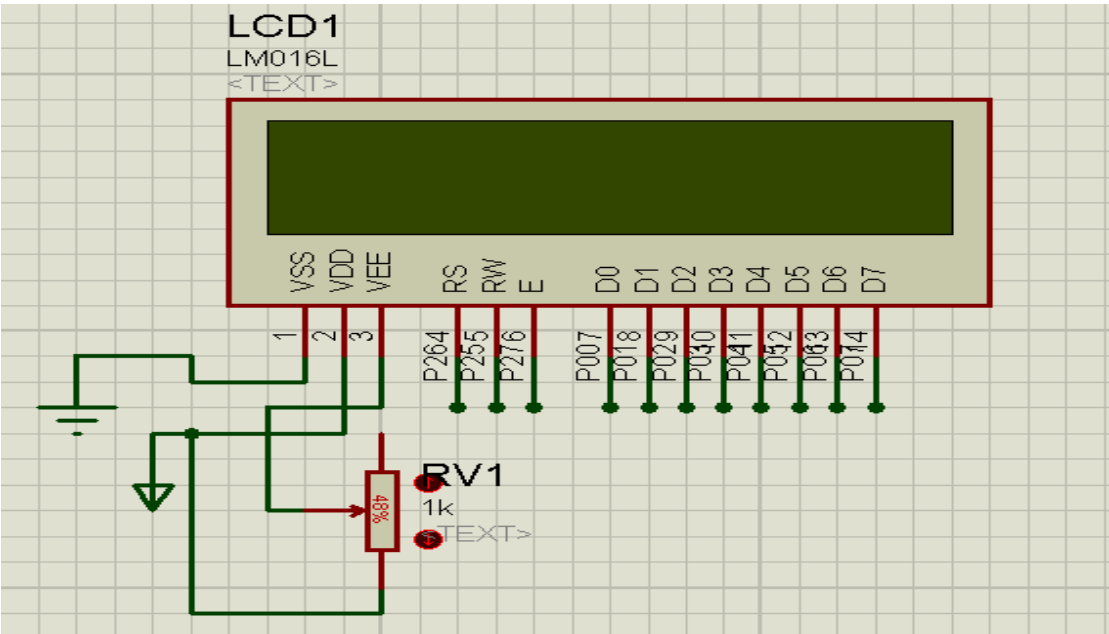


图 5LCD1602 液晶显示电路

LCD 液晶显示模块：P0 工作组控制数据/指令的输入，RS 为数据/命令选择端，RW 为读/写选择端，E 为使能信号。

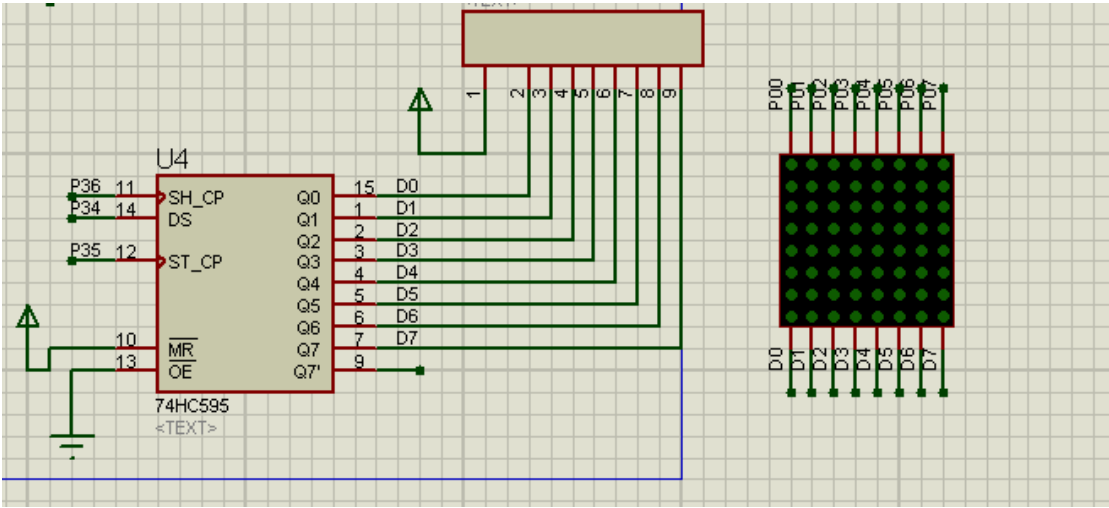


图 6LED 点阵

8x8LED 点阵显示模块:通过 74HC595 移位寄存器，在每个移位寄存器时钟输入（SH_CP）上升信号时，将数据串行输入到移位寄存器(ST_CP)。串行输入 8 个数据之后，来一个存储寄存器时钟输入上升信号，将数据存入到存储寄存器中。LED 点阵的行选由 D0-D7 来控制，列选由 P0 工作组控制。

三、实现功能说明

本次课程设计，我实现了抢答器的所有功能，其中增加抢答人数，我实现了15路抢答，一个题目最多可以由三个选手进行抢答，数码管上会短暂的显示选手编号，LCD屏会将三名选手抢答的先后顺序循环滚动显示（选手的名字及其编号）。当倒计时还剩最后10s时，会在8x8LED点阵上动态显示倒计时，数码管也同时显示倒计时。主持人的按键为开始、暂停、复位、设置倒计时时间。当主持人按下复位时，倒计时时间为20s，点阵图不亮，LCD的第二行会清空。

矩阵键盘模块：我采用了行扫描法，将第一行的电平置0，然后屏蔽行位信号，开始扫描列上的信号，如果列上有对应的键位按下，那么列位的电平置为低电平，从而得到该键值。以此类推，也可以得到其余的键值。

倒计时模块及15路选手抢答模块：采用定时计数器0中断，倒计时的默认值为20，这个数字也可以更改。通过按下K4，然后按下矩阵键盘上的S1-S10（1-9）来设置十位上的数字，或者可以使用S15，S16将时间减一或加一。倒计时的数字最大为99s。当倒计时开始时（定时计数器0开始工作），选手可以通过S1-S16按键来抢答。

主持人按键模块：暂停（K1）、开始（K1）、复位（K3）、设置倒计时时间（K4），当K1按下时，倒计时数字暂停，并且此时不允许抢答。当K2按下时，开始倒计时，允许选手抢答。当K3按下时，倒计时数字恢复为20，LCD屏的第二行会清空，LED点阵图不亮。

LCD液晶显示屏模块：程序初始化时，会在第一行显示“QIANG-DA-QI”，如果有人抢答成功，会在液晶显示屏的第二行显示对应的编号及名字。一道题最多可以由三个人抢答，三个人抢答之后，倒计时暂停，LCD会滚动循环显示一圈。

LED点阵动态倒计时模块：当倒计时小于10s时，LED点阵会动态显示9-0，同时数码管也会显示倒计时（二者之间互不干扰）。

四、结论

在本次课程设计中，虽然实现了全部的功能要求，但是仍然存在许多不足的地方，比如：虽然实现了三路同时抢答，但是同一个人可以抢答三次，然后抢答就结束了，这个问题目前还没与解决，想到的解决办法就是，加两次判断，每一次都判断这个人是否之前有抢答，如果之前有抢答，那么他的抢答信号将不写入LCD液晶屏。

遇到的问题有：刚开始LCD屏与数码管复用时，数码管会显示乱码，但是液晶屏是正常的。解决的办法，我将控制数码管的位选端单独做了位声明，而不

是直接声明整个 P2 工作组的值，然后在控制液晶屏工作时，我将数码管的位选放到了一个未用到的数码管上，那么同时也就解决了 P0 工作复用的问题，（P0 既控制数码管的段选，还有 LCD 液晶屏的数据指令的输出输入）。

在 LED 点阵与数码管复用时，刚开始会出现大面积的闪烁、数字闪烁问题，解决的办法是，在每次 LED 显示之前我都通过 74hc595 将 D0-D7 置为低电平，（当 D0-D7 为高电平时，P0-P7 为低电平时，相应的二极管会点亮），这样就解决了大面积闪烁的问题。将 LED 点阵显示输出时的延时减小，这样就解决了数码管和 LED 点阵的闪烁问题。

希望改进的地方：LED 动态倒计时是从 9-0，没有实现从设置的数字开始动态计时。这个功能可以改善，将两位数拆开，显示原理与显示一个数字相同，但是因为控制 led 点阵的芯片是 74hc595，那个芯片只能串行输入数据，所以函数实现起来特别麻烦 Q0-Q7 会有 256 种情况，如果想优化显示算法，可以改接线路，将 led 点阵的行选放到 P1 工作组，那么行选的所有情况可以用十六进制表示，不用 74hc595 锁存。

五、心得体会

本次课程设计中，由于自己之前完全不了解单片机，所以开始设计时，头脑中没有一点思绪。通过查阅了相关的资料，以及看视频学习单片机的工作原理。功能模块逐个加入，最终实现了一个简易的抢答器。

在设计过程中，遇到了许多的问题，使我认识到了自己能力上的不足，理论和实践之间还是有一定的差距。还有加深了对本专业的认识，增强了自己的动手实践能力。微机原理与接口技术这门课程，虽然指令十分复杂，但是通过学习这本书，了解了微机的工作原理以及大部分的芯片设置。课程设计刚开始时，对单片机的认识少之又少，通过看视频和书本学习，逐步对单片机的工作原理掌握，本次芯片使用了 STC89c52，该芯片的存储容量相比于 51 单片机有了一个很大的提升。单片机的学习是硬件学习的基础，只有将基础牢固，更高级的硬件电路学起来才不会吃力。

在本次学习中，我还学习了对于 keil c51 和 proteus 软件的使用，Proteus 自从有了单片机也就有了开发系统，随着单片机的发展开发系统也在不断发展。keil 是一种先进的单片机集成开发系统。它代表着汇编语言单片机开发系统的最新发展，首创多项便利技术，将开发的编程/仿真/调试/写入/加密等所有过程一气呵成，中间不须任何编译或汇编。说起课程设计，我认为最重要的就是做好设计的预习，认真的研究老师给的题目，选一个自己有兴趣的题目。其次，老师对实验的讲解要一丝不苟的去听去想，因为只有都明白了，做起设计就会事半功倍，如果没弄明白，就迷迷糊糊的去选题目做设计，到头来一点收获也没有。最后，

要重视程序的模块化，修改的方便，也要注重程序的调试，掌握其方法。在全组人竭尽全力，老师的精心指导下，程序基本编写成功，这是我们共同努力的结果，在享受我们成果之时，不得不感慨单片机的重要性与高难度性，所以为期两周的单片机课程设计没有浪费我我们学到了很多知识，也让我们对单片机有了更深一步的了解。

唯一有些遗憾的是，在本次课程设计中，硬件电路不需要自己设计，更不用自己连线，减少了工作量，但是对于单片机的认识并不到位。毕竟，电路图与实物连线之间还是有一定差距，只有通过自己的去实践，才能发现其中的乐趣。

附录（抢答器源代码）

```
#include <reg52.h>
#define uchar unsigned char
#define uint unsigned int
uchar table[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f}; //段码表
uchar code table1[]="QIANG-DA-QI";
uchar code table2[]="1-Lee 2-Clearlove7 3-Iboy 4-Meiko 5-Ray 6-Sount 7-Langxi 8-MLXG 9-Karsa
10-Xiaohu 11-Uzi 12-Ming 13-Jacklove 14-Baolan 15-Rookie "; //选手编号及名字
uchar code table3[]=""; //空白字符组，用来清空 LCD 屏
void delay(uint x); //延时函数
uint key_scan(); //矩阵键盘扫描函数
void display_num(uint i); //选手的编号在数码管上显示
void display_time(uint j); //倒计时在数码管上的显示
void display_led(uint j); //LED 的动态倒计时显示
void lcd_display(uchar i); //LCD 液晶屏的显示函数
void lcd_yiwei(); //LCD 液晶屏的滚动循环显示
void init(); //定时中断的初始化
void init_lcd(); //LCD 液晶屏初始化
void start_keyscan(); //主持人开始按键
void reset_keyscan(); //主持人复位按键
void pause_keyscan(); //主持人暂停按键
void setting_keyscan(); //主持人设置倒计时时间函数
void write_com(uchar com); //LCD 液晶屏写入指令函数
void write_data(uchar dat); //LCD 液晶屏写入数据函数
void led_in(); //LED 点阵图设计
void led_in1(); //LED 点阵图设计
void led_in2(); //LED 点阵图设计
void led_in3(); //LED 点阵图设计
void led_in4(); //LED 点阵图设计
void led_in5(); //LED 点阵图设计
void led_in6(); //LED 点阵图设计
void led_in7(); //LED 点阵图设计
void led_in8(); //LED 点阵图设计
void led_in9(); //LED 点阵图设计
void led_in10(); //LED 点阵图设计
void led_in11(); //LED 点阵图设计
void led_in12(); //LED 点阵图设计
void led_in13(); //LED 点阵图设计
void led_in14(); //LED 点阵图设计
void led_in15(); //LED 点阵图设计
void led_in16(); //LED 点阵图设计
void led_in17(); //LED 点阵图设计
void led_in18(); //LED 点阵图设计
void led_in19(); //LED 点阵图设计
void led_in20(); //LED 点阵图设计
void led_in21(); //LED 点阵图设计
void led_in22(); //LED 点阵图设计
void led_in23(); //LED 点阵图设计
void led_in24(); //LED 点阵图设计
uchar shi, ge;
uchar flag, i, j;
uint num=20, cnt;
sbit K1=P3^0; //暂停键位
sbit K2=P3^1; //开始键位
sbit K3=P3^2; //复位键位
sbit K4=P3^3; //设置倒计时时间键位
sbit rs=P2^6; //LCD 控制端声明
sbit rw=P2^5; //LCD 控制端声明
sbit lcden=P2^7; //LCD 控制端声明
```

```

sbit A0=P2^2;           //数码管位选端
sbit A1=P2^3;           //数码管位选端
sbit A2=P2^4;           //数码管位选端
sbit ser=P3^4;          //数据输入端
sbit rck=P3^5;          //存储寄存器时钟输入
sbit srck=P3^6;         //移位寄存器时钟输入

```

```

void main()
{
    uchar k=0;
    init();
    init_lcd();
    for(i=0;i<11;i++)
    {
        write_data(table1[i]);
        delay(10);
    }
    write_com(0x80+0x40);
    while(1)
    {
        start_keyscan();
        pause_keyscan();
        reset_keyscan();
        setting_keyscan();
        display_time(num);
        if(num<=9)
        {
            P0=0xff;
            display_led(num);
            led_in24();
        }
        if(flag==1)
        {
            i=0;
            display_num(i);
            i=key_scan();
            if(i!=0)
            {
                display_num(i);
                delay(300);
                lcd_display(i);
                k++;
                delay(10);
            }
        }
        if(k==3)
        {
            flag=0;
            TR0=0;
            lcd_yiwei();
            k=0;;
        }
    }
}

```

```

void init()           //定时计数器 0 初始化
{
    TMOD=0x01;
    TH0=(65536-50000)/256;
}

```

```

    TL0=(65536-50000)%256;
    EA=1;
    ET0=1;
    TR0=0;
    led_in24();
}

void init_lcd()          //LCD 液晶显示屏初始化设置
{
    A0=1;
    A1=1;
    A2=1;
    write_com(0x38);
    write_com(0x08);
    write_com(0x01);
    write_com(0x06);
    write_com(0x0c);
}

void write_com(uchar com)      //LCD 液晶屏写入指令
{
    A0=1;
    A1=1;
    A2=1;
    rs=0;
    rw=0;
    lcden=0;
    P0=com;
    lcden=1;
    delay(5);
    lcden=0;
}

void write_data(uchar dat)     //LCD 液晶屏写入数据
{
    rs=1;
    rw=0;
    lcden=0;
    P0=dat;
    lcden=1;
    delay(5);
    lcden=0;
}

void lcd_display(uchar i)      //LCD 液晶屏显示选手编号及姓名
{
    switch(i)
    {
        case(0):
        {
            write_com(0x80+0x40);
            for(j=0;j<40;j++)
            {
                write_data(table3[j]);
                delay(10);
            }
            break;
        }
        case(1):
    }
}

```

```

{
    for(j=0; j<6; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
    break;
}
case(2):
{
    for(j=6; j<19; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
    break;
}
case(3):
{
    for(j=19; j<26; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
    break;
}
case(4):
{
    for(j=26; j<34; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
    break;
}
case(5):
{
    for(j=34; j<40; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
    break;
}
case(6):
{
    for(j=40; j<48; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
    break;
}
case(7):
{
    for(j=48; j<57; j++)
    {
        write_data(table2[j]);
        delay(10);
    }
}

```

```

        }
        break;
    }
    case(8):
    {
        for(j=57; j<64; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(9):
    {
        for(j=64; j<72; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(10):
    {
        for(j=72; j<82; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(11):
    {
        for(j=82; j<89; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(12):
    {
        for(j=89; j<97; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(13):
    {
        for(j=97; j<109; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(14):
    {

```

```

        for(j=109; j<119; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
    case(15):
    {
        for(j=119; j<129; j++)
        {
            write_data(table2[j]);
            delay(10);
        }
        break;
    }
}
}

void lcd_yiwei()                //LCD 液晶屏循环移位显示
{
    uchar a;
    i=key_scan();
    if(i>0&&i<16)
    {
        delay(2000);
        for(a=0; a<=20; a++)
        {
            write_com(0x18);
            delay(300);
        }
        a=0;
        delay(300);
        for(a=0; a<=20; a++)
        {
            write_com(0x1c);
            delay(300);
        }
    }
}

void start_keyscan()            //主持人开始键
{
    if (K1==0)
    {
        delay(10);
        if (K1==0)
        {
            flag=1;
            TR0=1;
        }
    }
}

void pause_keyscan()            //主持人暂停键
{
    if (K2==0)
    {
        delay(10);
    }
}

```

```

        if (K2==0)
        {
            flag=0;
            TR0=0;
        }
    }
}

void reset_keyscan()                //主持人复位键
{
    if (K3==0)
    {
        delay(10);
        if (K3==0)
        {
            TR0=0;
            flag=0;
            num=20;
            i=0;
            display_time(num);
            lcd_display(0);
            write_com(0x80+0x40);
            led_in24();
        }
    }
}

void setting_keyscan()              //主持人设置倒计时时间
{
    uchar set_shi;
    uchar i;
    if (K4==0)
    {
        delay(10);
        if (K4==0)
        {
            while(!K4)
            {
                i=key_scan();
                if (i>=0&& i<10)
                {
                    set_shi=i*10;
                    num=set_shi;
                    display_time(num);
                }
                if (i==14)
                {
                    delay(200);
                    if (i==14) num--;
                }
                if (i==15)
                {
                    delay(200);
                    if (i==15) num++;
                }
                if (num==100) num=0;
                display_time(num);
            }
        }
    }
}

```

```

}
}

void delay(uint x)                //延时函数括号里面的数字，代表延时的时间（ms）
{
    uint i, j;
    for(i=x; i>0; i--)
        for(j=120; j>0; j--);
}

void display_num(uint i)          //数码管显示选手编号
{
    if(i>=0 && i<10)
    {
        A0=0;
        A1=1;
        A2=0;
        P0=table[i];
        delay(5);
    }
    if(i>=10)
    {
        shi=i/10;
        ge=i%10;
        A0=1;
        A1=1;
        A2=0;
        P0=table[shi];
        delay(300);
        A0=0;
        A1=1;
        A2=0;
        P0=table[ge];
        delay(5);
    }
}

void display_time(uint j)         //数码管显示倒计时
{
    if(j>=0 && j<10)
    {
        A0=0;
        A1=0;
        A2=0;
        P0=table[j];
        delay(5);
        A0=1;
        A1=0;
        A2=0;
        P0=table[0];
        delay(5);
    }
    if(j>=10)
    {
        shi=j/10;
        ge=j%10;
        A0=0;
        A1=0;
        A2=0;
    }
}

```



```

        P0=table[ge];
        delay(5);
        A0=1;
        A1=0;
        A2=0;
        P0=table[shi];
        delay(5);
    }
}

uint key_scan()                //矩阵键盘扫描，将得到的键值返回
{
    uchar n, keynum;
    P1=0x7f;
    n=P1;
    n&=0x0f;
    if (n!=0x0f)
    {
        delay(10);
        P1=0x7f;
        n=P1;
        n&=0x0f;
        if (n!=0x0f)
        {
            switch(n)
            {
                case (0x07):keynum=0;break;
                case (0x0b):keynum=1;break;
                case (0x0d):keynum=2;break;
                case (0x0e):keynum=3;break;
            }
        }
    }
    P1=0xbf;
    n=P1;
    n&=0x0f;
    if (n!=0x0f)
    {
        delay(10);
        P1=0xbf;
        n=P1;
        n&=0x0f;
        if (n!=0x0f)
        {
            switch(n)
            {
                case (0x07):keynum=4;break;
                case (0x0b):keynum=5;break;
                case (0x0d):keynum=6;break;
                case (0x0e):keynum=7;break;
            }
        }
    }
    P1=0xdf;
    n=P1;
    n&=0x0f;
    if (n!=0x0f)
    {
        delay(10);
    }
}

```

```

        P1=0xdf;
        n=P1;
        n&=0x0f;
        if (n!=0x0f)
        {
            switch(n)
            {
                case (0x07):keynum=8;break;
                case (0x0b):keynum=9;break;
                case (0x0d):keynum=10;break;
                case (0x0e):keynum=11;break;
            }
        }
    }
    P1=0xef;
    n=P1;
    n&=0x0f;
    if (n!=0x0f)
    {
        delay(10);
        P1=0xef;
        n=P1;
        n&=0x0f;
        if (n!=0x0f)
        {
            switch(n)
            {
                case (0x07):keynum=12;break;
                case (0x0b):keynum=13;break;
                case (0x0d):keynum=14;break;
                case (0x0e):keynum=15;break;
            }
        }
    }
    return keynum;
}

void display_led(uint num)                //LED 点阵，从 9-0 动态倒计时显示
{
    A0=1;
    A1=1;
    A2=1;
    P0=0xff;
    switch(num)
    {
        case(0):
        {
            led_in();
            delay(1);
            led_in1();
            delay(1);
            led_in2();
            delay(1);
            led_in3();
            delay(1);
            break;
        }
        case(1):
        {

```

```
        led_in4();
        delay(1);
        break;
    }
    case(2):
    {
        led_in5();
        delay(1);
        led_in6();
        delay(1);
        led_in7();
        delay(1);
        led_in8();
        delay(1);
        break;
    }
    case(3):
    {
        led_in9();
        delay(1);
        led_in10();
        delay(1);
        led_in11();
        delay(1);
        led_in12();
        delay(1);
        break;
    }
    case(4):
    {
        led_in13();
        delay(1);
        led_in14();
        delay(1);
        led_in15();
        delay(1);
        led_in16();
        delay(1);
        break;
    }
    case(5):
    {
        led_in17();
        delay(1);
        led_in18();
        delay(1);
        led_in19();
        delay(1);
        led_in20();
        delay(1);
        break;
    }
    case(6):
    {
        led_in();
        delay(1);
        led_in18();
        delay(1);
        led_in19();
```

```

        delay(1);
        led_in20();
        delay(1);
        break;
    }
    case(7):
    {
        led_in21();
        delay(1);
        led_in22();
        delay(1);
        led_in23();
        delay(1);
        led_in3();
        delay(1);
        break;
    }
    case(8):
    {
        led_in();
        delay(1);
        led_in18();
        delay(1);
        led_in19();
        delay(1);
        led_in3();
        delay(1);
        break;
    }
    case(9):
    {
        led_in17();
        delay(1);
        led_in18();
        delay(1);
        led_in19();
        delay(1);
        led_in3();
        delay(1);
        break;
    }
}
}

void led_in23()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<7;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xf7;

```

```

}

void led_in22()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<7;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xef;
}

void led_in21()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<7;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xdf;
}

void led_in20()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    for(i=0;i<5;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xfb;
}

```

```

void led_in19()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xf7;
}

```

```

void led_in18()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xef;
}

```

```

void led_in17()          //LED 点阵图的设计

```

```

{
uchar i;
for(i=0;i<4;i++)
{
    ser=1;
    srck=1;
    srck=0;
}
for(i=0;i<3;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
ser=1;
srck=1;
srck=0;
rck=1;
rck=0;
P0=0xdf;
}

void led_in16()          //LED 点阵图的设计
{
uchar i;
for(i=0;i<8;i++)
{
    ser=1;
    srck=1;
    srck=0;
}
rck=1;
rck=0;
P0=0xfb;
}

void led_in15()          //LED 点阵图的设计
{
uchar i;
for(i=0;i<3;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
ser=1;
srck=1;
srck=0;
for(i=0;i<4;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
rck=1;
rck=0;
P0=0xf7;
}

```

```

void led_in14()          //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<4;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xef;
}

```

```

void led_in13()          //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<4;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    for(i=0;i<4;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xdf;
}

```

```

void led_in12()          //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<8;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xfb;
}

```

```

void led_in11()          //LED 点阵图的设计
{

```



```

uchar i;
ser=1;
srck=1;
srck=0;
for(i=0;i<2;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
ser=1;
srck=1;
srck=0;
for(i=0;i<3;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
ser=1;
srck=1;
srck=0;
rck=1;
rck=0;
P0=0xf7;
}

void led_in10()          //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xef;
}

void led_in9()           //LED 点阵图的设计
{
    uchar i;

```

```

ser=1;
srck=1;
srck=0;
for(i=0;i<2;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
ser=1;
srck=1;
srck=0;
for(i=0;i<3;i++)
{
    ser=0;
    srck=1;
    srck=0;
}
ser=1;
srck=1;
srck=0;
rck=1;
rck=0;
P0=0xdf;
}

void led_in8()    //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<4;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xfb;
}

void led_in7()    //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;

```

```

        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xf7;
}

void led_in6()      //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<3;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xef;
}

void led_in5()      //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<2;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }

```

```

    }
    for(i=0;i<5;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xdf;
}

void led_in4()    //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<8;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xf7;
}

void led_in()    //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<8;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xdf;
}

void led_in1()    //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<6;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;

```

```

    rck=0;
    P0=0xef;
}

void led_in2()      //LED 点阵图的设计
{
    uchar i;
    ser=1;
    srck=1;
    srck=0;
    for(i=0;i<6;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    ser=1;
    srck=1;
    srck=0;
    rck=1;
    rck=0;
    P0=0xf7;
}

void led_in3()      //LED 点阵图的设计
{
    uchar i;
    for(i=0;i<8;i++)
    {
        ser=1;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
    P0=0xfb;
}

void led_in24()
{
    uchar i;
    for(i=0;i<8;i++)
    {
        ser=0;
        srck=1;
        srck=0;
    }
    rck=1;
    rck=0;
}

void timer_smg() interrupt 1      //定时计数器 0 每隔 50ms 产生一次中断
{
    uchar cnt;
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    cnt++;
    if(flag==1 && cnt==20)
    {
        num--;
    }
}

```

```
    cnt=0;
    if (num==0)
    {
        TR0=0;
        flag=0;
    }
}
```