

计算机与通信工程学院

单片机原理及应用实训总结报告

设计题目：抢答器控制模拟程序设计

学生姓名：

系 别：计算机与通信工程学院

专 业：信息工程

班 级：09-1

学 号：

指导教师：陈晓雷 副教授

2011 年 12 月 23 日

郑州轻工业学院

实 训 任 务 书

题目：抢答器控制模拟程序设计

专业、班级 信息工程 09-1 **学号** **姓名**

主要内容、基本要求、主要参考资料等：

基本要求：

具有 8 个抢答输入

显示抢答剩余时间（初始 10 秒）

显示抢答成功者

扩展要求：

抢答成功后有声音提示

时间分辨率小于 10ms

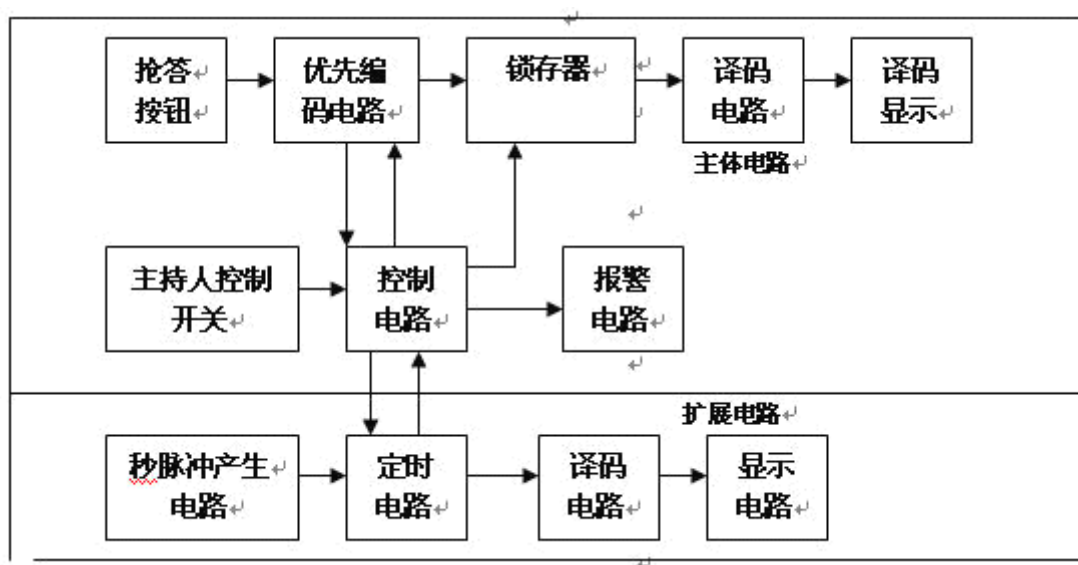
完善其他自定义功能

设计方案简述：

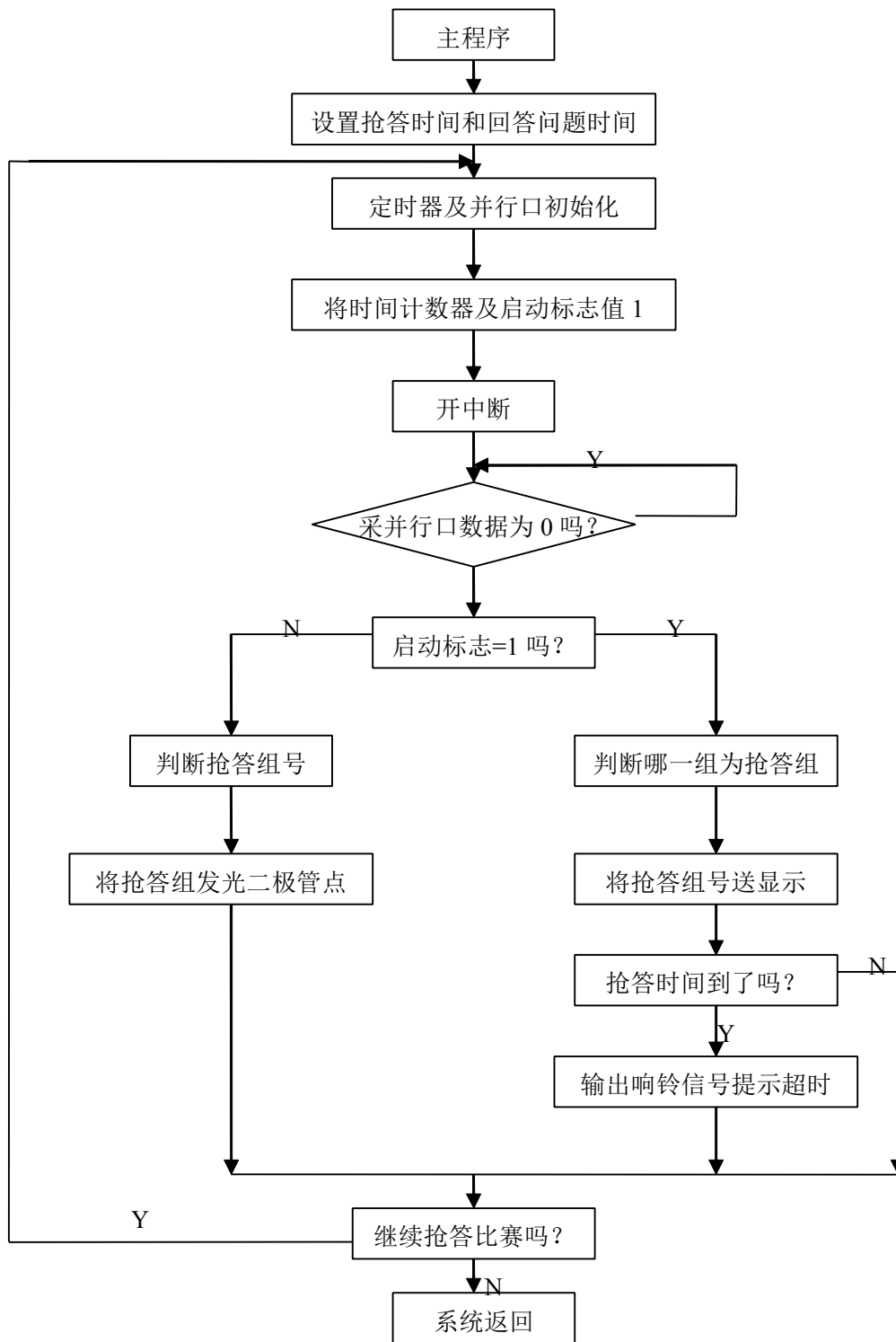
1. 定时抢答器的总体框图如下图所示，它由主体电路和扩展电路两部分组成。主体电路完成基本的抢答功能，即开始抢答后，当选手按动抢答键时，能显示选手的编号，同时能封锁输入电路，禁止其他选手抢答，扩展电路完成定时抢答的功能。

2. 定时抢答器的工作过程是：接通电源时，节目主持人将开关置于“消除”位置，抢答器处于禁止工作状态，编

号显示器灭灯，定时器倒计时，当定时时间到，却没有选手抢答时，系统报警，并封锁输入电路，禁止选手超时后抢答。当选手在定时时间内按动抢答键时，抢答器要完成以下四项工作：优先编码电路立即分辨出抢答者的编号，并由锁存器进行锁存，然后由译码显示电路显示编号；扬声器发出短暂的声响，提醒节目主持人注意；控制电路要对输入编码电路进行封锁，避免其他选手再次进行抢答；控制电路要使定时器停止工作，时间显示器上显示剩余的抢答时间，并保持到主持人将系统清零为止，当选手将问题回答完毕，主持人操作控制开关，使系统回复到禁止工作状态，以便进行下一轮抢答。



3 主程序流程图



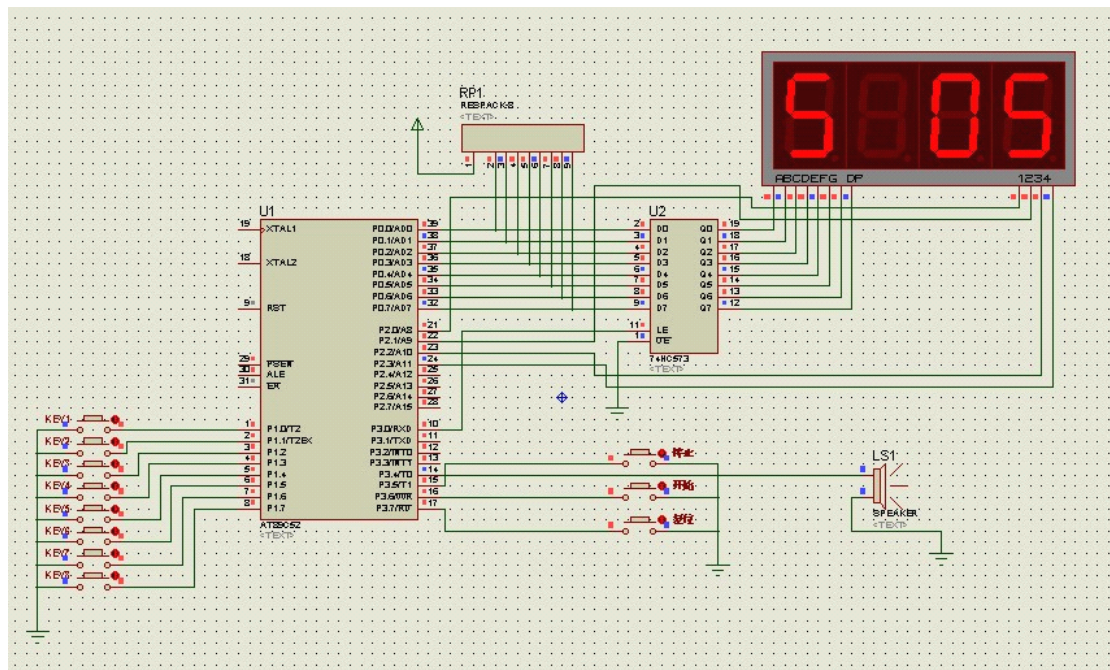
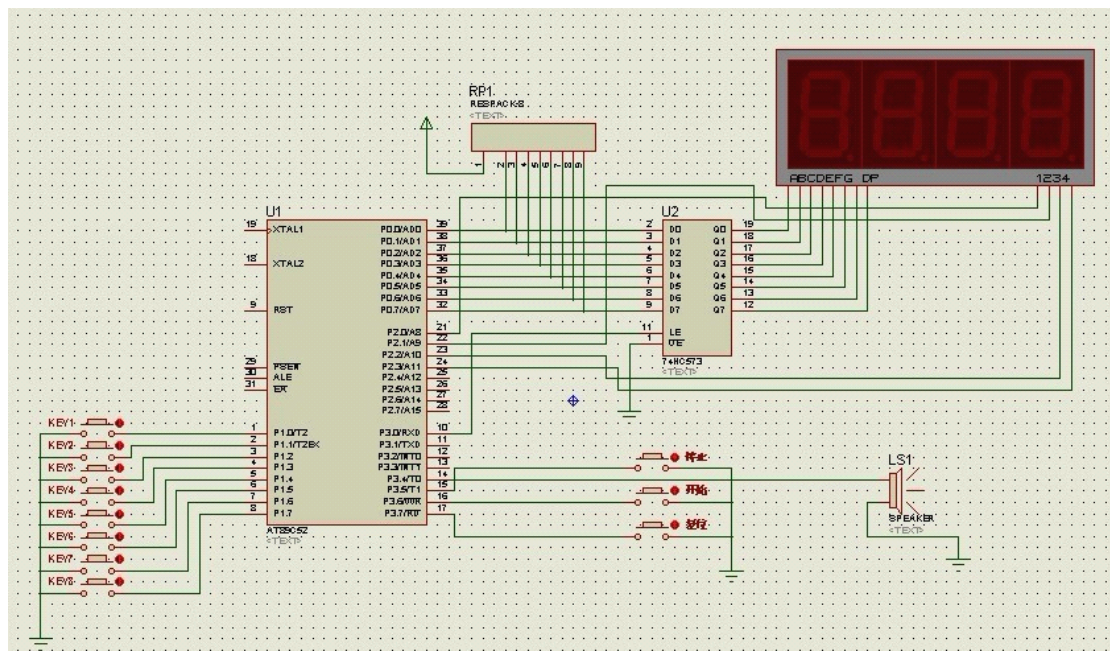
抢答器电路设计：

抢答电路的功能有两个：一是分辨选手按键的先后，并锁存抢答者的编号，供译码显示电路用；二是其他选手的按键操作无效。

选用优先编码器 74LS148 和 RS 锁存器, 74LS138, 74LS279 可完成上述功能。译码电路选用 74LS48 芯片工作原理为：

当主持人控制开关处于“清除”位置时，RS 触发器 R 为低电平，输出端（4Q~1Q）全部为低电平。于是 74LS48 的 BI=0, 显示器灭灯；74LS148 的选通输入 ST=0，74LS148 处于工作状态，此时锁存器电路不工作。当主持人开关拨到开“始位”置，优先编码电路和锁存电路同时开始工作，等待输入信号 I7, ..., I0。当有选手按下键时，74LS148 的输出 Y2Y1Y0=010, Yex=0。经 RS 锁存器后，CTR=1, BI=1，74LS279 处于工作状态，4Q3Q2Q=101。经过 74LS48 译码后，显示器显示选手编号。此外，CTR=1，是 74LS148 的 ST 为高电平，74LS148 处于禁止工作状态，封锁了其他选手按键的输入。当按下的键松开后，74LS148 的 Yes 为高电平，但由于 CTR 维持高电平不变，所以 74LS148 仍处于禁止工作状态，其他选手的输入不会被接收，保证了抢答者的优先性以及抢答电路的准确性。抢答完后主持人使抢答电路复位，以便进行下一轮抢答。

整体电路仿真图：



程序代码：

```
#include <reg52.h>
#define uint unsigned int
#define uchar unsigned char
/*-----
共阴极数码管编码表 0 - f 显示
```

*/


```
uchar code
table[]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x7f, 0x7c,
0x39, 0x5e, 0x79, 0x71};
sbit start= P3^6;
sbit stop=P3^5;
sbit reset = P3^7;
sbit key1 = P1^0;          //8 个抢答按键
sbit key2 = P1^1;
sbit key3 = P1^2;
sbit key4 = P1^3;
sbit key5 = P1^4;
sbit key6 = P1^5;
sbit key7 = P1^6;
sbit key8 = P1^7;
sbit buzzer = P3^4;
bit start_flag = 0;
bit stop_flag=0;
bit key1_flag = 0;        //按键抢答标志位，防止重复抢答
bit key2_flag = 0;
bit key3_flag = 0;
bit key4_flag = 0;
bit key5_flag = 0;
bit key6_flag = 0;
bit key7_flag = 0;
bit key8_flag = 0;
bit reset_flag = 0;      //复位键标志位，防止重复按下
bit action = 0;          //开始键标志位，防止重复按下

uchar second = 10;        //10 秒倒计时
uchar timer0_count = 0;  //定时器计数
uchar number = 0;         //抢答到的人

uchar number_display = 0;
/*-----*/
延时函数 延时 z 毫秒
/*-----*/

void delay(uint z)
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}

void fengming()           //蜂鸣函数
```

```
{
    //延时 1ms
    buzzer = 1;    //给 P3.7 口送高电平
    delay(5);    //延时 1ms
    buzzer = 0;
    delay(5);
}
/*-----
数码管显示驱动函数
-----*/
void display(uchar number, uchar second)
{
    uchar second_first, second_second;
    second_first = second / 10;    //计算出个位
    second_second = second % 10;    //计算出十位
    P2 = 0xfe;
    P0 = table[number];
    delay(2);
// P2 = 0xfd;
// P0 = 0x3f;
// delay(2);
    P2 = 0xfb;
    P0 = table[second_first];
    delay(2);
    P2 = 0xf7;
    P0 = table[second_second];
    delay(2);
}
/*-----
开始键扫描函数
-----*/
void start_keyscan()
{
    if(start == 0)
    {
        delay(8);    //延时防抖
        if((start == 0)&&(!start_flag))
        {
            start_flag = 1;
            action = 1;
            TR0 = 1;
        }
    }
    else
```



```
{
    start_flag = 0;
}
}

/*-----
八位抢答键扫描函数
-----*/

char key_scan8()
{
    if(key1 == 0)
    {
        delay(8);
        if((key1 == 0)&&(!key1_flag))           //防止按键重复被按下
        {
            key1_flag = 1;
            number  = 1;
            number_display = number;
            fengming();
        }
    }
    else
    {
        key1_flag = 0;
        number  = 0;
    }
    if(key2 == 0)
    {
        delay(8);
        if((key2 == 0)&&(!key2_flag))
        {
            key2_flag = 1;
            number  = 2;
            number_display = number;
            fengming();
        }
    }
    else
    {
        key2_flag = 0;
        number  = 0;
    }
    if(key3 == 0)
```

```
{
    delay(8);
    if((key3 == 0)&&(!key3_flag))
    {
        key3_flag = 1;
        number = 3;
        number_display = number;
        fengming();
    }
}
else
{
    key3_flag = 0;
    number = 0;
}
if(key4 == 0)
{
    delay(8);
    if((key4 == 0)&&(!key4_flag))
    {
        key4_flag = 1;
        number = 4;
        number_display = number;
        fengming();
    }
}
else
{
    key4_flag = 0;
    number = 0;
}
if(key5 == 0)
{
    delay(8);
    if((key5 == 0)&&(!key5_flag))
    {
        key5_flag = 1;
        number = 5;
        number_display = number;
        fengming();
    }
}
else
{
```

```
    key5_flag = 0;
    number = 0;
}
if(key6 == 0)
{
    delay(8);
    if((key6 == 0)&&!key6_flag))
    {
        key6_flag = 1;
        number = 6;
        number_display = number;
        fengming();
    }
}
else
{
    key6_flag = 0;
    number = 0;
}
if(key7 == 0)
{
    delay(8);
    if((key7 == 0)&&!key7_flag))
    {
        key7_flag = 1;
        number = 7;
        number_display = number;
        fengming();
    }
}
else
{
    key7_flag = 0;
    number = 0;
}
if(key8 == 0)
{
    delay(8);
    if((key8 == 0)&&!key8_flag))
    {
        key8_flag = 1;
        number = 8;
        number_display = number;
        fengming();
    }
}
```

```
    }
}
else
{
    key8_flag = 0;
    number = 0;
}

if(number_display != 0)
{
    return 1;
}
else
{
    return 0;
}
}
/*-----*/
复位键扫描函数
/*-----*/

void reset_keyscan()
{
    if(reset == 0)
    {
        delay(8);
        if((reset == 0)&&(!reset_flag))
        {
            reset_flag = 1;
            number_display = 0;
            second=10;
        }
    }
    else
    {
        reset_flag = 0;
    }
}

/*-----*/
主函数
/*-----*/

void main()
{
    TMOD=0x01;                //设置为定时器 0 工作方式 1
```

```
TH0=(65536-50000)/256;
TL0=(65536-50000)%256;
EA=1;
ET0=1;
TR0=0;
buzzer = 0;
while(1)
{
    start_keyscan();
    reset_keyscan();
    while(action==1)
    {
        if(second<=5&&second>0)
            fengming();
        while(!key_scan8()&&!stop==0)
        {
            display(number_display, second);
            if(second == 0)
            {
                second = 10;
                break;
            }
        }
        TR0 = 0;
        display(number_display, second);
        action = 0;
        break;
    }
    display(number_display, second);
}

/*-----
中断服务函数
-----*/

void timer0() interrupt 1
{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    timer0_count ++;
    if(timer0_count == 20)
    {
        timer0_count = 0;
        second --;
    }
}
```

```
        if(second<=5&&second>0)
            fengming() ;
        if(second == 0)
        {
            TR0 = 0;
            number_display = 0;
            action = 0;
        }
    }
}
```

设计体会：

通过这次课程设计，我想说：为完成这次课程设计我们确实很辛苦，但苦中仍有乐，和同学们相互帮助，大学里三年的相处还赶不上这几天来的实在，我感觉我和同学们之间的距离更加近了。这个工程确实很累，LED 亮了起来，喇叭响起的是我一生以来最好听的声音，我们的心中就不免兴奋，不免激动。前面的种种艰辛这时就变成了最甜美的回忆！这次学习给我留下了深刻的印象，使我受益匪浅。而且对于论文的总体构思也有了很多经验，相信自己经过这一次的训练与学习，对于今后会更加努力，做得最好。

总之，通过这次课程设计的制作与写作让我懂得了很多，这次课程设计由我们的老师的指导然后我们自己去图书馆，上网查资料然后自己和我们这组的成员一起合作商讨完成的，原以为会很简单的，但无论在制作与写作过程中我们都遇到了理论课上所不曾遇到的问题，第一是综合性太强，这次制作不仅涉及到我们所学到的单片机，数字电子技术等课程的知识还涉及到很多我们

的选修及课外的知识。第二是理论联系实际性太强，把所学的搜集到得知识运用的实际中不是一件容易的事，不仅由元件从课本上的符号到实物的认识还是理论的测量结果与实际值的误差。

课程设计已结束，对我而言，知识上的收获重要，精神上的丰收更加可喜。让我知道了学无止境的道理。我们每一个人永远不能满足于现有的成就，人生就像在爬山，一座山峰的后面还有更高的山峰在等着你。挫折是一份财富，经历是一份拥有。这次课程设计必将成为我人生旅途上一个非常美好的回忆。

参考资料：《单片机实验与课程设计（proteus 仿真版）》浙江大学出版社（第一版）

