

## 第 28 课，一个按键选择播放六首音乐

上一课，我们学习了用 4 个键控制播放 3 首音乐，另一个键停止播放。但是我们只能选择 3 首音乐，是不是太少了？因此这一课，我们用一个按键 K1 控制选择播放更多的音乐，共 6 首乐曲。如果按下另外三个按键中的任意一个，音乐就立即停止。

```
#define uchar unsigned char //定义一下方便使用
#define uint unsigned int
#define ulong unsigned long
#include <reg52.h> //包括一个 52 标准内核的头文件
```

```
char code dx516[3] _at_ 0x003b;//这是为了仿真设置的
```

```
sbit BEEP=P1^7; //喇叭输出脚
sbit P10=P1^0;
sbit K1= P3^2;
sbit K2= P3^5;
sbit K3= P2^4;
sbit K4= P2^5;
```

```
uchar th0_f; //在中断中装载的 T0 的值高 8 位
```

```
uchar tl0_f; //在中断中装载的 T0 的值低 8 位
```

```
//T0 的值,及输出频率对照表
```

```
uchar code freq[36*2]={
    0xA9,0xEF,//00220HZ ,1 //0
    0x93,0xF0,//00233HZ ,1#
    0x73,0xF1,//00247HZ ,2
    0x49,0xF2,//00262HZ ,2#
    0x07,0xF3,//00277HZ ,3
    0xC8,0xF3,//00294HZ ,4
    0x73,0xF4,//00311HZ ,4#
    0x1E,0xF5,//00330HZ ,5
    0xB6,0xF5,//00349HZ ,5#
    0x4C,0xF6,//00370HZ ,6
    0xD7,0xF6,//00392HZ ,6#
    0x5A,0xF7,//00415HZ ,7
    0xD8,0xF7,//00440HZ 1 //12
    0x4D,0xF8,//00466HZ 1# //13
    0xBD,0xF8,//00494HZ 2 //14
    0x24,0xF9,//00523HZ 2# //15
    0x87,0xF9,//00554HZ 3 //16
    0xE4,0xF9,//00587HZ 4 //17
    0x3D,0xFA,//00622HZ 4# //18
```

```

0x90,0xFA,//00659HZ 5 //19
0xDE,0xFA,//00698HZ 5# //20
0x29,0xFB,//00740HZ 6 //21
0x6F,0xFB,//00784HZ 6# //22
0xB1,0xFB,//00831HZ 7 //23
0xEF,0xFB,//00880HZ `1
0x2A,0xFC,//00932HZ `1#
0x62,0xFC,//00988HZ `2
0x95,0xFC,//01046HZ `2#
0xC7,0xFC,//01109HZ `3
0xF6,0xFC,//01175HZ `4
0x22,0xFD,//01244HZ `4#
0x4B,0xFD,//01318HZ `5
0x73,0xFD,//01397HZ `5#
0x98,0xFD,//01480HZ `6
0xBB,0xFD,//01568HZ `6#
0xDC,0xFD,//01661HZ `7 //35
};

//定时中断 0,用于产生唱歌频率
timer0() interrupt 1
{
    TL0=tl0_f;TH0=th0_f; //调入预定值
    BEEP=~BEEP; //取反音乐输出 IO
}

//*****
//音乐符号串解释函数
//入口:要解释的音乐符号串,输出的音调串,输出的时长串
changedata(uchar *song,uchar *diao,uchar *jie)
{
    uchar i,i1,j;
    char gaodi;//高低+/-12 音阶
    uchar banyin;//有没有半个升音阶
    uchar yinchang;//音长
    uchar code jie7[8]={0,12,14,16,17,19,21,23}; //C 调的 7 个值

    *diao=*song;
    for(i=0,i1=0;;)
    {
        gaodi=0; //高低=0
        banyin=0;//半音=0
        yinchang=4;//音长 1 拍
        if((*song+i)=='|' || (*(song+i)==' ') i++;
    }
}

```

```
//拍子间隔和一个空格过滤
```

```
switch(*(song+i))  
{  
    case ',': gaodi=-12;i++;//低音  
    break;  
  
    case '^': gaodi=12;i++;//高音  
    break;  
}  
  
if(*(song+i)==0) //遇到 0 结束  
{  
    *(diao+i1)=0; //加入结束标志 0  
    *(jie+i1)=0;  
    return;  
}  
  
j=*(song+i)-0x30; i++; //取出基准音  
j=jie7[j]+gaodi; //加上高低音
```

```
yinc: switch(*(song+i))  
{  
    case '#': //有半音 j 加一个音阶  
        i++;j++;  
        goto yinc;  
  
    case '-': //有一个音节加长  
        yinchang+=4;  
        i++;  
        goto yinc;  
  
    case '_': //有一个音节缩短  
        yinchang/=2;  
        i++;  
        goto yinc;  
  
    case '!': //有一个加半拍  
        yinchang=yinchang+yinchang/2;  
        i++;  
        goto yinc;
```

```

    }

    *(diao+i1)=j; //记录音符
    *(jie+i1)=yinchang; //记录音长
    i1++;
}
}
}
//*****
//奏乐函数
//入口:要演奏的音乐符号串
void play(uchar *songdata)
{
    uchar i,c,j=0;
    uint n;
    uchar xdata diaodata[112]; //音调缓冲
    uchar xdata jiedata[112]; //音长缓冲

    changedata(songdata,diaodata,jiedata); //解释音乐符号串
    TR0=1;
    for(i=0;diaodata[i]!=0;i++) //逐个符号演奏
    {
        tl0_f=freq[diaodata[i]*2]; //取出对应的定时值送给 T0
        th0_f=freq[diaodata[i]*2+1];
        for(c=0;c<jiedata[i];c++) //按照音长延时
        {
            for(n=0;n<32000;n++);
            if((!K1)||(!K2)||(!K3)||(!K4))//发现按键,立即退出播放
            {
                TR0=0;
                return;
            }
        }
        TR0=0;
        for(n=0;n<500;n++); //音符间延时

        TR0=1;
    }
    TR0=0;
}

//仙剑
uchar code xianjian[]={
"|3_3_3_2_3-|2_3_2_2_,6,6_,7_|12_1_,7,6_,5_|,6---|"

```

```

"3_3_3_2_3.6|5_6_5_5_22_3|45_4_32_1_|3.--3_|"
"67_6_55_3_|5--3_5_|26_5_32_3_|3---|"
"26_6_6-|16_6_66_7_|`17_6_76_7_|3.--3_|"
"67_6_55_3_|5--3_5_|67_6_76_7_|3---|"
"26_6_6-|16_6_66_7_|`17_6_7.5_|6---|"
};
uchar code song3[]={
"5-5_3_2_1_|3---|6-6_4_2_1_"
",7--,5_|1.3_5.1_|,7.3_5_5_|"
"6.7_`1.6_|6_5_5-3_2_|1.1_13_2_|"
"1.1_12_3_|2.1_,62_3_|2-- ,5|"
"1.3_5.1_|,7.3_55_|6.7_`1.6_|"
"6_5_5-3_2_|1.1_13_2_|1.1_12_3_"
"2.,6_,71_2_|1--"
};
//世上只有妈妈好
uchar code mamahao[]={
"6.5_35|`16_5_6-|35_6_53_2_|1_,6_5_3_2-|"
"2.3_55_6_|321-|5.3_2_1_,6_1_|,5--"
};
//我是一只菠萝
uchar code boluo[]={
"1-|2_4_3_2_55|5_6_3_4_22|2_4_3_2_1_`1_7_6_|5_4_2_3_11|"
"2_4_3_2_55|5_6_3_4_22|2_4_3_2_1_5_2_3_|1---|"
};
//小星星
uchar code xingxing[]={
"1155|665-|4433|221-|"
"5544|332-|5544|332-|"
"1155|665-|4433|221-|"
};

//老鼠爱大米
uchar code dami[]={
"321-|3.2_1-|3231|3_6.5-|3_5_665|65-3_2_|122_3_2_|12--|"
};
//一个按键选择播放六首音乐
void main(void) // 主程序
{
    uchar i;

    TMOD = 0x01; //使用定时器 0 的 16 位工作模式
    TR0 = 0;
    ET0 = 1; //定时器 0 中断

```

```

EA = 1;    //打开总中断

while(1)
{
    if(!K1)
    {
        while(!K1);

        switch(i)
        {
            case 0:
                play(xianjian); //播放音乐
                break;
            case 1:
                play(song3);    //播放音乐
                break;
            case 2:
                play(mamahao);  //播放音乐
                break;
            case 3:
                play(boluo);    //播放音乐
                break;
            case 4:
                play(xingxing); //播放音乐
                break;
            case 5:
                play(dami);     //播放音乐
                break;
        }
        i++;if(i==6)i=0;
    }
}

```

这里用了一个 switch 结构，判断当前按键的次序号并播放对应的音乐。

作业：

思考，为什么这里按键没有做去抖，也可以正常运行。