

第 22 课，音阶声音输出试验

这一课，我们不再输出简单嘀声了，而是要输出各种不同频率的音乐声。先输出基本的音阶，12345671。

为了输出准确的音阶频率，我们需要用定时器输出来控制蜂鸣器的驱动，这里用的 T0。

我们再每一次定时器中断溢出时取反 P17 引脚，以形成频率驱动蜂鸣器，定时器 0 工作在 16 位方式，需要在中断里重新置入初始值。这个值就决定了 P17 输出的频率。我们在程序里先做好了一张表，预先写好了每个音阶的频率需要设置的初始值。到时调入对应的值进去 T0，不断溢出时就 P17 可以输出对应的频率。

在这个程序里，我们自动地输出 8 个音符，每个音符保持 1 秒钟左右。

```
#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

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; //调入预定时值
    BEEP=~BEEP; //取反音乐输出 IO
    P2=~P2;
}

//音阶声音自动输出试验
void main(void) // 主程序
{
    ulong n;
    uchar i;

    uchar code jie8[8]={12,14,16,17,19,21,23,24}; //1234567`1 八个音符在频率表中的位置

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

    while(1)
    {
        for(i=0;i<8;i++) //循环播放 8 个音符
        {

```

```
        tI0_f=freq[jie8[i]*2];    //置一个音符的值
        th0_f=freq[jie8[i]*2+1];
        for(n=0;n<50000;n++); //延时 1 秒左右
    }
}
}
```

请编译，运行。可以听到 8 个音符不断循环输出。

作业：循环输出更多的音符