基于 MSP430G2 的音乐切换器

一、内容

通过 MSP430G2 播放自己所设置的歌曲，并通过按键 S2 切换另一首歌曲

二、思路与方法

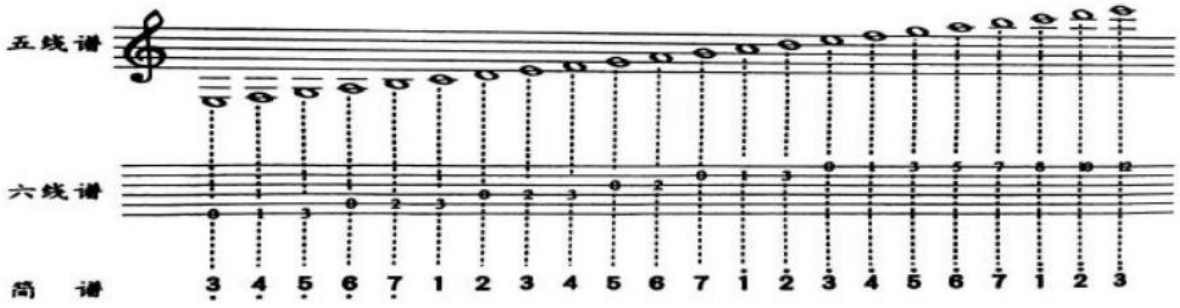
(1)思路：通过老师上课所讲的 F6638 音乐器播放实例，想利用 MSP430G2 来进行音乐播放，阅读网上单片机播放音乐例程并加以改编，并试想利用按键 S2 来进行歌曲的实时切换

(2)音乐：通过 MSP430 蜂鸣器音高音长对照表，将自己喜欢的音乐通过音乐简谱改成相应代码，利用播放函数 play_song()进行歌曲播放。演奏乐曲对于一个音符应该包括两个部分，声调用简单的延时-电平翻转来实现，改变了延时的时间就改变了声调，而时间通过计数比较来实现，当计数值相等时就跳出循环演奏下一个音符。

MSP430 蜂鸣器音高、音长对照表

注意：此表仅适用于时钟主频为 8MHz，TimerA3 分频！！

音高对照表：



0x53 0x4a 0x40 0x39 0x34 0x30 0x2b 0x26 0x22 0x20 0x1c 0x17 0x18 0x14 0x10 0x0c 0x09 0x05 0x01 无无

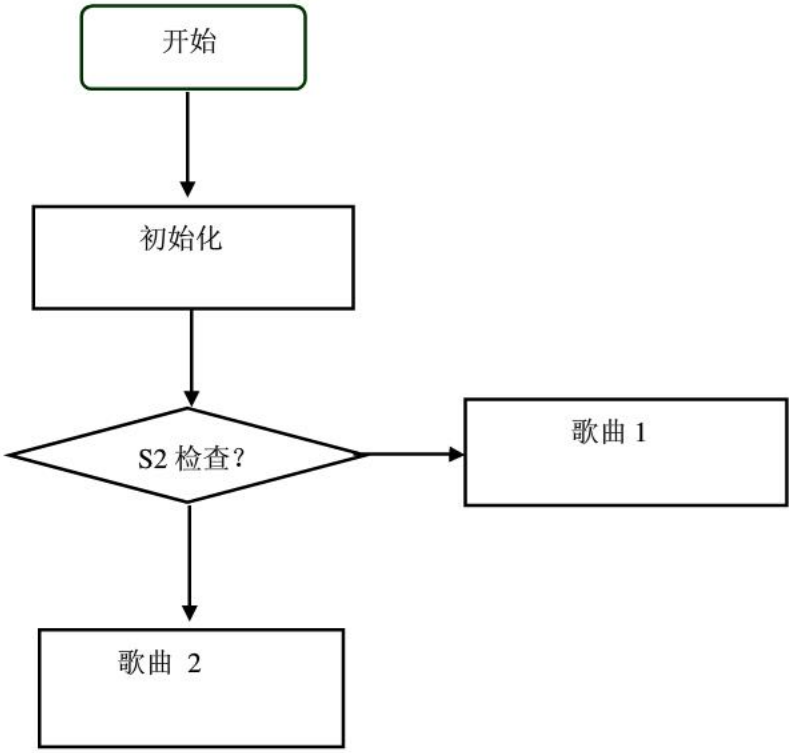
音长对照表：

四分音符　X: 0x40
八分音符　X: 0x20
十六分音符 X: 0x10

(3)按键 **S2:**通过中断服务、事件检测、事件处理函数,通过按键 S2 切换歌曲

(4)硬件：无源蜂鸣器、MSP430G2 单片机

有流程图：



三、程序调试

（1）遇到的问题与解决方法

按键 S2 切换歌曲开始不能进行实时切换，首先是我将实验是检测按键的程序进行整改加入主程序中 while(1)，这样只有长按 S2 键才能播放下一曲。后来查阅书籍关于 MSP430G2 中断服务的程序后，调用这些函数，并设置变量 i 放在两个音乐播放函数中，通过判断 i=1 或 0 进行选歌。

开始编曲时候并未按照音高、音长对照表进行编曲，所以导致歌曲无调子，后在搜集到资料后进行整改进行改曲。开始蜂鸣器声音略小，后发现是正负导线接反所致。

（2）程序段

```
/*********************************************
  时钟频率务必为 8MHz，定时器为 8 分频
*********************************************/
#include  <msp430g2553.h>
typedef unsigned char uchar;

#include  "music.h"    //乐曲 1
#include  "te.h"       //乐曲 2
```

```
#define   Buzzer        BIT3
#define   Buzzer_Port   P2OUT
#define   Buzzer_DIR    P2DIR

uchar counter;
void Play_Song(void);
void Delay_Nms(uchar n);
void ss(void);
void P1_IODect();
void P13_Onclick();
static int i=0;


/***************主函数***************/
void main(void)
{
      WDTCTL = WDTPW + WDTHOLD;//关闭看门狗
           P1DIR |= BIT0;
           P1OUT |= BIT0;
           P1REN |= BIT3;
           P1OUT |=BIT3;
           P1DIR &=~BIT3;
            P1IES |= BIT3;
            P1IE |= BIT3;
    BCSCTL1=CALBC1_8MHZ;    //晶振选择 DCO 中的 8MHz
    DCOCTL=CALDCO_8MHZ;     //选择系统主时钟为 8MHz//
    CCTL0 = CCIE;
    CCR0 = 7200;             //设定拍速
    TACTL |= TASSEL_2 + ID_3;      //TimerA 定时器分频要选 8 分频
    Buzzer_DIR |= Buzzer;       //设置控制蜂鸣器的 IO 方向为输出
    _EINT();         //打开全局中断
    //循环演奏歌曲

    while(1)
    {if (i==0)//按键没按下
        {Play_Song();}
     else {ss();}
    }
}


/*****************************************
函数名称：TimerA_ISR
功    能：定时器 A 的中断服务函数
```

```
*****************************************/
#pragma vector = TIMER0_A0_VECTOR
__interrupt void TimerA_ISR(void)
{
    counter++;

}
/*****************************************
函数名称：PORT1_ISR
功    能：响应 p1 口的外部中断服务
*****************************************/
#pragma vector =PORT1_VECTOR
__interrupt void PORT1_ISR(void)
{
   P1_IODect();
   P1IFG=0;
}
/*****************************************
函数名称：P1_IODect()
功    能：判断具体引发中断的 I/O,并调用相应 I/O 的中断事件处理函数
*****************************************/
void P1_IODect()
{
   unsigned int Push_Key=0;
   Push_Key=P1IFG&(~P1DIR);
   __delay_cycles(10000);
   if((P1IN&Push_Key)==0)
   {
       switch(Push_Key)
       {
       case BIT3:P13_Onclick();  break;
       default:                  break;
       }
   }

}
/*****************************************
函数名称：P13_Onclick()
功    能：事件处理函数
*****************************************/
void P13_Onclick()
{

   if(i==0)
```

```
    {
        i=1;
    }
    else
        i=0;
    P1OUT ^=BIT0;
}

/*****************************************
函数名称：Delay_Nms
功    能：延时 N 个 ms 的函数
参    数：n--延时长度
返回值  ：无
*****************************************/
void Delay_Nms(uchar n)
{
    uchar i,j;

    for( i = 0;i < n; i++ )
    {
        for( j = 0;j < 3;j++ )
          _NOP();
    }
}
/*****************************************
函数名称：Play_Song
        ss
*****************************************/
void Play_Song(void)
{
    uchar Temp1,Temp2;//Temp1 放音调决定了音调的高低，Temp2 放音长决定了某个音的演奏时间
    uchar addr = 0;   //SONG 数组中每两个为一组第一字节为音调，第二字节为音长

    counter = 0; //中断计数器清 0
    while(i==0)
    {
        Temp1 = songsong[addr++];
        if ( Temp1 == 0xFF )        //休止符
        {
            TACTL &=~MC_1;          //停止计数
            Delay_Nms(100);
        }
```

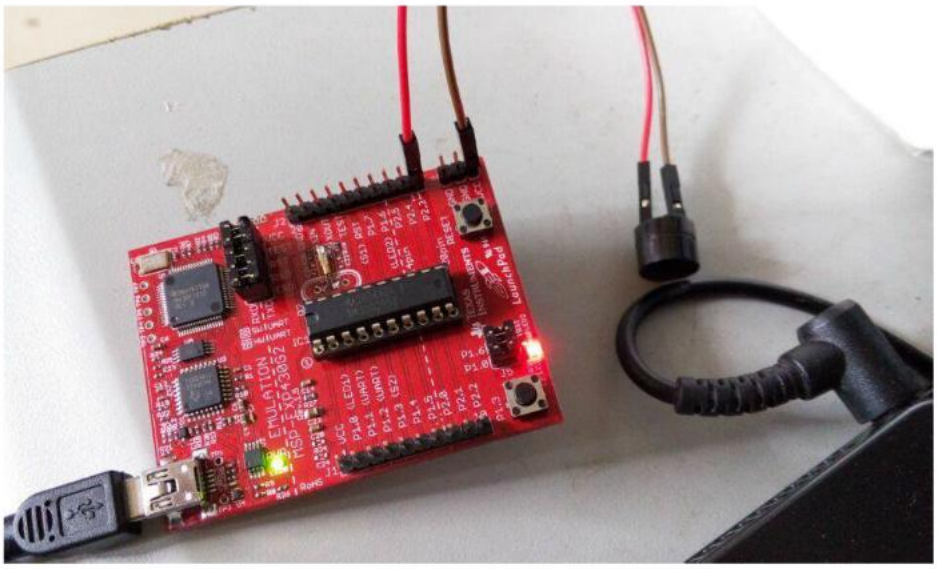
```
        else if ( Temp1 == 0x00 )   //歌曲结束符
        {
            return;
        }
        else
        {
            Temp2 = songsong[addr++];
            TACTL |=MC_1;           //开始计数
            while(1)
            {
                Buzzer_Port ^= Buzzer;

                Delay_Nms(Temp1);
                if ( Temp2 == counter )
                {
                    counter = 0;
                    break;
                }
            }
        }
    }
}
void ss(void)
{
    uchar Temp1,Temp2;//Temp1 放音调决定了音调的高低，Temp2 放音长决定了某个音的演奏时间
    uchar addr = 0;   //SONG 数组中每两个为一组第一字节为音调，第二字节为音长

    counter = 0; //中断计数器清 0
    while(i==1)
    {
        Temp1 = gg[addr++];
        if ( Temp1 == 0xFF )        //休止符
        {
            TACTL &=~MC_1;          //停止计数
            Delay_Nms(100);
        }
        else if ( Temp1 == 0x00 )   //歌曲结束符
        {
            return;
        }
        else
        {
```

```
            Temp2 = gg[addr++];
            TACTL |=MC_1;           //开始计数
            while(1)
            {
                Buzzer_Port ^= Buzzer;

                Delay_Nms(Temp1);
                if ( Temp2 == counter )
                {
                    counter = 0;
                    break;
                }
            }
        }
    }
}
const unsigned char songsong[]=        //歌曲 1 CCR0=7200，格式为：频率常数，节拍常数，频率常数，节拍常数，
{
0x26,0x20,0x20,0x20,0x20,0x20,0x26,0x10,0x20,0x10,0x20,0x80,
0x26,0x20,0x30,0x20,0x30,0x20,0x39,0x10,0x30,0x10,0x30,0x80,
0x26,0x20,0x20,0x20,0x20,0x20,0x1c,0x20,0x20,0x80,0x2b,0x20,
0x26,0x20,0x20,0x20,0x2b,0x10,0x26,0x10,0x2b,0x80,0x26,0x20,
0x30,0x20,0x30,0x20,0x39,0x10,0x26,0x10,0x26,0x60,0x40,0x10,
0x39,0x10,0x26,0x20,0x30,0x20,0x30,0x20,0x39,0x10,0x26,0x10,
0x26,0x80,0x26,0x20,0x2b,0x10,0x2b,0x10,0x2b,0x20,0x30,0x10,
0x39,0x10,0x26,0x10,0x2b,0x10,0x2b,0x20,0x2b,0x40,0x40,0x20,
0x20,0x10,0x20,0x10,0x2b,0x10,0x26,0x30,0x30,0x80,0x18,0x20,
0x18,0x20,0x26,0x20,0x20,0x20,0x20,0x40,0x26,0x20,0x2b,0x20,
0x30,0x20,0x30,0x20,0x1c,0x20,0x20,0x20,0x20,0x80,0x1c,0x20,
0x1c,0x20,0x1c,0x20,0x30,0x20,0x30,0x60,0x39,0x10,0x30,0x10,
0x20,0x20,0x2b,0x10,0x26,0x10,0x2b,0x10,0x26,0x10,0x26,0x10,
0x2b,0x10,0x2b,0x80,0x18,0x20,0x18,0x20,0x26,0x20,0x20,0x20,
0x20,0x60,0x26,0x10,0x2b,0x20,0x30,0x20,0x30,0x20,0x1c,0x20,
0x20,0x20,0x20,0x80,0x26,0x20,0x30,0x10,0x30,0x10,0x30,0x20,
0x39,0x20,0x26,0x10,0x2b,0x10,0x2b,0x20,0x2b,0x40,0x40,0x10,
0x40,0x10,0x20,0x10,0x20,0x10,0x2b,0x10,0x26,0x30,0x30,0x80,0x00,};

const unsigned char gg[]=        //歌曲 2  CCR0=7200
{       0x18, 0x30, 0x1C , 0x10, 0x20, 0x40, 0x1C , 0x10,0x18, 0x10, 0x20 , 0x10,
       0x1C, 0x10, 0x18 , 0x40,0x1C, 0x20, 0x20 , 0x20,0x1C, 0x20, 0x18 , 0x20,
       0x20, 0x80, 0xFF , 0x20,0x30, 0x1C, 0x10 , 0x18,0x20, 0x15, 0x20 ,
```

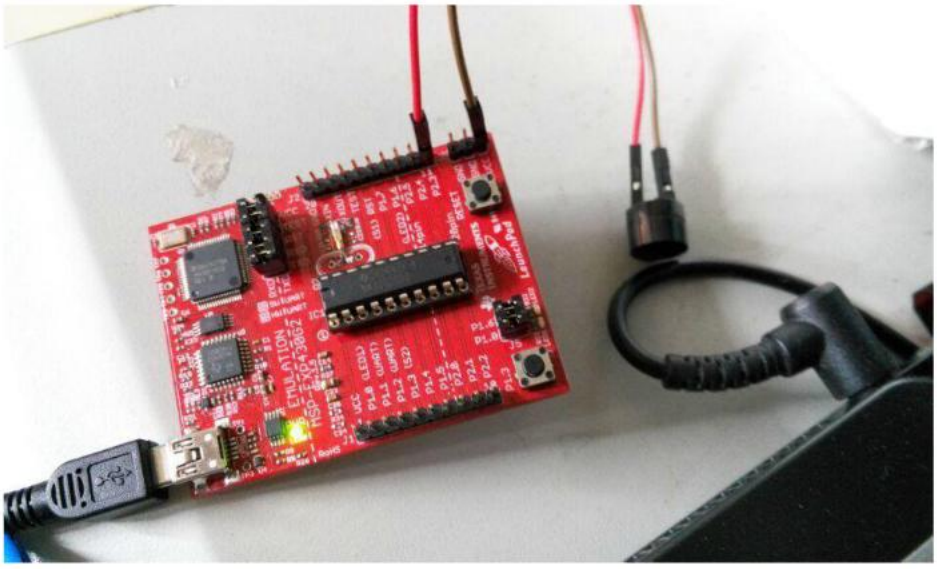
```
0x1C,
        0x20, 0x20, 0x20 , 0x26,0x40, 0x20, 0x20 , 0x2B,0x20, 0x26, 0x20 ,
0x20,
        0x20, 0x30, 0x80 , 0xFF,0x20, 0x20, 0x1C , 0x10,0x18, 0x10, 0x20 ,
0x20,
        0x26, 0x20, 0x2B , 0x20,0x30, 0x20, 0x2B , 0x40,0x20, 0x20, 0x1C ,
0x10,
        0x18, 0x10, 0x20 , 0x20,0x26, 0x20, 0x2B , 0x20,0x30, 0x20, 0x2B ,
0x40,
        0x20, 0x30, 0x1C , 0x10,0x18, 0x20, 0x15 , 0x20,0x1C, 0x20, 0x20 ,
0x20,
        0x26, 0x40, 0x20 , 0x20,0x2B, 0x20, 0x26 , 0x20,0x20, 0x20, 0x30 ,
0x80,
        0x20, 0x30, 0x1C, 0x10,0x20, 0x10, 0x1C , 0x10,0x20, 0x20, 0x26 ,
0x20,
        0x2B, 0x20, 0x30 , 0x20,0x2B, 0x40, 0x20 , 0x15,0x1F, 0x05, 0x20 ,
0x10,
        0x1C, 0x10, 0x20 , 0x20,0x26, 0x20, 0x2B , 0x20,0x30, 0x20, 0x2B ,
0x40,
        0x20, 0x30, 0x1C , 0x10,0x18, 0x20, 0x15 , 0x20,0x1C, 0x20, 0x20 ,
0x20,
        0x26, 0x40, 0x20 , 0x20,0x2B, 0x20, 0x26 , 0x20,0x20, 0x20, 0x30 ,
0x30,
        0x20, 0x30, 0x1C , 0x10,0x18, 0x40, 0x1C , 0x20,0x20, 0x20, 0x26 ,
0x40,
        0x13, 0x60, 0x18, 0x20,0x15, 0x40, 0x13 , 0x40,0x18, 0x80, 0x00
};
```

四、调试结果

红灯亮第一首歌：



红灯灭第二首歌：



结果说明：调试结果，达到了预期通过 S2 切换歌曲的功能

五、总结与体会

通过这次的课程设计，我学会了 MSP430 单片机定时器、中断服务模块、I/O 输入输出系统等。

起初的构想是制造一个音乐流水灯，将 LED 和所播放的音乐结合起来，产生动人的效果，但是由于在调试的过程中无法将 LED 闪烁的频率与音乐的声音有效的结合故改为通过按键切换歌曲。同时在基于 MSP430G2 的歌曲切换器的设计中也遇到了问题，但是通过查阅资料、同学交流、自己不断调试，最终达到预期效果。

其次，我觉得自己的设计仍有很大的不足，按照这个设计只能实现两首歌曲的切换，而不能进行多首且循环切换，所以仍有很大改进空间，希望以后有机会可以改进。

纪钢老师将制作单片机作品定为考核内容，这更有助我们将理论知识与实际应用相结合起来，而不是通过简单的考试。且通过单片机设计也帮助我们复习了所学的 C 语言知识，在结课以后也能有更深的印象，为将来学习或者工作打下了坚实的基础。最后，非常感谢纪老师的悉心指导，使我能够解决上课时的问题、发现单片机的乐趣所在，相信以后我也定会再与老师交流学习单片机。