

# 实验8 创新实验 单片机播放音乐

21220717温雁北 22210129彭佳豪 21220626王子阳

### 1.使用步骤

- 1.用proteus打开PlayMusic.pdsprj
- 2.点击左下角调试,播放音乐

### 2.目的和需求

在课程实验中我们学会了如何使用单片机驱动蜂鸣器Buzzer鸣叫, 由此可以设想:能否通过编写特定程序,改变蜂鸣器或音响的音高,使其播放一首完整音乐?

### 3.设计思路

### 3.1 器件选用:

CPU:AT89C51 SOUNDER:音响

### 3.2 音乐储存:

将音乐音符用c语言数组的形式存储

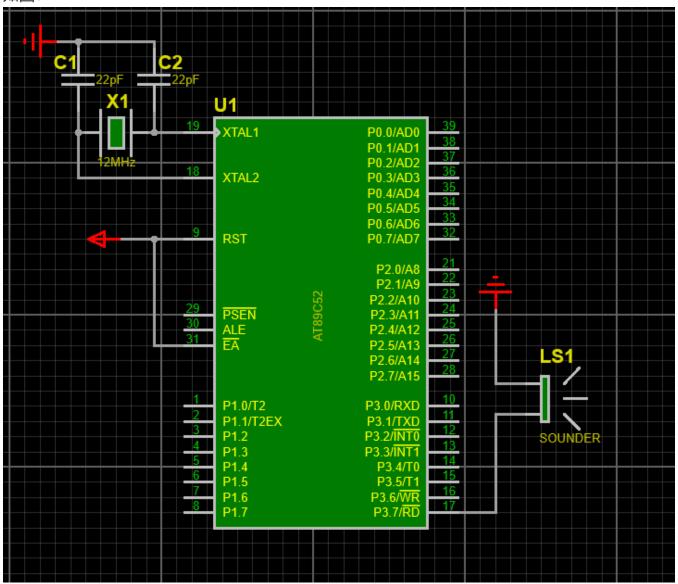
#### 3.3 播放:

通过单片机的定时器产生不同频率的方波信号,并控制IO口输出,从而模拟不同音高的音符。 通过预设的音符频率表和时值表,以及对音符编码的解析,代码可以按照指定的乐谱播放出相应 的旋律。核心思想是利用单片机的定时器产生精确的时序信号,控制输出引脚的高低电平,从而 产生不同频率的声音。

### 4.实现过程:

### 4.1 电路图连接与绘制:

如图:



### 4.2 音乐的获取:

选取了较为经典且简单的三首歌曲:

- 挥着翅膀的女孩
- 同一首歌
- 两只蝴蝶

#### 下载wav音频后,使用特定软件将wav转换为C语言数组:

₩ Wav文件数据读取(支持文件直拖)

文件目录: <code>|::\Users\lalham\Desktop\text2audio.v</code>

选择

X

unsigned char

 $[datt[32714]=\{0x00, 0x00, 0x$ 0, 0x00, 0x00 , 0x00, 0x00, 0xFF, 0xFF, 0xFE, 0xFF, 0xFD, 0xFF, 0xFC, 0xFF, 0xFB, 0xFF, 0xFB, OxFF, OxFB, OxFF, OxFB, OxFF, OxFC, OxFF, OxFC, OxFF, OxFE, OxFF, OxFF, OxFF, OxFF, O kff, 0xff, 0xff, 0xff, 0x01, 0x00, 0x03, 0x00, 0x05, 0x00, 0x05, 0x00, 0x05, 0x 00, 0x07, 0x00, 0x09, 0x00, 0x0A, 0x00, 0x0C, 0x00, 0x0E, 0x00, 0x09, 0x00, 0x0 7, 0x00, 0x07, 0x00, 0x06, 0x00, 0x04, 0x00, 0x01, 0x00, 0xFF, 0xFF, 0xFA, 0xFF , OxF8, OxFF, OxF6, OxFF, OxF5, OxFF, OxF2, OxFF, OxEF, OxFF, OxF0, OxFF, OxF2, OxFF, OxF6, OxFF, OxF4, OxFF, OxF3, OxFF, OxF7, OxFF, OxFD, OxFF, OxFC, OxFF, O kFA, 0xFF, 0xF9, 0xFF, 0xFD, 0xFF, 0x00, 0x00, 0x01, 0x00, 0x04, 0x00, 0x03, 0x 00, 0x06, 0x00, 0x0C, 0x00, 0x17, 0x00, 0x18, 0x00, 0x16, 0x00, 0x17, 0x00, 0x1 5, 0x00, 0x15, 0x00, 0x14, 0x00, 0x0E, 0x00, 0x05, 0x00, 0xFE, 0xFF, 0xF9, 0xFF , OxF7, OxFF, OxF3, OxFF, OxEB, OxFF, OxE2, OxFF, OxEO, OxFF, OxEE, OxFF, OxF6, OxFF, OxFB, OxFF, OxFB, OxFF, OxFE, OxFF, OxFC, OxFF, OxFC, OxFF, k04, 0x00, 0x09, 0x00, 0x01, 0x00, 0x00, 0x00, 0x02, 0x00, 0x02, 0x00, 0xFE, 0x FF, OxFB, OxFF, OxFA, OxFF, OxF6, OxFF, OxF9, OxFF, OxO5, OxO0, Ox15, OxO0, Ox1 5, 0x00, 0x0E, 0x00, 0x0E, 0x00, 0x10, 0x00, 0x03, 0x00, 0xFF, 0xFF, 0xFF, 0xFF , OxFC, OxFF, OxF5, OxFF, OxFA, OxFF, OxFE, OxFF, OxF8, OxFF, OxEF, OxFF, OxEC, 0xFF, 0xEE, 0xFF, 0xED, 0xFF, 0xF3, 0xFF, 0xFF, 0xFF, 0x09, 0x00, 0x10, 0x00, 0 kOE, 0x00, 0x12, 0x00, 0x12, 0x00, 0x06, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xO4, 0x 00, 0x04, 0x00, 0x06, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFF, 0xFC, 0xFF, 0xF 7, 0xFF, 0xF1, 0xFF, 0xEA, 0xFF, 0xF7, 0xFF, 0x0C, 0x00, 0x15, 0x00, 0x0B, 0x00 . 0x01. 0x00. 0xFA. 0xFF. 0xF9. 0xFF. 0xF7. 0xFF. 0xF9. 0xFF. 0xFB. 0xFF. 0xFB. OxFF, OxFB, OxFF, OxFA, OxFF, OxF3, OxFF, OxEB, OxFF, OxEB, OxFF, OxEE, OxFF, O xF1, 0xFF, 0xF2, 0xFF, 0x01, 0x00, 0x08, 0x00, 0x06, 0x00, 0x01, 0x00, 0x0C, 0x 📦 00 0-07 0-00 0-07 0-00 0-07 0-00 0-07 0-00 0-10 0-00 0-25 0-00 0-1

声道: 1 采样率:8000 采样深度:16数据大小:32714

## 4.3 程序编写:

```
//SoundPlay.h
#ifndef SOUNDPLAY H REVISION FIRST
#define SOUNDPLAY H REVISION FIRST
                      ***************
                          12000000 // 徧 Z 12000000HZ
#define SYSTEM OSC
                                    // ' ij j ,//ÿ4
                    4/5
#define SOUND SPACE
    BeepIO = P3^7;
                               //
sbit
unsigned int code FreTab[12] = { 262,277,294,311,330,349,369,392,415,440,466,494 }; //¤'Z h
unsigned char code SignTab[7] = { 0,2,4,5,7,9,11 };
unsigned char code LengthTab[7]= { 1,2,4,8,16,32,64 };
unsigned char Sound_Temp_TH0,Sound_Temp_TL0; //
//****************
                       ***********
void InitialSound(void)
{
      BeepIO = 0;
      Sound_Temp_TH1 = (65535-(1/1200)*SYSTEM_OSC)/256; // TL1Ön ij
                                                                     (10ms ij
      Sound_Temp_TL1 = (65535-(1/1200)*SYSTEM_OSC)%256; // TH1Ön ij
      TH1 = Sound Temp TH1;
      TL1 = Sound_Temp_TL1;
      TMOD = 0 \times 11;
      ET0 = 1;
      ET1
               = 0;
      TR0
               = 0;
      TR1 = 0;
      EA = 1;
}
void BeepTimer0(void) interrupt 1 //
{
      BeepIO = !BeepIO;
      TH0 = Sound_Temp_TH0;
      TL0 = Sound_Temp_TL0;
}
                    ***************
void Play(unsigned char *Sound,unsigned char Signature,unsigned Octachord,unsigned int Speed)
```

```
{
                                                // µ Z <sup>fi</sup>
       unsigned int NewFreTab[12];
       unsigned char i,j;
       unsigned int Point,LDiv,LDiv0,LDiv1,LDiv2,LDiv4,CurrentFre,Temp_T,SoundLength;
       unsigned char Tone, Length, SL, SH, SM, SLen, XG, FD;
       for(i=0;i<12;i++)
                                                       // ن Ż "
                                                                                μZñ
       {
               j = i + Signature;
               if(j > 11)
               {
                       j = j-12;
                       NewFreTab[i] = FreTab[j]*2;
               }
               else
                       NewFreTab[i] = FreTab[j];
               if(Octachord == 1)
                       NewFreTab[i]>>=2;
               else if(Octachord == 3)
                       NewFreTab[i]<<=2;</pre>
       }
       SoundLength = 0;
       while(Sound[SoundLength] != 0x00) //
       {
               SoundLength+=2;
       }
       Point = 0;
       Tone = Sound[Point];
                                                      // h
       Length = Sound[Point+1];
                                                         // 1 ij (
       LDiv0 = 12000/Speed;
                                                                               10ms)
       LDiv4 = LDiv0/4;
                                                             //
                                                                          ij
       LDiv4 = LDiv4-LDiv4*SOUND_SPACE;
                                         // ° � �
       TR0
                   = 0;
       TR1 = 1;
       while(Point < SoundLength)</pre>
       {
```

```
SL=Tone%10;
SM=Tone/10%10;
SH=Tone/100;
if(SL!=0)
{
      if (SM==1) CurrentFre >>= 2;
                                              //
      if (SM==3) CurrentFre <<= 2;</pre>
                                              //
      Temp_T = 65536-(50000/CurrentFre)*10/(12000000/SYSTEM_OSC);//
       Sound_Temp_TH0 = Temp_T/256;
       Sound_Temp_TL0 = Temp_T%256;
       TH0 = Sound_Temp_TH0;
       TL0 = Sound_Temp_TL0 + 12; // 12 H x h IJ
                                   Æ
SLen=LengthTab[Length%10];
                        //
                                   // (0 1 ^{\circ} 2 )
XG=Length/10%10;
FD=Length/100;
LDiv=LDiv0/SLen;
                                   //
                                                 ij ( 10 گى ms)
if (FD==1)
      LDiv=LDiv+LDiv/2;
if(XG!=1)
                                           // c
      if(XG==0)
              if (SLen<=4)</pre>
                     LDiv1=LDiv-LDiv4;
              else
                     LDiv1=LDiv*SOUND_SPACE;
       else
              LDiv1=LDiv/2;
                                        //
                                                    ⟨?⟩
else
       LDiv1=LDiv;
if(SL==0) LDiv1=0;
      LDiv2=LDiv-LDiv1;
                                   // ij
 if (SL!=0)
{
      TR0=1;
      for(i=LDiv1;i>0;i--) // 涨 η。
       {
              while(TF1==0);
              TH1 = Sound_Temp_TH1;
```

//

//

//

```
TL1 = Sound_Temp_TL1;
                              TF1=0;
                       }
               }
               if(LDiv2!=0)
               {
                      TR0=0; BeepIO=0;
                      for(i=LDiv2;i>0;i--) // ]
                       {
                              while(TF1==0);
                              TH1 = Sound_Temp_TH1;
                              TL1 = Sound_Temp_TL1;
                              TF1=0;
                       }
               }
               Point+=2;
               Tone=Sound[Point];
               Length=Sound[Point+1];
       }
       BeepIO = 0;
}
#endif
```

```
PlayMusic.c
#include <REG52.H>
#include "SoundPlay.h"
void Delay1ms(unsigned int count)
{
        unsigned int i,j;
        for(i=0;i<count;i++)</pre>
        for(j=0;j<120;j++);
}
                    *********Music****
//挥着翅膀的女孩
unsigned char code Music_Girl[]={ 0x17,0x02, 0x17,0x03, 0x18,0x03, 0x19,0x02, 0x15,0x03,
                                  0x16,0x03, 0x17,0x03, 0x17,0x03, 0x17,0x03, 0x18,0x03,
                                  0x19,0x02, 0x16,0x03, 0x17,0x03, 0x18,0x02, 0x18,0x03,
                                  0x17,0x03, 0x15,0x02, 0x18,0x03, 0x17,0x03, 0x18,0x02,
                                  0x10,0x03, 0x15,0x03, 0x16,0x02, 0x15,0x03, 0x16,0x03,
                                  0x17,0x02, 0x17,0x03, 0x18,0x03, 0x19,0x02, 0x1A,0x03,
                                  0x1B,0x03, 0x1F,0x03, 0x1F,0x03, 0x17,0x03, 0x18,0x03,
                                  0x19,0x02, 0x16,0x03, 0x17,0x03, 0x18,0x03, 0x17,0x03,
                                  0x18,0x03, 0x1F,0x03, 0x1F,0x02, 0x16,0x03, 0x17,0x03,
                                  0x18,0x03, 0x17,0x03, 0x18,0x03, 0x20,0x03, 0x20,0x02,
                                  0x1F,0x03, 0x1B,0x03, 0x1F,0x66, 0x20,0x03, 0x21,0x03,
                                  0x20,0x03, 0x1F,0x03, 0x1B,0x03, 0x1F,0x66, 0x1F,0x03,
                                  0x1B,0x03, 0x19,0x03, 0x19,0x03, 0x15,0x03, 0x1A,0x66,
                                  0x1A,0x03, 0x19,0x03, 0x15,0x03, 0x15,0x03, 0x17,0x03,
                                  0x16,0x66, 0x17,0x04, 0x18,0x04, 0x18,0x03, 0x19,0x03,
                                  0x1F,0x03, 0x1B,0x03, 0x1F,0x66, 0x20,0x03, 0x21,0x03,
                                  0x20,0x03, 0x1F,0x03, 0x1B,0x03, 0x1F,0x66, 0x1F,0x03,
                                  0x1B,0x03, 0x19,0x03, 0x19,0x03, 0x15,0x03, 0x1A,0x66,
                                  0x1A,0x03, 0x19,0x03, 0x19,0x03, 0x1F,0x03, 0x1B,0x03,
                                  0x1F,0x00, 0x1A,0x03, 0x1A,0x03, 0x1A,0x03, 0x1B,0x03,
                                  0x1B,0x03, 0x1A,0x03, 0x19,0x03, 0x19,0x02, 0x17,0x03,
                                  0x15,0x17, 0x15,0x03, 0x16,0x03, 0x17,0x03, 0x18,0x03,
                                  0x17,0x04, 0x18,0x0E, 0x18,0x03, 0x17,0x04, 0x18,0x0E,
                                  0x18,0x66, 0x17,0x03, 0x18,0x03, 0x17,0x03, 0x18,0x03,
                                  0x20,0x03, 0x20,0x02, 0x1F,0x03, 0x1B,0x03, 0x1F,0x66,
                                  0x20,0x03, 0x21,0x03, 0x20,0x03, 0x1F,0x03, 0x1B,0x03,
```

```
0x1F,0x66, 0x1F,0x04, 0x1B,0x0E, 0x1B,0x03, 0x19,0x03,
                                   0x19,0x03, 0x15,0x03, 0x1A,0x66, 0x1A,0x03, 0x19,0x03,
                                   0x15,0x03, 0x15,0x03, 0x17,0x03, 0x16,0x66, 0x17,0x04,
                                   0x18,0x04, 0x18,0x03, 0x19,0x03, 0x1F,0x03, 0x1B,0x03,
                                   0x1F,0x66, 0x20,0x03, 0x21,0x03, 0x20,0x03, 0x1F,0x03,
                                   0x1B,0x03, 0x1F,0x66, 0x1F,0x03, 0x1B,0x03, 0x19,0x03,
                                   0x19,0x03, 0x15,0x03, 0x1A,0x66, 0x1A,0x03, 0x19,0x03,
                                   0x19,0x03, 0x1F,0x03, 0x1B,0x03, 0x1F,0x00, 0x18,0x02,
                                   0x18,0x03, 0x1A,0x03, 0x19,0x0D, 0x15,0x03, 0x15,0x02,
                                   0x18,0x66, 0x16,0x02, 0x17,0x02, 0x15,0x00, 0x00,0x00};
//同一首歌
unsigned char code Music Same[]={ 0 \times 0F, 0 \times 01, 0 \times 15, 0 \times 02, 0 \times 16, 0 \times 02, 0 \times 17, 0 \times 66, 0 \times 18, 0 \times 03,
                                   0x17,0x02, 0x15,0x02, 0x16,0x01, 0x15,0x02, 0x10,0x02,
                                   0x15,0x00, 0x0F,0x01, 0x15,0x02, 0x16,0x02, 0x17,0x02,
                                   0x17,0x03, 0x18,0x03, 0x19,0x02, 0x15,0x02, 0x18,0x66,
                                   0x17,0x03, 0x19,0x02, 0x16,0x03, 0x17,0x03, 0x16,0x00,
                                   0x17,0x01, 0x19,0x02, 0x1B,0x02, 0x1B,0x70, 0x1A,0x03,
                                   0x1A,0x01, 0x19,0x02, 0x19,0x03, 0x1A,0x03, 0x1B,0x02,
                                   0x1A,0x0D, 0x19,0x03, 0x17,0x00, 0x18,0x66, 0x18,0x03,
                                   0x19,0x02, 0x1A,0x02, 0x19,0x0C, 0x18,0x0D, 0x17,0x03,
                                   0x16,0x01, 0x11,0x02, 0x11,0x03, 0x10,0x03, 0x0F,0x0C,
                                   0x10,0x02, 0x15,0x00, 0x1F,0x01, 0x1A,0x01, 0x18,0x66,
                                   0x19,0x03, 0x1A,0x01, 0x1B,0x02, 0x1B,0x03, 0x1B,0x03,
                                   0x1B,0x0C, 0x1A,0x0D, 0x19,0x03, 0x17,0x00, 0x1F,0x01,
                                   0x1A,0x01, 0x18,0x66, 0x19,0x03, 0x1A,0x01, 0x10,0x02,
                                   0x10,0x03, 0x10,0x03, 0x1A,0x0C, 0x18,0x0D, 0x17,0x03,
                                   0x16,0x00, 0x0F,0x01, 0x15,0x02, 0x16,0x02, 0x17,0x70,
                                   0x18,0x03, 0x17,0x02, 0x15,0x03, 0x15,0x03, 0x16,0x66,
                                   0x16,0x03, 0x16,0x02, 0x16,0x03, 0x15,0x03, 0x10,0x02,
                                   0x10,0x01, 0x11,0x01, 0x11,0x66, 0x10,0x03, 0x0F,0x0C,
                                   0x1A,0x02, 0x19,0x02, 0x16,0x03, 0x16,0x03, 0x18,0x66,
                                   0x18,0x03, 0x18,0x02, 0x17,0x03, 0x16,0x03, 0x19,0x00,
                                   0x00,0x00 };
//两只蝴蝶
unsigned char code Music_Two[] ={ 0x17,0x03, 0x16,0x03, 0x17,0x01, 0x16,0x03, 0x17,0x03,
                                   0x16,0x03, 0x15,0x01, 0x10,0x03, 0x15,0x03, 0x16,0x02,
                                   0x16,0x0D, 0x17,0x03, 0x16,0x03, 0x15,0x03, 0x10,0x03,
                                   0x10,0x0E, 0x15,0x04, 0x0F,0x01, 0x17,0x03, 0x16,0x03,
                                   0x17,0x01, 0x16,0x03, 0x17,0x03, 0x16,0x03, 0x15,0x01,
```

```
0x10,0x03, 0x15,0x03, 0x16,0x02, 0x16,0x0D, 0x17,0x03,
                                  0x16,0x03, 0x15,0x03, 0x10,0x03, 0x15,0x03, 0x16,0x01,
                                  0x17,0x03, 0x16,0x03, 0x17,0x01, 0x16,0x03, 0x17,0x03,
                                  0x16,0x03, 0x15,0x01, 0x10,0x03, 0x15,0x03, 0x16,0x02,
                                  0x16,0x0D, 0x17,0x03, 0x16,0x03, 0x15,0x03, 0x10,0x03,
                                  0x10,0x0E, 0x15,0x04, 0x0F, 0x01, 0x17, 0x03, 0x19, 0x03,
                                  0x19,0x01, 0x19,0x03, 0x1A,0x03, 0x19,0x03, 0x17,0x01,
                                  0x16,0x03, 0x16,0x03, 0x16,0x02, 0x16,0x0D, 0x17,0x03,
                                  0x16,0x03, 0x15,0x03, 0x10,0x03, 0x10,0x0D, 0x15,0x00,
                                  0x19,0x03, 0x19,0x03, 0x1A,0x03, 0x1F,0x03, 0x1B,0x03,
                                  0x1B,0x03, 0x1A,0x03, 0x17,0x0D, 0x16,0x03, 0x16,0x03,
                                  0x16,0x0D, 0x17,0x01, 0x17,0x03, 0x17,0x03, 0x19,0x03,
                                  0x1A,0x02, 0x1A,0x02, 0x10,0x03, 0x17,0x0D, 0x16,0x03,
                                  0x16,0x01, 0x17,0x03, 0x19,0x03, 0x19,0x03, 0x17,0x03,
                                  0x19,0x02, 0x1F,0x02, 0x1B,0x03, 0x1A,0x03, 0x1A,0x0E,
                                  0x1B,0x04, 0x17,0x02, 0x1A,0x03, 0x1A,0x03, 0x1A,0x0E,
                                  0x1B,0x04, 0x1A,0x03, 0x19,0x03, 0x17,0x03, 0x16,0x03,
                                  0x17,0x0D, 0x16,0x03, 0x17,0x03, 0x19,0x01, 0x19,0x03,
                                  0x19,0x03, 0x1A,0x03, 0x1F,0x03, 0x1B,0x03, 0x1B,0x03,
                                  0x1A,0x03, 0x17,0x0D, 0x16,0x03, 0x16,0x03, 0x16,0x03,
                                  0x17,0x01, 0x17,0x03, 0x17,0x03, 0x19,0x03, 0x1A,0x02,
                                  0x1A,0x02, 0x10,0x03, 0x17,0x0D, 0x16,0x03, 0x16,0x01,
                                  0x17,0x03, 0x19,0x03, 0x19,0x03, 0x17,0x03, 0x19,0x03,
                                  0x1F,0x02, 0x1B,0x03, 0x1A,0x03, 0x1A,0x0E, 0x1B,0x04,
                                  0x17,0x02, 0x1A,0x03, 0x1A,0x03, 0x1A,0x0E, 0x1B,0x04,
                                  0x17,0x16, 0x1A,0x03, 0x1A,0x03, 0x1A,0x0E, 0x1B,0x04,
                                  0x1A,0x03, 0x19,0x03, 0x17,0x03, 0x16,0x03, 0x0F,0x02,
                                  0x10,0x03, 0x15,0x00, 0x00,0x00 };
main()
{
        InitialSound();
        while(1)
        {
                Play(Music_Girl, 0, 3, 360);
                Delay1ms(500);
                Play(Music_Same, 0, 3, 360);
                Delay1ms(500);
                Play(Music_Two, 0, 3, 360);
```

```
Delay1ms(500);
}
```