

音乐的四大基本要素

一般来说，所有音乐的构成有四个基本要素，而其中最重要的是“音的高低”和“音的长短”，决定了曲子有别于其他曲子：

1 音的高低：任何一首曲子都是高低相间的音调组成的，从钢琴上直观看就是从左往右声音越高。

2 音的长短：指一个音调持续的时间长度。

3 音的力度：音乐的力度很容易理解，也叫强度。一首音乐作品总会有一些音符的力度比较强一些，有些地方弱一些。而力度的变化是音乐作品中表达情感的因素之一。

4 音质：也可以称音色。也就是发出音乐的乐器或人声。

简谱中常用概念及符号

1 音符：记录音的高低和长短的符号，常用 1 2 3 4 5 6 7 表示

休止符：表示无声音，用 0 表示，也有音长

2 音符的音高：阿拉伯数字头上点高音，脚下点低音，没点则是中音，且阿拉伯数字越大，音符音高越大

EG: 3 中音

 5_· 低音

_·3 高音

3 音符长短：长短定义是在相对时间上定义的，因此引入拍子概念作为参照，四分音符为一拍，手掌一上一下是一拍。设节拍时间为 T。

EG: 全音符 5--- 8T

二分音符	5-	2T
四分音符	5	T
八分音符	<u>5</u>	T/2
十六分音符	<u><u>5</u></u>	T/4
三十二分音符	5(3 下划线)	T/8

4 半音与全音：相邻的两个音之间最小的距离叫半音，两个半音距离构成一个全音

5 变化音：将音符升高半音，叫升音，用“#”标在音符左上；将音符降低半音用符号“b”标在音符左上。

6 附点音符：附点就是记在音符右边的小圆点，表示增加前面音符时值的一半，带附点的音符叫做附点音符。

EG： 假设节拍时间 T，则 $6\cdot$ 时值为 $T+T/2$

7 小节线：用竖线将每一小节划分开线叫小节线。

定时器初值计算方法：

设音符频率为 $X=440\text{Hz}$ （标准音），晶振为 11.0592MHz

(1) 计算机器周期

$$\text{机器周期} = (1/11.0592\text{M}) * 12 = 1.085\mu\text{s}$$

(2) 计算音符频率周期，音符频率对应半周期

$$\text{音符频率周期} = 1/(2X) = 1/(2*440) = 1136.363636\mu\text{s}$$

(3) 计算计时周期数 = 音符频率周期/机器周期

$$= 1136.363636\mu\text{s} / 1.085\mu\text{s} = 1047.34$$

(4) 计算定时器初值 = 65536 - [计时周期数] = 64489d=FBE9H

/******

* 定义音符表 晶振:11.0592MHz

* 作者：夏侯佐鑫 日期：2012/1/14

*****/

//音频变量	定时器值	音频率	音名
#define LOW_1	0xF91F	//261.626Hz	c1
#define LOW_1s	0xF982	//277.183Hz	c1#
#define LOW_2	0xF9DF	//293.665Hz	d1
#define LOW_2s	0xFA37	//311.127Hz	d1#
#define LOW_3	0xFA8A	//329.628Hz	e1
#define LOW_4	0xFAD9	//349.228Hz	f1
#define LOW_4s	0xFB23	//369.994hz	f1#
#define LOW_5	0xFB68	//391.995Hz	g1
#define LOW_5s	0xFBAA	//415.305Hz	g1#
#define LOW_6	0xFBE9	//440.000Hz	a1(标准音)
#define LOW_6s	0xFC24	//466.164hz	a1#
#define LOW_7	0xFC5B	//493.883Hz	b1
#define MID_1	0xFC8F	//523.251Hz	c2
#define MID_1s	0xFCC1	//554.365Hz	c2#
#define MID_2	0xFCEF	//587.330Hz	d1

```
#define MID_2s    0xFD1B    //622.254Hz  d2#
#define MID_3     0xFD45    //659.255Hz  e2
#define MID_4     0xFD6C    //698.456Hz  f2
#define MID_4s    0xFD91    //739.989Hz  f2#
#define MID_5     0xFDB4    //783.991Hz  g2
#define MID_5s    0xFDD5    //830.609Hz  g2#
#define MID_6     0xFDF4    //880.000Hz  a2
#define MID_6s    0xFE12    //932.328Hz  a2#
#define MID_7     0xFE2D    //987.767HZ  b2

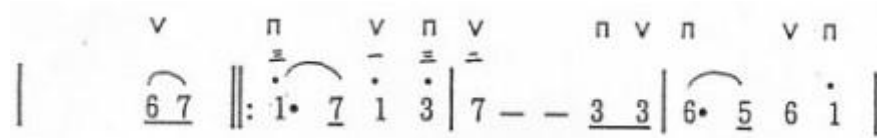
#define HIGH_1    0xFE48    //1046.500Hz  c3
#define HIGH_1s   0xFE60    //1108.730Hz  c3#
#define HIGH_2    0xFE78    //1174.660Hz  d3
#define HIGH_2s   0xFE8E    //1244.510Hz  d3#
#define HIGH_3    0xFEA3    //1318.510Hz  e3
#define HIGH_4    0xFEB6    //1396.910HZ  f3
#define HIGH_4s   0xFEC9    //1479.980Hz  f3#
#define HIGH_5    0xFEDA    //1567.980Hz  g3
#define HIGH_5s   0xFEED    //1661.220Hz  g3#
#define HIGH_6    0xFEFA    //1760.000Hz  a3
#define HIGH_6s   0xFF09    //1864.660Hz  a3#
#define HIGH_7    0xFF17    //1975.530Hz  b3
```

音乐结构体定义及音乐播放

(1) 定义结构体

```
typedef struct
{
    UINT16 frq; //音频
    UINT16 tim; //时常
} music_struct;
```

(2) 音乐编码数组，以《天空之城》的前几个小结为例



```
code music_struct tkzc[] =
{
    {MID_6, T/2},
    {MID_7, T/2},
    {HIGH_1, T+T/2},
    {MID_7, T/2},
    {HIGH_1, T},
    {HIGH_3, T},
    {MID_7, 4*T},
    {MID_3, T/2},
    {MID_3, T/2},
    {MID_6, T+T/2},
```

```
{MID_5, T/2},  
  
{MID_6, T},  
  
{HIGH_1, T},  
  
{MID_5, 2*T},  
  
{0, T},  
  
{MID_3, T/2},  
  
{MID_3, T/2},  
  
{MID_4, T+T/2},  
  
{MID_3, T/2},  
  
{MID_4, T/2},  
  
{HIGH_1, T+T/2},  
  
.....  
  
.....  
  
{0, 0} //结束符，判断音乐结束  
  
};
```

- (3) 值得注意的是，播放音乐最好使用两个定时器，一个用于音长的定时，一个用于音频确定