使用Arduino演奏Flower Dance

杨远帆 物理学系

摘要 本文中，作者改进了 Arduino 的 tone 函数，使蜂鸣器发声效果改善并演奏。除此之外，作者尝试去改变声音的响度及音色。

一、 引言

Arduino 自身拥有的 tone 函数使蜂鸣器发声的效果并不理想，作者编写了新的 mytone 函数并将其用于演奏 Flower Dance。演奏时液晶屏会将曲谱以图形的方式大致显示出来。作者另外尝试改变蜂鸣器发声的响度合音色，不过并未与演奏结合。

二、 实验装置

ArduinoUNO R3 开发板，UNO9 合 1 扩展板中的蜂鸣器，LCD1602 液晶屏

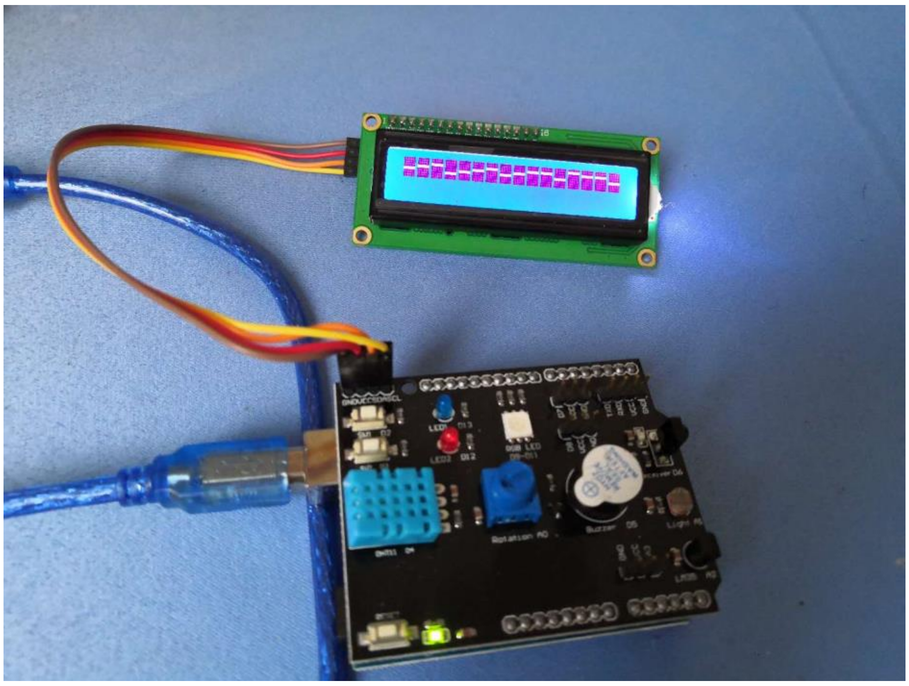

图 1 实验装置

三、 实验内容

1、改进 tone 函数

由 arduino 官网[1]可知，tone 函数的作用为产生 50%占空比、指定频率的方

波。在实际使用时，使用 tone 函数让蜂鸣器发声所得不同频率的声音区别极小，显然未能完成其功能。按照 tone 函数所说明的功能重写发声函数也得到相同的结果。

改进 tone 函数得 mytone 函数，见附录 A。首先生成周期为 2*delayus 的方波，该周期通常为数十至数百微秒。然后以该方波为基础生成所需频率的声音。在所需声音的一个周期内重复方波，去除最后一个不完整的方波。最后在持续时间内重复所需声音的周期，得到所需时长、频率的声音。

使用 mytone 函数发声，所得频率无太大偏差，各频率声音区分程度良好，明显优于使用 tone 函数发声。

2、改变声音的响度

示例程序见附录 B。改变 A 的值，即改变占空比，此时响度会发生变化。响度与占空比正相关。但是改变占空比时频率有时会发生变化，推测与实验器材有关，暂未将其与 mytone 整合。

3、改变声音的音色

尝试得知，改变 mytone 函数中的 delayus 或是将作为基础的方波改为三角波等方式均可一定程度上改变音色。同时，由于实际所得的声音为两个频率的叠加，所以从听觉上所得声音的频率有些小的改变。

4、演奏

程序见附录 C。

由十二平均律计算得所需声音的频率并得到周期。由于曲谱较长，为避免动态内存不足，需要将曲谱记录到项目存储空间中。而为了达到上述目的且不影响演奏的流畅性，使用序号标记音符并使用 case 语句将序号转换为频率，一拍用 12 表示。

演奏的同时，在液晶屏上以图形的方式显示曲谱。由于分辨率不足，将每四个音符用同一个符号表示，从而大致显示。因为所演奏的曲目中音符跳跃较大，上述大致显示的做法并无不妥。为避免屏幕内容滚动时上下交错，仅使用液晶屏

第一行。

四、 实验结果

1、先生成频率较大的波形，再以此为基础生成频率较小的波形可有效改善发声状况，并且前者能够影响音色。这种方法发声效果更好的原因推测与实验器材或人的听觉系统有关。
2、使用 PWM 可控制蜂鸣器发出声音的响度，但频率会改变。频率改变的原因推测是实验器材的限制。

五、 参考资料

[1] https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/

附录

A mytone 函数

```
void mytone(int T,int duration)//duration:ms  T:us
{
  if(T==0)delay(duration);
  else
  {
    int i,j;
    for(i=0;i<floor(1000*(float)duration/T);i++)
    {
      for(j=1;j<floor(T/2/delayus);j++)
      {
        digitalWrite(Buzzer,HIGH);
        delayMicroseconds(delayus);
        digitalWrite(Buzzer,LOW);
        delayMicroseconds(delayus);
      }
      delayMicroseconds(T-2*delayus*(j-1));
    }
    delayMicroseconds(1000*duration-i*T);
  }
}
```

B 调节响度程序示例

```
#define A 100
void setup() {
  pinMode(5,OUTPUT);
}
```

```
void loop() {
  for(int t=0;t<2000;t+=2)
{
  digitalWrite(5,HIGH);
  delayMicroseconds(A);
  digitalWrite(5,LOW);
  delayMicroseconds(2000-A);
}
}
```

C 《Flower Dance》代码

```
#include <avr/pgmspace.h>
#include<Wire.h>
#include<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,1);

#define B3  1

#define C4  2
#define C4U 3
#define D4  4
#define D4U 5
#define E4  6
#define F4  7
#define F4U 8
#define G4  9
#define G4U 10
#define A4  11
#define A4U 12
#define B4  13

#define C5  14
#define C5U 15
#define D5  16
#define D5U 17
#define E5  18
#define F5  19
#define F5U 20
#define G5  21
#define G5U 22
#define A5  23
#define A5U 24
#define B5  25
```

```
#define C6  26
#define C6U 27
#define D6  28
#define D6U 29
#define E6  30
#define F6  31
#define F6U 32
#define G6  33                                    //使用序号代替周期，方便存储
在项目存储空间中，使用 case 换为周期

#define delayus 80
#define Buzzer 5                                 //蜂鸣器
#define spd 96                                   //speed

void mytone(int T,int duration)//duration:ms  T:us
{
  if(T==0)delay(duration);
  else
  {
    int i,j;
    for(i=0;i<floor(1000*(float)duration/T);i++)
    {
      for(j=1;j<floor(T/2/delayus);j++)
      {
        digitalWrite(Buzzer,HIGH);
        delayMicroseconds(delayus);
        digitalWrite(Buzzer,LOW);
        delayMicroseconds(delayus);
      }
      delayMicroseconds(T-2*delayus*(j-1));
    }
    delayMicroseconds(1000*duration-i*T);
  }
}

const unsigned char tune[]PROGMEM=                //谱子
{                                                     //升 CDFGA  C→CU D→DU
F→FU G→GU A→AU
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  0,
```

D5U, G4U, B4, D5U, C5U, F5U,
C5U, A5, G5U, F5U, G5U, D6U,
G5U, D6U, C6U, C6U, D6U, C6U, A5U, F5U,
G5U, 0, D5U,
D5U, G4U, B4, D5U, C5U, F5U,
C5U, F5U, G5U, A5U, D5U, B5, C5U, A5U, B4, G5U, A4U, G5,
G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U,
G5U, D6U, C6, D6U, G5U, D6U, C6, D6U, G5U, D6U, C6, D6U, G5U, D5U,
B5, D5U, A5U, D5U, B5, D5U, C6U, D5U, A5U, A4U, G5U, A4U, F5U, D5U, F5U,
E5, G4U, D5U, G4U, C5U, G4U, E5, G4U, D5U, D4U, C5U, D4U, B4, D4U, D5U, D4U,
C5U, E4, B4, E4, A4U, E4, G4U, E4, G4, G4U, A4U,
B4, D4U, A4U, D4U, B4, D4U, C5U, D4U, A4U, D4U, G4U, D4U, F4U, D4U, F4U,
G4U, B3, F4U, B3, G4U, B3, B4, B3, F4U, B3, E4, B3, D4U, D4U, F4U,
E4, E5, D5U, C5U, B4, A4U, D5U, C5U, D5U, E5, D5U, C5U, B4, A4U,
G4U, D5U, F4U, D5U, G4U,

D5U, G4U, B4, D5U, C5U, F5U,
C5U, A5, G5U, F5U, G5U, D6U,
G5U, D6U, C6U, C6U, D6U, C6U, A5U, F5U,
G5U, 0, D5U,
D5U, G4U, B4, D5U, C5U, F5U,
C5U, F5U, G5U, A5U, D5U, B5, C5U, A5U, B4, G5U, A4U, G5,
G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U,
G5U, D6U, C6, D6U, G5U, D6U, C6, D6U, G5U, D6U, C6, D6U, G5U,

D5U, G4U, B4, D5U, C5U, F5U,
C5U, A5, G5U, F5U, G5U, D6U,
G5U, D6U, C6U, C6U, D6U, C6U, A5U, F5U,
G5U, 0, D5U,
D5U, G4U, B4, D5U, C5U, F5U,
C5U, F5U, G5U, A5U, D5U, B5, C5U, A5U, B4, G5U, A4U, G5,
G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U, G5U, D6U, C6U, D6U,
G5U, D6U, C6, D6U, G5U, D6U, C6, D6U, G5U, D6U, C6, D6U, G5U, D5U,
B5, D5U, A5U, D5U, B5, D5U, C6U, D5U, A5U, A4U, G5U, A4U, F5U, D5U, F5U,
E5, G4U, D5U, G4U, C5U, G4U, E5, G4U, D5U, D4U, C5U, D4U, B4, D4U, D5U, D4U,
C5U, E4, B4, E4, A4U, E4, G4U, E4, G4, G4U, A4U,
B4, D4U, A4U, D4U, B4, D4U, C5U, D4U, A4U, D4U, G4U, D4U, F4U, D4U, F4U,
G4U, B3, F4U, B3, G4U, B3, B4, B3, F4U, B3, E4, B3, D4U, D4U, F4U,
E4, E5, D5U, C5U, B4, A4U, D5U, C5U, D5U, E5, D5U, C5U, B4, A4U,
G4U, D5U, F4U, D5U, G4U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,

G5U, B4, B5, B4, C6U, C5U, D6U, D5U, C6U, C5U, B5, G5U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, 0, G5U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, C6U, C5U, D6U, D5U, C6U, C5U, B5, G5U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, 0, G5U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, C6U, C5U, D6U, D5U, C6U, C5U, B5, G5U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, D5U, G5U, B5, D5U, G5U, B5, D6, G6,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, C6U, C5U, D6U, D5U, D6U, D5U, D6U, D5U, D6U, D5U, F6U, F5U,

G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, B5, B4, A5U, B4, F5U, B4, G5U, B4, F5U, B4, D5U, B4, F5U, B4,
G5U, G4U, B4, D5U, G5U, G4U, B4, D5U, A5U, C5U, D5U, G5U, A5U, C5U, B5, A5U,
G5U, B4, D5U, G5U, B5, D5U, G5U, B5, D6, G6,

0, C5U, D5U, B5, A5U, G5U, A5U, F5U,
B5, D5U, A5U, D5U, B5, D5U, C6U, D5U, A5U, A4U, G5U, A4U, F5U, E5, F5U,
G5U, B4, F5U, B4, G5U, B4, B5, B4, F5U, B4, E5, B4, D5U, D5U, F5U,
E5, G4U, D5U, G4U, C5U, G4U, E5, G4U, D5U, D4U, C5U, D4U, B4, D4U, D5U, D4U,
C5U, E4, B4, E4, A4U, E4, G4U, E4, G4, G4U, A4U,

B4, D4U, A4U, D4U, B4, D4U, C5U, D4U, A4U, D4U, G4U, D4U, F4U, D4U, F4U,

```
  G4U,B3,F4U,B3,G4U,A4U,B4,B3,F4U,B3,E4,B3,D4U,D4U,F4U,
  E4,E5,D5U,C5U,B4,A4U,D5U,C5U,D5U,E5,D5U,C5U,B4,A4U,
  G4U,D5U,F4U,D5U,G4U,D4U,

  B4,D4U,A4U,D4U,B4,D4U,C5U,D4U,A4U,D4U,G4U,D4U,F4U,D4U,F4U,
  G4U,B3,F4U,B3,G4U,B3,B4,B3,F4U,B3,E4,B3,D4U,D4U,F4U,
  E4,E5,D5U,C5U,B4,A4U,D4U,D5U,C5U,B4,A4U,G4U,
  E4,E6,D6U,C6U,B5,A5U,G5U,F5U,E5,D5U,C5U,B4,A4U,G4U,F4U,
  B4,D4U,A4U,D4U,B4,D4U,C5U,D4U,A4U,D4U,G4U,D4U,F4U,D4U,F4U,
  G4U,B3,G4U,F4U,G4U,A4U,B4,B3,F4U,B3,E4,B3,D4U,D4U,F4U,
  E4,E5,D5U,C5U,B4,A4U,D5U,C5U,D5U,E5,D5U,C5U,B4,A4U,
  G4U,D5U,F4U,D5U,G4U,0,

  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C6U,D6U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C6U,D6U,G5U,C5U,D5U,C5U,G4U,C5U,
  D5U,C5U,G5U,C5U,D5U,C5U,G4U,C5U
};

const unsigned char durt[] PROGMEM=     //节拍，将一拍换为12
{
  6,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,6,
  48,
  15,3,3,3,12,12,
  9,3,9,3,12,12,
  12,12,6,2,2,2,6,6,
  30,12,6,
  15,3,3,3,12,12,
  6,6,6,6,3,3,3,3,3,3,3,3,
  3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,
  3,3,3,3,3,3,3,3,3,3,3,3,6,6,
  3,3,3,3,3,3,3,3,3,3,3,3,6,3,3,
  3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,
  3,3,3,3,3,3,3,3,6,6,12,
  3,3,3,3,3,3,3,3,3,3,3,3,6,3,3,
  3,3,3,3,3,3,3,3,3,3,3,3,6,3,3,
  6,3,3,3,3,6,3,3,3,3,3,3,3,3,
```

6, 6, 6, 6, 24,

15, 3, 3, 3, 12, 12,
9, 3, 9, 3, 12, 12,
12, 12, 6, 2, 2, 2, 6, 6,
30, 12, 6,
15, 3, 3, 3, 12, 12,
6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 12,

15, 3, 3, 3, 12, 12,
9, 3, 9, 3, 12, 12,
12, 12, 6, 2, 2, 2, 6, 6,
30, 12, 6,
15, 3, 3, 3, 12, 12,
6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 6,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 6, 6, 12,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
6, 3, 3, 3, 3, 6, 3, 3, 3, 3, 3, 3, 3, 3,
6, 6, 6, 6, 24,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 12, 6,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
30, 12, 6,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 12, 6,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
30, 12, 6,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 12, 6,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
12, 3, 3, 3, 3, 3, 3, 3, 3, 12,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 12, 36,

3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 6, 6, 12,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
6, 3, 3, 3, 3, 6, 3, 3, 3, 3, 3, 3, 3, 3,
6, 6, 6, 6, 18, 6,

3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
6, 3, 3, 3, 3, 6, 6, 3, 3, 3, 3, 6,
6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 6, 3, 3,
6, 3, 3, 3, 3, 6, 3, 3, 3, 3, 3, 3, 3, 3,
6, 6, 6, 6, 18, 6,

6, 6, 6, 6, 6, 6, 6, 6,

```
  6,6,6,6,6,6,6,6,
  3,3,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,6,
  3,3,6,6,6,6,6,6,6,
  6,6,6,6,6,6,6,60
};

uint8_t a[8]={0x1f,0x0,0x0,0x0,0x0,0x0,0x0,0x0};         //用来在液晶屏上大致显示
谱子
uint8_t b[8]={0x0,0x1f,0x0,0x0,0x0,0x0,0x0,0x0};
uint8_t c[8]={0x0,0x0,0x1f,0x0,0x0,0x0,0x0,0x0};
uint8_t d[8]={0x0,0x0,0x0,0x1f,0x0,0x0,0x0,0x0};
uint8_t e[8]={0x0,0x0,0x0,0x0,0x1f,0x0,0x0,0x0};
uint8_t f[8]={0x0,0x0,0x0,0x0,0x0,0x1f,0x0,0x0};
uint8_t g[8]={0x0,0x0,0x0,0x0,0x0,0x0,0x1f,0x0};
uint8_t h[8]={0x0,0x0,0x0,0x0,0x0,0x0,0x0,0x1f};

int len;

void setup() {
  pinMode(Buzzer,OUTPUT);
  len=sizeof(tune)/sizeof(tune[0]);
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);

  lcd.createChar(0,a);
  lcd.createChar(1,b);
  lcd.createChar(2,c);
  lcd.createChar(3,d);
  lcd.createChar(4,e);
  lcd.createChar(5,f);
  lcd.createChar(6,g);
  lcd.createChar(7,h);
  lcd.home();

  lcd.print("Flower Dance");
  delay(1000);

  lcd.autoscroll();
  lcd.setCursor(16,0);
}
```

```
//for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.print(" ");
void loop() {
  int n;
  for(int x=0;x<len;x++)
  {
    switch(pgm_read_byte(&tune[x]))
    {
      case       0:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.printstr("
");delay(62.5*pgm_read_byte(&durt[x]));break;
      case
1:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(7);mytone(2408,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
2:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(7);mytone(2273,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
3:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(7);mytone(2145,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
4:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(7);mytone(2024,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
5:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(6);mytone(1911,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
6:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(6);mytone(1803,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
7:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(6);mytone(2703,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
8:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(6);mytone(1607,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
9:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(5);mytone(1517,5000/s
pd*pgm_read_byte(&durt[x]));break;
      case
10:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(5);mytone(1432,5000/
spd*pgm_read_byte(&durt[x]));break;
      case
11:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(5);mytone(1351,5000/
spd*pgm_read_byte(&durt[x]));break;
      case
12:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(5);mytone(1275,5000/
spd*pgm_read_byte(&durt[x]));break;
       case
13:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(4);mytone(1204,5000/
spd*pgm_read_byte(&durt[x]));break;
       case
14:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(4);mytone(1136,5000/
spd*pgm_read_byte(&durt[x]));break;
       case
15:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(4);mytone(1073,5000/
spd*pgm_read_byte(&durt[x]));break;
       case
16:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(4);mytone(1012,5000/
spd*pgm_read_byte(&durt[x]));break;
       case
17:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(3);mytone(956,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
18:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(3);mytone(902,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
19:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(3);mytone(851,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
20:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(3);mytone(804,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
21:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(2);mytone(758,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
22:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(2);mytone(716,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
23:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(2);mytone(676,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
24:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(2);mytone(638,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
25:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(1);mytone(602,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
26:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(1);mytone(568,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
```

```
27:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(1);mytone(536,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
28:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(1);mytone(506,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
29:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(0);mytone(477,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
30:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(0);mytone(451,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
31:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(0);mytone(426,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
32:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(0);mytone(402,5000/s
pd*pgm_read_byte(&durt[x]));break;
       case
33:for(n=0;n<ceil(pgm_read_byte(&durt[x])/3);n++)lcd.write(0);mytone(379,5000/s
pd*pgm_read_byte(&durt[x]));break;
    }
  }
  while(1);
}
```