Arduino学习笔记一：用arduino实现蜂鸣器播放音乐

首先感谢arduino中文社区的tahoroom，开始做这个的时候在他的这个项目里我学了很多东西:<https://www.arduino.cn/thread-7404-1-1.html>

1. Arduino的Tone函数

关于arduino播放音乐的原理，可以看上面链接的tahoroom的文章，他讲得相当详细。

Arduino官方提供的Tone函数用于驱动蜂鸣器或扬声器发出我们需要的声音，通常的应用如下：

tone(pin,frequency,duration);//控制pin脚上的音频输出

//pin：arduino的控制脚位；

//frequency:输出的声音频率，换到音乐上可理解为旋律音高；

//duration:旋律持续时间，单位ms；

noTone(pin);//停止播放；

详细的tone函数定义也可以参考官方介绍：

<http://arduino.cc/en/Tutorial/Tone>

<http://arduino.cc/en/Reference/Tone>

1. pitches.h的添加

为了避免在主程序里添加太多的定义，我们把音高的定义独立出来放在一个头文件pitches.h里，然后在主程序里引用即可，下次需要同样的功能时，也可以直接拷贝调用该头文件。

代码如下：

//pitches.h 音高定义

//pitches.h 定义了不同音符对应的不同频率

#define NOTE\_B0 31 //B0 ····7

#define NOTE\_C1 33 //C1 ···1

#define NOTE\_CS1 35 //C1#

#define NOTE\_D1 37 //D1 ···2

#define NOTE\_DS1 39 //D1#

#define NOTE\_E1 41 //E1 ···3

#define NOTE\_F1 44 //F1 ···4

#define NOTE\_FS1 46 //F1#

#define NOTE\_G1 49 //G1 ···5

#define NOTE\_GS1 52 //G1#

#define NOTE\_A1 55 //A1 ···6

#define NOTE\_AS1 58 //A1#

#define NOTE\_B1 62 //B1 ···7

#define NOTE\_C2 65 //C2 ··1

#define NOTE\_CS2 69 //C2#

#define NOTE\_D2 73 //D2 ··2

#define NOTE\_DS2 78 //D2#

#define NOTE\_E2 82 //E2 ··3

#define NOTE\_F2 87 //F2 ··4

#define NOTE\_FS2 93 //F2#

#define NOTE\_G2 98 //G2 ··5

#define NOTE\_GS2 104 //G2#

#define NOTE\_A2 110 //A2 ··6

#define NOTE\_AS2 117 //A2#

#define NOTE\_B2 123 //B2 ··7

#define NOTE\_C3 131 //C3 ·1

#define NOTE\_CS3 139 //C3#

#define NOTE\_D3 147 //D3 ·2

#define NOTE\_DS3 156 //D3#

#define NOTE\_E3 165 //E3 ·3

#define NOTE\_F3 175 //F3 ·4

#define NOTE\_FS3 185 //F3#

#define NOTE\_G3 196 //G3 ·5

#define NOTE\_GS3 208 //G3#

#define NOTE\_A3 220 //A3 ·6

#define NOTE\_AS3 233 //A3#

#define NOTE\_B3 247 //B3 ·7

#define NOTE\_C4 262 //C4 1（中央C大调）

#define NOTE\_CS4 277 //C4#

#define NOTE\_D4 294 //D4 2

#define NOTE\_DS4 311 //D4#

#define NOTE\_E4 330 //E4 3

#define NOTE\_F4 349 //F4 4

#define NOTE\_FS4 370 //F4#

#define NOTE\_G4 392 //G4 5

#define NOTE\_GS4 415 //G4#

#define NOTE\_A4 440 //A4 6

#define NOTE\_AS4 466 //A4#

#define NOTE\_B4 494 //B4 7

#define NOTE\_C5 523 //C5 1·

#define NOTE\_CS5 554 //C5#

#define NOTE\_D5 587 //D5 2·

#define NOTE\_DS5 622 //D5#

#define NOTE\_E5 659 //E5 3·

#define NOTE\_F5 698 //F5 4·

#define NOTE\_FS5 740 //F5#

#define NOTE\_G5 784 //G5 5·

#define NOTE\_GS5 831 //G5#

#define NOTE\_A5 880 //A5 6·

#define NOTE\_AS5 932 //A5#

#define NOTE\_B5 988 //B5 7·

#define NOTE\_C6 1047 //C6 1··

#define NOTE\_CS6 1109 //C6#

#define NOTE\_D6 1175 //D6 2··

#define NOTE\_DS6 1245 //D6#

#define NOTE\_E6 1319 //E6 3··

#define NOTE\_F6 1397 //F6 4··

#define NOTE\_FS6 1480 //F6#

#define NOTE\_G6 1568 //G6 5··

#define NOTE\_GS6 1661 //G6#

#define NOTE\_A6 1760 //A6 6··

#define NOTE\_AS6 1865 //A6#

#define NOTE\_B6 1976 //B6 7··

#define NOTE\_C7 2093 //C7 1···

#define NOTE\_CS7 2217 //C7#

#define NOTE\_D7 2349 //D7 2···

#define NOTE\_DS7 2489 //D7#

#define NOTE\_E7 2637 //E7 3···

#define NOTE\_F7 2794 //F7 4···

#define NOTE\_FS7 2960 //F7#

#define NOTE\_G7 3136 //G7 5···

#define NOTE\_GS7 3322 //G7#

#define NOTE\_A7 3520 //A7 6···

#define NOTE\_AS7 3729 //A7#

#define NOTE\_B7 3951 //B7 7···

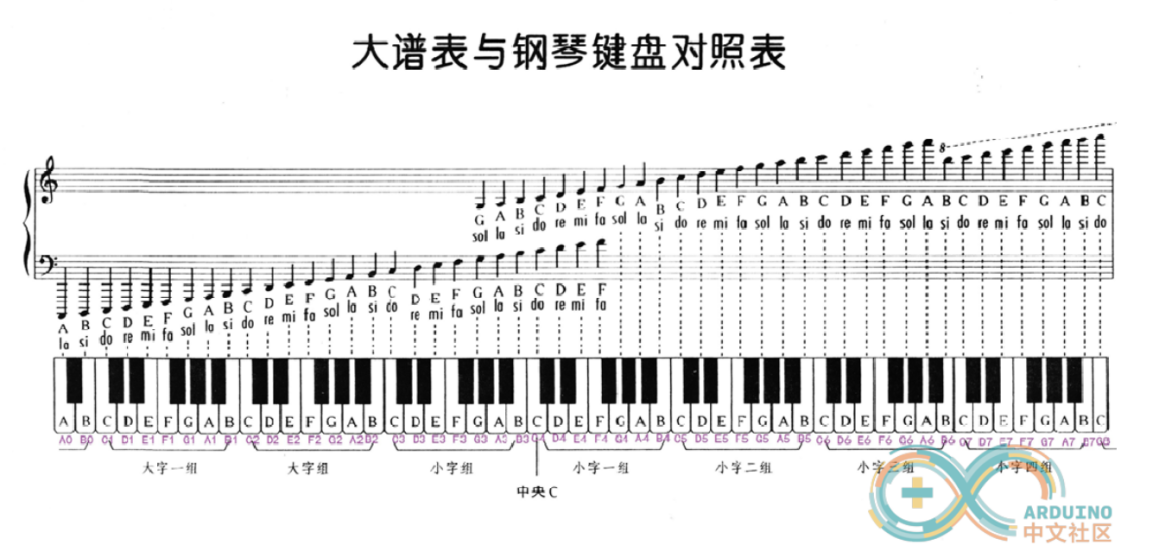
#define NOTE\_C8 4186 //C8 1····

#define NOTE\_CS8 4435 //C8#

#define NOTE\_D8 4699 //D8 2····

#define NOTE\_DS8 4978 //D8#

上面大量的宏定义，是用宏的名字代替了频率名，对应到下图的键盘的按键上。同时，在上面宏的注释里，我加了相对应的简谱，方面不懂音乐的我们便于编程应用。



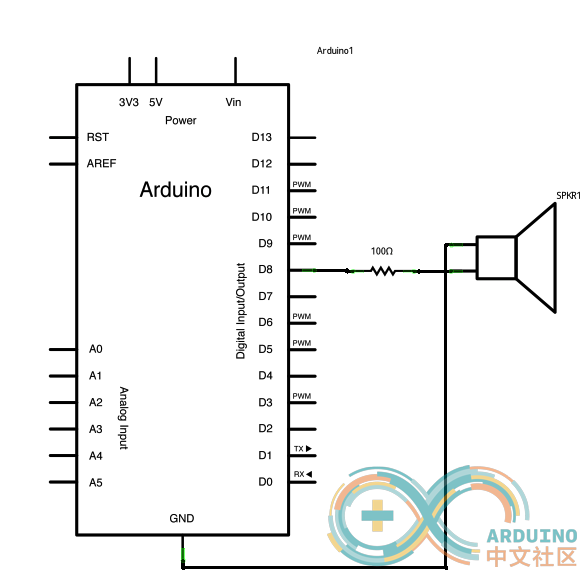
关于创建Arduino的头文件，在Arduino的IDE界面代码框的右上角的向下小三角处点击创建新标签，然后输入自己的头文件名称，记得加上“.h”扩展名。然后输入内容，再保存，便创建好一个头文件了。

1. arduino扬声器接线

**注意：**因为扬声器（或蜂鸣器）驱动电压要求比较小，远远小于5V，所以最好在扬声器上串联一个大于10uf的电解电容，电解电容负极连接喇叭的正极，电容正极连接另外一端。不接电容长时间使用，会烧毁扬声器。或者串联一个100Ω电阻也可以。

可以再串联一个电位器来调节电流大小，从而控制音量大小。

大概接线如下图所示：



1. 音乐编程

在arduino程序里给音乐编程，重要的旋律函数和持续时间函数。

下面以《两只老虎》为例，编写一个播放《两只老虎》旋律的程序。

首先是《两只老虎》的简谱：



代码如下：

#include “pitches.h” //含有所有旋律音高宏定义的头文件

void setup(){

}

void loop(){

play();//播放音乐的命令

delay(300);

}

//旋律函数

int melody[]={

NOTE\_C4, //1

NOTE\_D4, //2

NOTE\_E4, //3

NOTE\_C4, //1

NOTE\_C4, //1

NOTE\_D4, //2

NOTE\_E4, //3

NOTE\_C4, //1

NOTE\_E4, //3

NOTE\_F4, //4

NOTE\_G4, //5

NOTE\_E4, //3

NOTE\_F4, //4

NOTE\_G4, //5

NOTE\_G4, //5

NOTE\_A4, //6

NOTE\_G4, //5

NOTE\_F4, //4

NOTE\_E4, //3

NOTE\_C4, //1

NOTE\_G4, //5

NOTE\_A4, //6

NOTE\_G4, //5

NOTE\_F4, //4

NOTE\_E4, //3

NOTE\_C4, //1

NOTE\_D4, //2

NOTE\_G3, //.5

NOTE\_C4, //1

0,

NOTE\_D4, //2

NOTE\_G3, //.5

NOTE\_C4, //1

0,

};

//持续时间函数

int noteDurations[]={

8,8,8,8,

8,8,8,8,

8,8,4,

8,8,4,

8,8,8,8,4,4,

8,8,8,8,4,4,

4,4,4,4,

4,4,4,4,

};

//播放命令

void play()

{

for(int thisNote = 0;thisNote<29;thisNote++){

//计算旋律持续时间

//1秒（1000ms）除以音符类型

//比如：四分音符：1000/noteDurations[thisNote],

//八分音符：1000/noteDurations[thisNote].

int noteDuration = 1000/noteDurations[thisNote];

tone(8,melody[thisNote], noteDuration);//播放音乐

//为了区别不同旋律，设置一个最小的间隔时间

//一般设置为旋律持续时间的130%比较合适

int pauseBetweenNotes = noteDuration \* 1.30;

delay(pauseBetweenNotes);

//停止当前旋律播放

noTone(8);

}  
}

//end