



Study Notes of Iteaduino Part III

-To Perform Simple Songs with Buzzer

In the last note, we used digital pins on Iteaduino to control ON/OFF of LED lamp and PWM to adjust brightness of LED lamp, etc. In this chapter, we will use the digital pins on Iteaduino to drive buzzer so as to perform simple songs.

Buzzer is one type of electronic beeper of integrated structure with DC voltage supply, widely used in computers, printers, photocopiers, alarm, electronic toys, automotive electronic equipments, telephones, timers and other electronic products as a sounding device. There are two types: piezoelectric buzzer and electromagnetic buzzer. The one used in the test is an active buzzer, as shown in Figure 1. Active buzzers can generate continuous sound by directly connecting to a rated power (indicated on the label of the new buzzers). In this test, we use Iteaduino to drive the buzzer to make a simple song - "Happy Birthday".



Figure 1

To do this test, we will need:

- Iteaduino board x 1
- Buzzer electronic brick x 1

Construction of hardware

Connect buzzer electronic brick directly to D3 pin interface on Iteaduino with 3Pin connection cable, then the cathode of the buzzer is connected to GND terminal of Iteaduino board, while signal is connected to D3 pin of Iteaduino via S pin. The actual circuit construction is shown as in Figure 2:

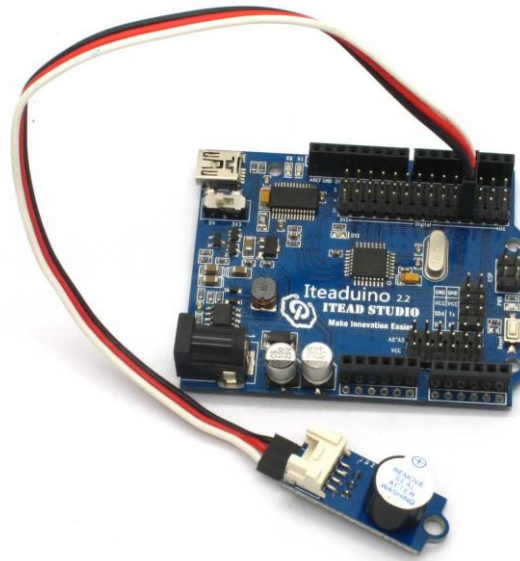


Figure 2

Writing of control program

We will need "Tone_Itead" library for this program, so we should download the library first :

<https://code.google.com/p/tone-itead/>

Tone_Itead library supports arduino1.0.1 and newer versions, in which we made some changes based on Tone library so that it is compatible with our Buzzer electronic brick (it makes sound when input low level).

Next, I will briefly introduce some functions in Tone library :

makeTone.begin(speakerPin)

Function : to select to control pins of buzzer

Parameters :

speakerPin : Pin N.O. for connecting to S port of buzzer

makeTone.play(note)

Function : to let buzzer make a sound

Parameters :

note : value of certain note



makeTone.stop()

Function : to stop making sounds

A song is composed of a number of notes, and a note corresponds to a frequency. If we know the corresponding frequency, and let Iteaduino output different frequencies to control the buzzer, the buzzer will make corresponding sounds. The relationship between different note and frequency is shown in the following three tables:

Tone							
Note	1	2	3	4	5	6	7
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371
Tone							
Note	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

Tone							



Note	1	2	3	4	5	6	7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

The notation of "Happy Birthday" is shown in Figure 3:

祝你生日快乐
HAPPY birthday TO YOU
(大众乐谱网站制谱)

1=F 5 5 6 5 1 7 — 5 5
 Hap - py birth_ day to you, Hap - py
 祝 你 生 日 快 乐, 祝 你

6 5 2 1 — 5 5 5 3 1
 birth day to you, Hap - py birth_ day to
 生 日 快 乐, 祝 你 生 日 快

7 6 4 4 3 1 2 1 —
 you, Hap - py birth_ day to you.
 乐, 祝 你 永 远 快 乐。

Figure 3

This song is in F, according to the corresponding table of notes to frequency, we can find that the corresponding frequency of 5 is 262Hz. In file Tone_Itead.h, there is "# define NOTE_C4 262", thus we can define frequency 262Hz as NOTE_C4, and other notes can also be analogized in the same way, and finally we can get the complete codes of all the notes { NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_F4, NOTE_E4, NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_G4,



```
NOTE_F4,NOTE_C4, NOTE_C4, NOTE_C5, NOTE_A4, NOTE_F4, NOTE_E4,NOTE_D4,NOTE_AS4,NOTE_AS4, NOTE_A4,  
NOTE_F4, NOTE_G4, NOTE_F4}
```

The complete program used in the test is as below:

```
#include <Tone_Itead.h>  
  
int speakerPin = 3;  
  
Tone makeTone;  
  
// notes to play; see Tone_Itead.h for frequencies;  
  
int notes[] = {  
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_F4, NOTE_E4,  
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_G4, NOTE_F4,  
    NOTE_C4, NOTE_C4, NOTE_C5, NOTE_A4, NOTE_F4, NOTE_E4, NOTE_D4,  
    NOTE_AS4, NOTE_AS4, NOTE_A4, NOTE_F4, NOTE_G4, NOTE_F4};  
  
// number of beats for each note  
  
int beats[] = {  
    1, 1, 2, 2, 2, 4,  
    1, 1, 2, 2, 2, 4,  
    1, 1, 2, 2, 2, 2, 2,  
    1, 1, 2, 2, 2, 4};  
  
// Calculate song length  
  
int songLength = sizeof(notes) / sizeof(int);  
  
int tempo = 220; // in milliseconds  
  
void playNote(int note, int beat){  
    makeTone.stop(); // speaker reset  
    makeTone.play(note); // play tone  
    delay(tempo * beat); // for specified number of beats  
    makeTone.stop(); // speaker reset
```



```
    delay(tempo / 4); // pause between notes
}

void setup() {
    pinMode(speakerPin,OUTPUT);
    makeTone.begin(speakerPin); // set up piezo speaker
}

void loop() {
    for (int i = 0; i < songLength; i++)
    {
        playNote(notes[i],beats[i]); // make sound
    }
}
```

Compile and download the above program into Iteaduino , the operation is the same as that in the last test.

After the program is downloaded, you can hear the buzzer play 《Happy birthday》 repeatedly.

You can also make other songs by yourself. First, find the music score of songs that you want to make, write the corresponding frequency of each note according to the above table, then find the corresponding name of each frequency, and finally change the contents in notes[] {} and beats [] {}.

The corresponding notes to different frequencies in file Tone_Itead.h range from B0 to D8, the corresponding frequencies are as follows:

```
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
```



```
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
```



```
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
```




To purchase the boards and electronic brick modules that you need, please visit our Taobao store:

<http://itead.taobao.com/>