ESP32实现驱动无源蜂鸣器

有源和无源蜂鸣器知识科普

有源和无源这里的“源”不是指电源，而是指震荡源。也就是说，有源蜂鸣器内部带震荡源，所以只要一通电就会叫。而无源内部不带震荡源，所以如果用直流信号无法令其鸣叫。必须用2K~5K的方波去驱动它。有源蜂鸣器往往比无源的贵，就是因为里面多个震荡电路。

连接电路图

图形用户界面, 应用程序

描述已自动生成

附录

**附录A：基础代码**

#include <Arduino.h>

//定义蜂鸣器连接的GPIO口

#define BUZZER\_PIN 23

void setup() {

  //设置BUZZER\_PIN为输出模式

  pinMode(BUZZER\_PIN,OUTPUT);}

void loop() {

  //发出蜂鸣声

  tone(BUZZER\_PIN,1000);//发出1000Hz的频率

  delay(1000);//持续1秒钟

  noTone(BUZZER\_PIN);//停止发声

  delay(1000);//延迟1秒钟

}

**附录B：周杰伦《晴天》**

#define buzzer 23

int freq[] = {131, 147, 165, 175, 196, 220, 247,262, 294, 330, 349,370, 392,411, 440, 494,523, 587, 659, 698, 784, 880, 988};

//这里从左往右对应低音中音高音7个音以及中音中4与5之间的半音，5与6之间的半音的频率

int qianzou[] = {5, 7, 12, 7, 3, 7, 12, 7, 5, 7, 12, 7, 0, 12, 6, 7};

//这些不是对应的简谱的音阶，而是对应的音阶在freq数组里的序号

int part1[]={12,12,7,7,8,9};

int part2[]={12,12,7,7,8,9,8,7,4};

int part3[]={12,12,7,7,8,9};

int part4[]={9,10,9,8,10,9,8,7};

int part5[]={4,7,8,9,10,9,8,7};

int part6[]={8,9,9,9,9,8,9,8,7,15};

int part7[]={7,7,7,7,7,6,7,7};

int part8[]={7,7,7,6,7,7};

int part9[]={7,7,7,12,12,12};

int part10[]={12,12,12,12,12,12};

int part11[]={12,12,12,12,10,9};

int part12[]={7,7,7,7,5,6,7,12,10,9,7,7};

int part13[]={7,7,7,7,9,7,5,6,7,12,10,9,7,8};

int part14[]={9,8,10,9};

int part15[]={7,12,15,16,15,12,7};

int part17[]={7,14,14};

int part18[]={14,12,12};

int part19[]={12,10,9,8,9,10,9};

int part20[]={9,11,13,9};

int part21[]={10,12,15,17,15,16,16};

int part22[]={16,16,12,12,14,12,10,8,9,10,12,14,7,14,15,15};

int part23[]={16,16,12,12,14,12,10,5,6,7,8,9,8,9,7};

int channel = 0;

int resolution = 8;

int noteDuration = 500;

void singing(int arr[],int size1)

{

  for(int i=0;i<size1;i++){

    ledcWriteTone(channel, freq[arr[i]]);

    delay(noteDuration);

  }

}

void setup() {

  Serial.begin(115200);

  ledcSetup(channel, freq[0], resolution);

  ledcAttachPin(buzzer, channel);

}

void loop()

{

  delay(1000);

  singing(qianzou,16);

  singing(qianzou,16);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part1,6);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part2,9);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part3,6);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part4,8);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part5,8);

  singing(part6,9);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part5,8);

  singing(part6,9);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part7,8);

  singing(part8,6);

  singing(part8,6);

  singing(part9,6);

  singing(part10,6);

  singing(part11,6);

  delay(500);

  ledcWriteTone(channel,0);

  delay(1000);

  singing(part12,12);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part13,14);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part14,4);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part15,7);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part17,3);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part18,3);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part19,7);

  ledcWriteTone(channel,0);

  delay(1000);

  singing(part20,4);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part21,7);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part22,16);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part14,4);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part15,7);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part17,3);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part18,3);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part19,7);

  ledcWriteTone(channel,0);

  delay(1000);

  singing(part20,4);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part21,7);

  ledcWriteTone(channel,0);

  delay(500);

  singing(part23,13);

  ledcWriteTone(channel,0);

  delay(500);

  ledcWriteTone(channel, freq[9]);

  delay(500);

  ledcWriteTone(channel, freq[7]);

  delay(800);

  ledcWriteTone(channel,0);

  delay(5000);

}