

به نام خدا

12/21/2021

گزارش آزمایش شماره 9 (موسیقی و header)

آزمایشگاه ریزپردازنده و زبان اسمبلی

محمد جواد زندیه , ابوالفضل بکیاسای کیوی
دانشگاه صنعتی امیرکبیر دانشکده مهندسی کامپیوتر

پرسش: چند مورد از کاربرد های پیزوالکتریک در دنیای واقعی را نام ببرید.

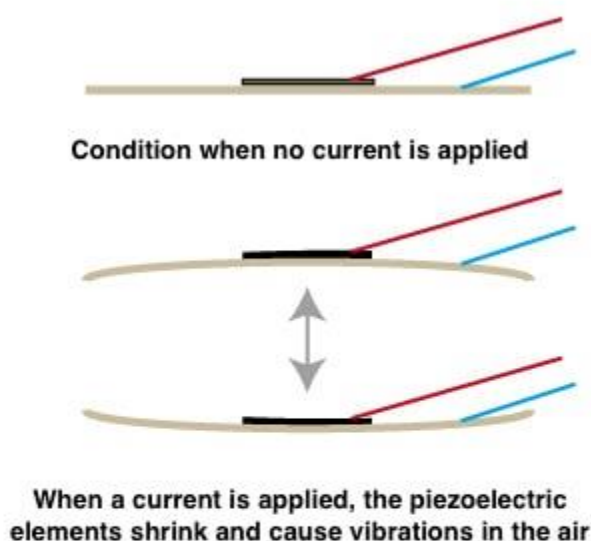
برای تولید صوت در مواردی که بازه فرکانسی محدودی موردنیاز باشد می توان بجای استفاده از بلندگو از پیزو استفاده نمود.

همچنین برای تولید هرگونه حرکت مکانیکی کوچکی می توان از این المان ریز استفاده نمود.

یک مثال کاربردی آن پرینتر های جوهر افشان است. در پرینتر های جوهر افشان، صفحه دیافراگمی پیزوالکتریک با نوسان خود باعث خروج تدریجی قطرات جوهر از نازل می گردد.

همچنین یکی از کاربرد هایی که جدیداً مورد استفاده از پیزوالکتریک هاست تولید برق از نوسان پیزو است. برای مثال با قرار دادن یک قطعه پیزو در کفش به هنگام راه رفتن می توان از پیزوالکتریک برق تولید کرد.

پرسش: اسپیکر پیزوالکتریک ما چگونه کار می کند؟ فکر می کنید چرا این روش کار انتخاب شده است؟



اگر ولتاژی که به پیزو وارد می شود تغییر نمایند، این قطعه سرامیکی نوسان خواهد داشت (به دلیل خاصیت پیزوالکتریک ها). با اتصال یک صفحه دیافراگمی به پیزو می توان از این نوسان تولید صدا نمود.

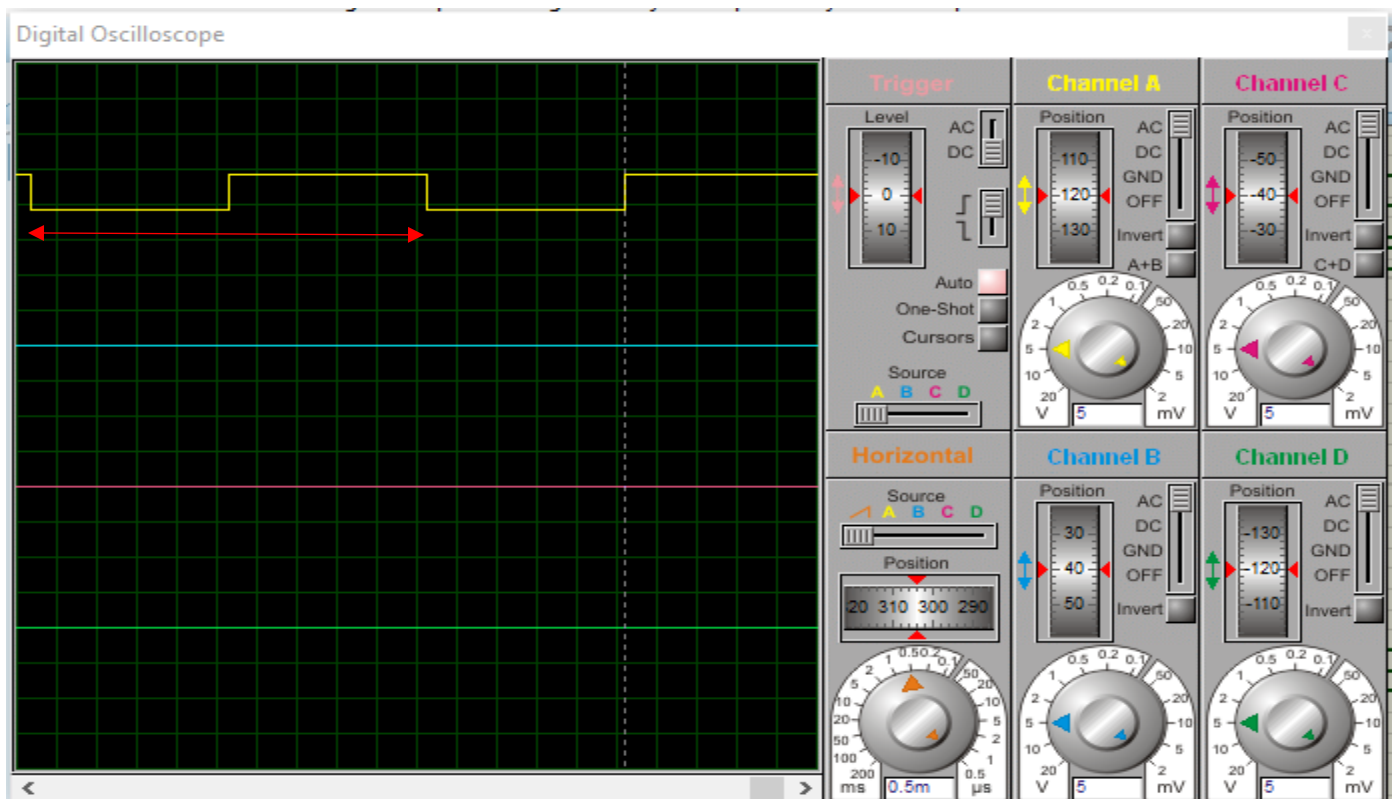
پرسش: تایمری که دستور tone استفاده می کند با خیلی از پین های برد مشترک است. بررسی کنید که به چه روش هایی می توانید آن تایمر را به هم بریزید که دستور tone خراب شود و صداهای مطلوب را اجرا نکند.

اگر از PWM در پین های 3 و 11 همزمان با دستور tone استفاده کنیم این تایمر نمی تواند به درستی کار خود را انجام دهد و دستور tone در این مورد محدودیت دارد.
اگر از فرکانس های زیر 31 هرتز استفاده کنیم این دستور نمی تواند درست عمل کند.
اگر همزمان از دستور tone در پین های دیگری هم استفاده شود، این تایمر نمیداند که تایم کدام را نگهدارد و دچار مشکل می شود.

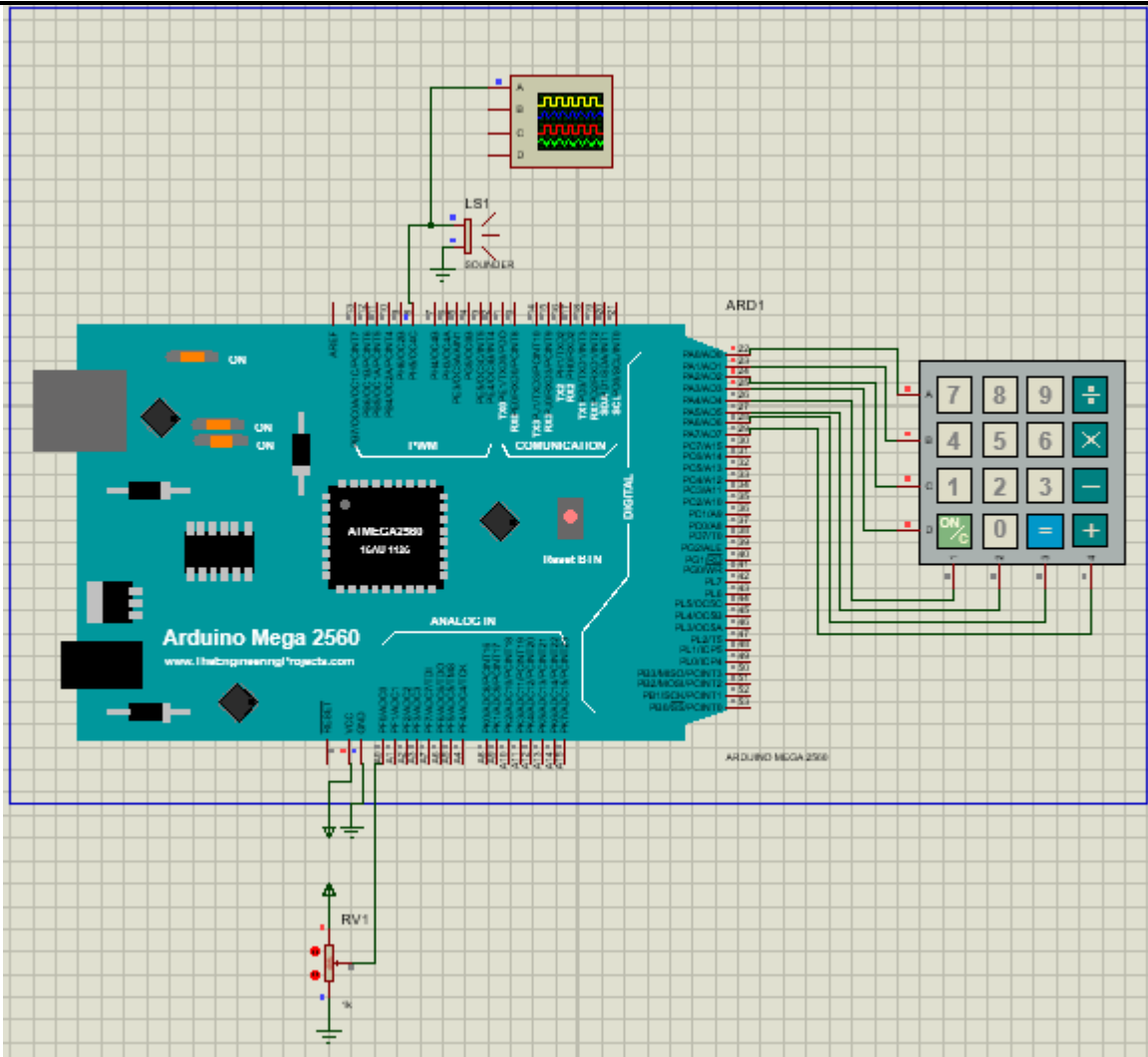
The limitations of the tone() function include:

1. Not being able to use PWM on pins 3 and 11 while you use tone()
2. You can't go lower than 31 hertz.
3. When using tone() on different pins, you have to turn off the tone on the current pin with noTone() before using tone() on a different pin.

پرسش: یک اسیلوسکوپ به سیم اسپیکر متصل کنید. چه اتفاقی دارد می افتد؟
پالس های مربعی را می بینیم که با تغییر فرکانس، صوت های مختلف را تولید می کند.
این تغییر فرکانس با تغییر عرض پالس قابل مشاهده است.



عرض پالس که با فلش قرمز نشان داده شده است در نت های مختلف متفاوت است و به دلیل تفاوت در فرکانسی است که تولید می شود توسط پین Arduino



شماره های 0 تا 5 از صفحه کلید هر یک برای تولید ملودی خاصی می باشد و ملودی شماره 5 همان ملودی Ode to Joy است که در دستور کار خواسته شده است و کد آن هم در زیر قرار داده شده است و ملودی های 0 تا 4 هم ملودی هایی است که به صورت آماده در فایل آزمایش موجود بود.

*هدر themes, themes1, themes2, themes3, themes4 در فایل پروژه پیش فرض بود.

کد پروژه :

```
#include "pitches.h"
#include "themes.h"
#include "themes1.h"
#include "themes2.h"
#include "themes3.h"
```

```

#include "themes4.h"
#include "Ode_to_Joy.h"
#include <Keypad.h>

//keypad object initializing
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
{'7','8','9','/'},
{'4','5','6','*'},
{'1','2','3','-'},
{'C','0','=','+'}
};
byte rowPins[ROWS] = {22, 23, 24, 25};
byte colPins[COLS] = {26, 27, 28, 29};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins,
ROWS, COLS);

float scale = 0;

void setup() {
  keypad.addEventListener(keypadEvent);
}

void loop() {
  char c = keypad.getKey();
}

void play_music(int mel[], int durs[], int len){
  for(int thisNote = 0; thisNote < len; thisNote++){
    scale = analogRead(A0)/512.0;
    int noteDuration = 500/(durs[thisNote]);
  }
}

```

```

    tone(8, (int)(mel[thisNote]*scale), noteDuration);
    delay((int)(noteDuration*1.3));
}
}

void keypadEvent(KeypadEvent key){
    if(keypad.getState() == PRESSED){
        if(key == '0')
            play_music(melody, noteDurations,
sizeof(melody)/sizeof(int));
        else if(key == '1')
            play_music(melody1, noteDurations1,
sizeof(melody1)/sizeof(int));
        else if(key == '2')
            play_music(melody2, noteDurations2,
sizeof(melody2)/sizeof(int));
        else if(key == '3')
            play_music(melody3, noteDurations3,
sizeof(melody3)/sizeof(int));
        else if(key == '4')
            play_music(melody4, noteDurations4,
sizeof(melody4)/sizeof(int));
        else if(key == '5'){
            for(int thisNote = 0; thisNote <
sizeof(melody5)/sizeof(int); thisNote++){
                scale = analogRead(A0)/512.0;
                int noteDuration = 500/(noteDurations5[thisNote]);
                tone(8, (int)(melody5[thisNote]*scale),
noteDuration);
                delay((int)(noteDuration*1.3));
            }
        }
    }
}

```

```
}  
}
```

هدر مربوط به Ode_to_Joy.h :

```
int melody5[] = {  
    NOTE_E4, NOTE_E4, NOTE_F4, NOTE_G4,  
    NOTE_G4, NOTE_F4, NOTE_E4, NOTE_D4,  
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_E4,  
    NOTE_E4, NOTE_D4, NOTE_D4,  
    NOTE_E4, NOTE_E4, NOTE_F4, NOTE_G4,  
    NOTE_G4, NOTE_F4, NOTE_E4, NOTE_D4,  
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_E4,  
    NOTE_D4, NOTE_C4, NOTE_C4,  
    NOTE_D4, NOTE_D4, NOTE_E4, NOTE_C4,  
    NOTE_D4, NOTE_E4, NOTE_F4, NOTE_E4, NOTE_C4,  
    NOTE_D4, NOTE_E4, NOTE_F4, NOTE_E4, NOTE_D4,  
    NOTE_C4, NOTE_D4, 0, 0  
};  
float noteDurations5[] = {  
    2, 2, 2, 2,  
    2, 2, 2, 2,  
    2, 2, 2, 2,  
    1.33, 4, 1,  
    2, 2, 2, 2,  
    2, 2, 2, 2,  
    2, 2, 2, 2,  
    1.33, 4, 1,  
    2, 2, 2, 2,  
    2, 4, 4, 2, 2,  
    2, 4, 4, 2, 2,  
    2, 2, 2, 2  
};
```