به نام خدا

آزمایش شماره 9: موسیقی و header

سپهر توکلی ـ ۹۸۳۱۱۱۱ محمد چوپان ـ ۹۸۳۱۱۲۵

پیشگزارش :

اسپیکر پیزوالکتریک چطور کار میکند؟ فکر میکنید چرا این روش انتخاب شده؟

مواد پیزوالکتریک موادی اند که بر اثر تغییر شکل، تغییرات ولتاژ ایجاد میکنند و برعکس، تغییرات ولتاز میتواند در آنها تغییر شکل ایجاد کند .در اسپیکر های پیزوالکتریک، اعمال تغییرات ولتاژ در قالب موج مربعی باعث تغییر شکل ورقهی پیزوالکتریکی در اسپیکر میشود و ورقه نوسان میکند و با اتصال یک صفحه دیافراگمی به پیزو میتوان از این نوسان صدا تولید کرد. و این تغییرات با فرکانس های متخلف باعث ایجاد صدا با فرکانس های مختلف میشود .دلیل استفاده از این روش این است که با استفاده از این روش میتوان صداهایی با فرکانس های مختلف (بم یا زیر) تولید کرد و مواد با خاصیت پیزوالکتریکی هم کمیاب نیستند و ساخت این اسپیکر ها راحت است .

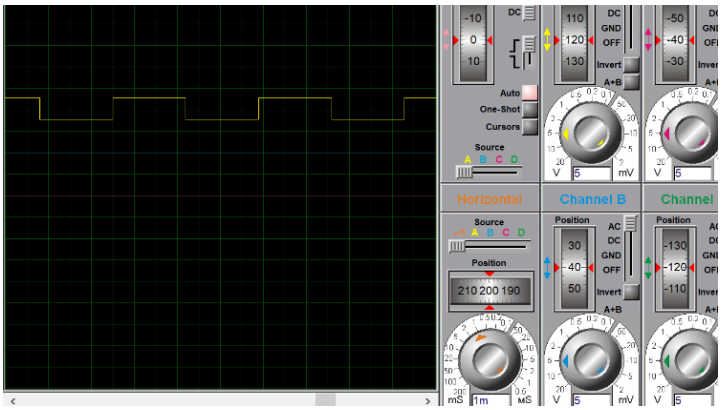
تایمری که دستور tone**استفاده میکند با خیلی از پین های برد مشترک است. بررسی کنید که به چه روش هایی میتوانید آن تایمر را به هم بریزید که دستور** tone**خراب شود و صداهای مطلوب را اجرا نکند** .

به طور کلی میتوان کلاک خود بورد را تغییر داد.
هنگامی که پیاده سازی دستور toneآغاز می شود، اجرا دستورات بعدی متوقف نمی شود و منتظر نمی ماند تا اجرای یک دستور به پایان برسد بنابراین درصورتی که استفاده های دیگری از پین در فرمان های بعدی کرده باشیم برای مثال چندین اجرا toneداشته باشیم همگی روی هم افتاده و تداخل می کند و به درستی اجرا نمی شود. برای حل این مشکل به ازای هر toneباید یک delay

داشته باشیم به اندازه ای که اجرای این toneطول می کشد. درغیر این صورت دستور tone خراب شده و صدای مطلوب را پخش نخواهد کرد .

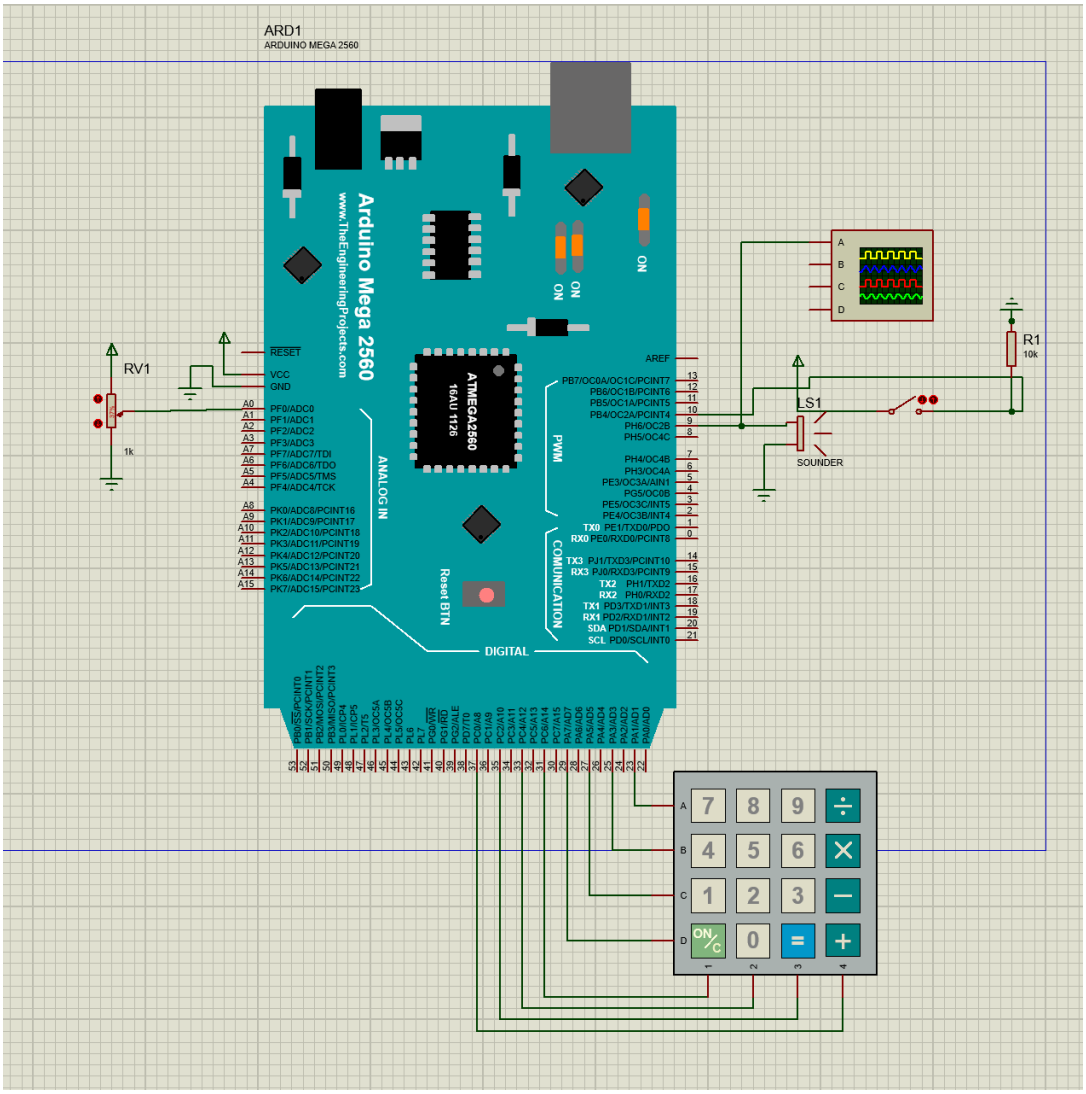
یک اوسیلوسکوپ به سیم اسپیکر متصل کنید. چه اتفاقی می افتد؟

موج های مربعی با duty 50% cycleدیده میشود که با فرکانس های مختلف پشت هم تکرار میشوند .



شرح آزمایش :

در ابتدا مدار مورد نیاز شامل یک کلید، یک صفحه کلید، یک اسپیکر و یک رئوستا را به صورت زیر بستیم :



1. **برنامه ای بنویسید که هنگام فشرده شدن هر کلید فایل ملودی متناظر با آن را پخش کند .**

برای انجام این کار از فایل هایی که در کانال برای مشخص کردن فرکانس نت ها و نت ملودی ها بود استفاده کردیم .

و کد زیر را نوشتیم :

```
#include "pitches.h"
#include "themes.h"
#include <Keypad.h>

const byte rows = 4; //four rows
const byte cols = 4; //four columns
char keys[rows][cols] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'o','0','=','+'}
};
byte rowPins[rows] = {23, 25, 27, 29}; //connect to the row pinouts of the keypad
byte colPins[cols] = {31, 33, 35, 37}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols );

#define PIEZO 9

void setup() {
  // put your setup code here, to run once:
    keypad.addEventListener(keypadEvent);
    pinMode(10,INPUT);
}

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

void playMelody(int melody[], float  noteDurations[], int m_size, int num) {
  for (int thisNote = 0; thisNote < m_size; thisNote++) {
    int key=digitalRead(10);
    float scale = analogRead(A0) / 512.0;
    if(key == LOW){
      scale = 1.0;
    }
    int noteDuration = num /  noteDurations[thisNote];
    tone(PIEZO, (int)(melody[thisNote] * scale), noteDuration);
    delay((int)noteDuration * 1.2);
  }

  noTone(9);
}
```

```
void keypadEvent(KeypadEvent key){
  if(key == '1'){
    playMelody(odeToJoy, odeToJoyDurations, sizeof(odeToJoy) / sizeof(int), 800);
  }
  if(key == '2'){
    playMelody(jingleBells, jingleBellsDurations, sizeof(jingleBells) / sizeof(int), 500);
  }
  if(key == '3'){
    playMelody(PiratesOfCaribbean, PiratesOfCaribbeanDurations, sizeof(PiratesOfCaribbean) / sizeof(int), 800);
  }
  if(key == '4'){
    playMelody(mario, marioDurations, sizeof(mario) / sizeof(int), 600);
  }
  if(key == '5'){
    playMelody(iran, iranDurations, sizeof(iran) / sizeof(int), 800);
  }
}
```

در ابتدای کد keypadرا تعریف کردیم و پین اسپیکر را مشخص کردیم و کتابخانه های مورد نیاز را includeکردیم .

در تابع setup، تابع keypadEventرا به eventlistenerدر keypadاضافه میکنیم و پین مربوط به پتانسیومتر را ورودی تعریف میکنیم .

در تابع loop، برنامه همواره منتظر ورود یک عدد از سمت keypadاست تا بعد از دریافت ورودی keypadEventرا اجرا کند .

در تابع keypadEvent، با توجه به عدد انتخاب شده از keypad، تابع playMelodyبرای آهنگ متناظر با آن ورودی صدا شده میشود .

در تابع playMelody، در یک for loopیکی یکی نت ها را به کمک تابع toneاجرا میکند

2 .فرکانس کلاک بورد را پایین بیاورید تا به جایی برسید که موسیقی به نرمی اجرا شود .متوجه میشوید که فرکانس نتها هم عوض می شود، پس با ایجاد تغییری کوچک درکد، آن را هم رفع کنید .

این کار انجام داده شد و برای روان تر اجرا شدن موسقی باید متغیر numدر ورودی تابع playMelodyرا تغییر میدادیم .

3یک پتانسیومتر به برد وصل کنید .همانطور که قبلا تابع mapورودی آنالوگ را استفاده کردید، اکنون برنامه ای بنویسید که هر دفعه که دکمه فشرده میشود، بنا به وضعیت پتانسیومتر یک نت زیرتر یا بمتر اجرا کند .

این کار در قسمتی از کد در تابع playMelodyبه کمک متغیر scaleانجام میشود. اگر دکمه فشرده شده باشد متغیر scaleبا استفاده از ورودی پتانسیومتر تعریف میشود و عددی بین 0 تا 2 خواهد داشت. اگر کلید فشار داده نشده باشد این متغیر مقدار 1 خواهد داشت. و در نهایت مقدار این متغیر در ورودی فرکانس تابع toneضرب میشود و با تغییر فرکانس صدا را بم تر یا زیر تر میکند .

```
float scale = analogRead(A0) / 512.0;
if(key == LOW){
  scale = 1.0;
}
int noteDuration = num /  noteDurations[thisNote];
tone(PIEZO, (int)(melody[thisNote] * scale), noteDuration);
```

۴)با دانشی که دارید، برنامه ای بنویسید که ملودی Ode to Joy بتهوون را پخش کند .

این بخش همراه با قسمت اول انجام داده شد. برای اینکار در فایل themes.h کد های زیر را اضافه کردیم :

```
int odeToJoy[] = {
  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 odeToJoyDurations[] = {
  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
};
```