

به نام خدا

عنوان آزمایش : اجرای ملودی ساده با استفاده از آردوینو و بیزر (Piezo Buzzer)

هدف آزمایش: هدف از این آزمایش، آشنایی با اصول تولید صدا با استفاده از بیزر و برنامه‌نویسی آردوینو است. در این آزمایش، قصد داریم ملودی ساده‌ای را به وسیله آردوینو و یک بیزر پیزو به صدا درآوریم و نحوه محاسبه مدت زمان هر نت و اجرای آن را مورد بررسی قرار دهیم.

وسایل آزمایش:

برد آردوینو (Arduino Uno) یا هر مدل مشابه

بیزر پیزو (Piezo Buzzer)

سیم‌های جامپر (Jumper Wires)

بردبورد (Breadboard)

مقاومت

شرح آزمایش: در این آزمایش، از آردوینو برای تولید صدا استفاده می‌کنیم. کد نوشته شده، شامل یک آرایه از نت‌ها و دقایق نت‌ها است که به کمک بیزر پیزو اجرا می‌شود. در ابتدا نت‌ها و مدت زمان هر نت در آرایه‌ها مشخص می‌شود، سپس کد آردوینو نت‌ها را به صورت متوالی پخش می‌کند.

نت‌ها در آرایه `melody[]` قرار دارند.

مدت زمان هر نت در آرایه `noteDurations[]` به صورت واحد تقسیم شده است (در اینجا، واحد اصلی ۱۰۰۰ میلی‌ثانیه است).

سپس برای هر نت از تابع `tone()` استفاده شده تا فرکانس صدای مربوط به هر نت در پین مشخص (پین ۸ در اینجا) تولید شود.

توضیحات آزمایش:

در این آزمایش، ابتدا یک آرایه از نت‌ها و دقایق آن‌ها به ترتیب مشخص می‌شود. با استفاده از این آرایه‌ها، آردوینو نت‌ها را یکی یکی تولید می‌کند. مدت زمان هر نت به کمک محاسبات ساده از روی تقسیم ۱۰۰۰ میلی‌ثانیه بر تعداد دقایق نت (یعنی ۴ برای نت ربع و ۸ برای نت هشتی) محاسبه می‌شود.

توسط دستور `tone(8, melody[thisNote], noteDuration)`، آردوینو فرکانس مناسب برای هر نت را از طریق بیزر ارسال می‌کند. در نهایت، پس از هر نت، یک تأخیر ۳۰ میلی‌ثانیه‌ای برای جداسازی صداها از هم قرار می‌دهیم.

توضیح کد نرم افزاری

pitches.h: وارد کردن کتابخانه

cpp

```
"include "pitches.h#
```

این کتابخانه به آردوینو اجازه می‌دهد تا از نام‌های منطقی برای نت‌های موسیقی استفاده کند. هر نُت، یک مقدار فرکانس معین دارد که در این کتابخانه تعریف شده است. مثلاً NOTE_C4 معادل فرکانس ۲۶۱/۶۳ هرتز است.

تعریف نُت‌ها و مدت زمان آن‌ها:

cpp

```
int melody[] = {NOTE_C4, NOTE_G3, NOTE_G3, NOTE_GS3, NOTE_G3, 0, NOTE_B3, NOTE_C4};
```

```
int noteDurations[] = {4, 8, 8, 4, 4, 4, 4, 4};
```

آرایه melody[] شامل نت‌های موسیقی است که باید پخش شوند.

آرایه noteDurations[] مدت زمان هر نُت را به صورت واحدی از نُت‌ها (ربع یا هشتم و غیره) مشخص می‌کند.

محاسبه مدت زمان هر نُت و پخش آن:

cpp

```
int noteDuration = 1000 / noteDurations[thisNote];
```

```
tone(8, melody[thisNote], noteDuration);
```

```
delay(noteDuration + 30);
```

مدت زمان هر نُت با تقسیم ۱۰۰۰ میلی‌ثانیه بر مقدار noteDurations[thisNote] محاسبه می‌شود.

سپس با استفاده از تابع tone(), فرکانس نُت مربوطه در پین ۸ آردوینو به مدت noteDuration تولید می‌شود.

با دستور delay(), پس از هر نُت یک توقف ۳۰ میلی‌ثانیه‌ای ایجاد می‌شود تا نُت‌ها از هم تفکیک شوند.

تکرار نشدن ملودی:

cpp

```
void loop() {
```

```
    // no need to repeat the melody } }.
```

```
#include "pitches.h"

// notes in the melody:

int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_G3, NOTE_G3, 0, NOTE_B3, NOTE_C4};

// note durations: 4 = quarter note, 8 = eighth note, etc.:

int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {
    // to calculate the note duration, take one second
    // divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000/noteDurations[thisNote];
    tone(8, melody[thisNote],noteDuration);
    //pause for the note's duration plus 30 ms:
    delay(noteDuration +30);
  }
}

void loop() {
  // no need to repeat the melody.
}
```

نحوه اتصال مدار: اتصال بیزر به آردینو

یکی از پایه‌های بیزر به پین ۸ آردینو متصل می‌شود.

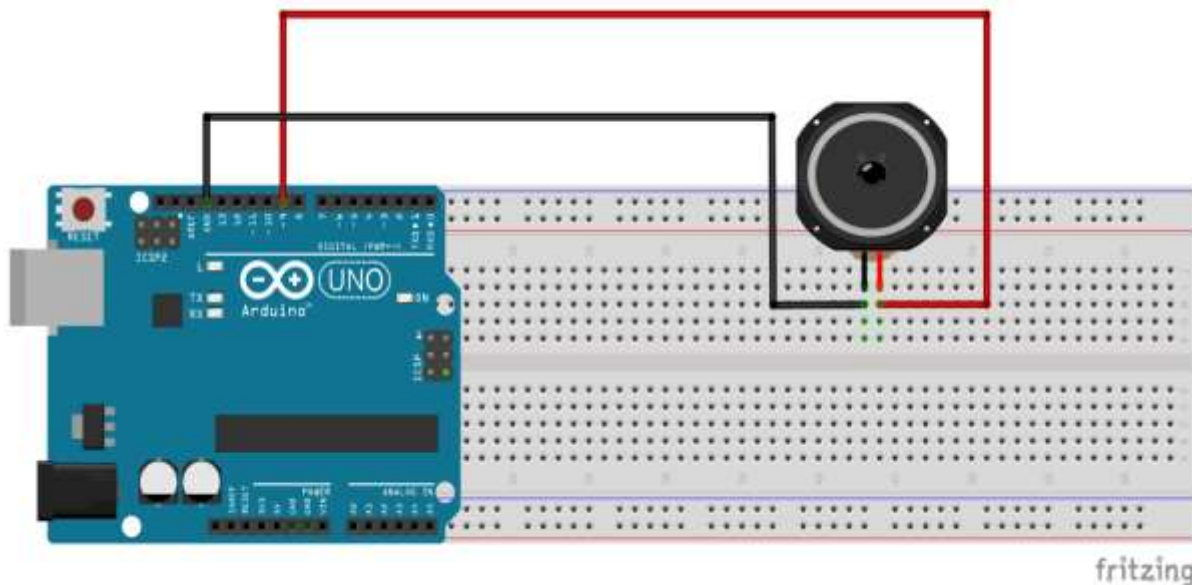
پایه دیگر بیزر به زمین (GND) متصل می‌شود.

در صورت استفاده از مقاومت، مقاومت باید بین پایه مثبت بیزر و پین ۸ آردینو قرار گیرد.

اتصال به کامپیوتر:

آردینو باید از طریق کابل USB به کامپیوتر متصل شود تا کد به آن بارگذاری شود.

شماتیک مدار:



نتیجه گیری: ما توانستیم یک ملودی ساده را با استفاده از آردینو و بیزر پیرو تولید کنیم. این آزمایش، اصول اولیه تولید صدا در آردینو را آموزش می‌دهد و به ما کمک می‌کند تا درک بهتری از نحوه استفاده از پین‌های دیجیتال برای تولید صدا و همچنین استفاده از تابع tone() در آردینو بدست آوریم.