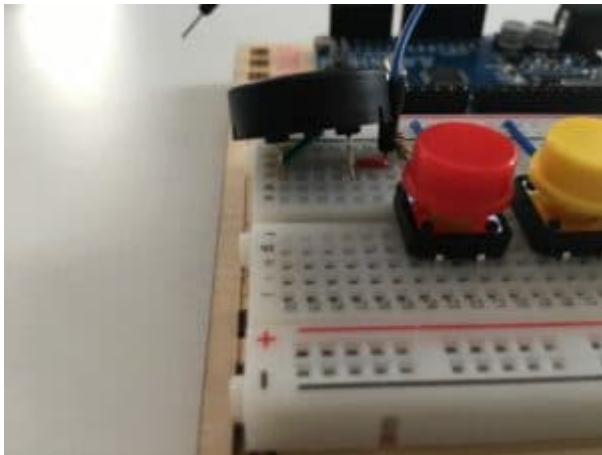


Mit einem Piezo Töne erzeugen



Wenn du mit deinem Arduino einfache Töne erzeugen möchtest, bietet sich ein günstiger Piezo an. Für musikalischen Hochgenuss sorgst du damit zwar nicht, aber für schlichte Anwendungen reicht er allemal.

Lerne in diesem Tutorial, wie du einen Piezo mit deinem Arduino verbindest und Töne in verschiedenen Höhen ausgibst.

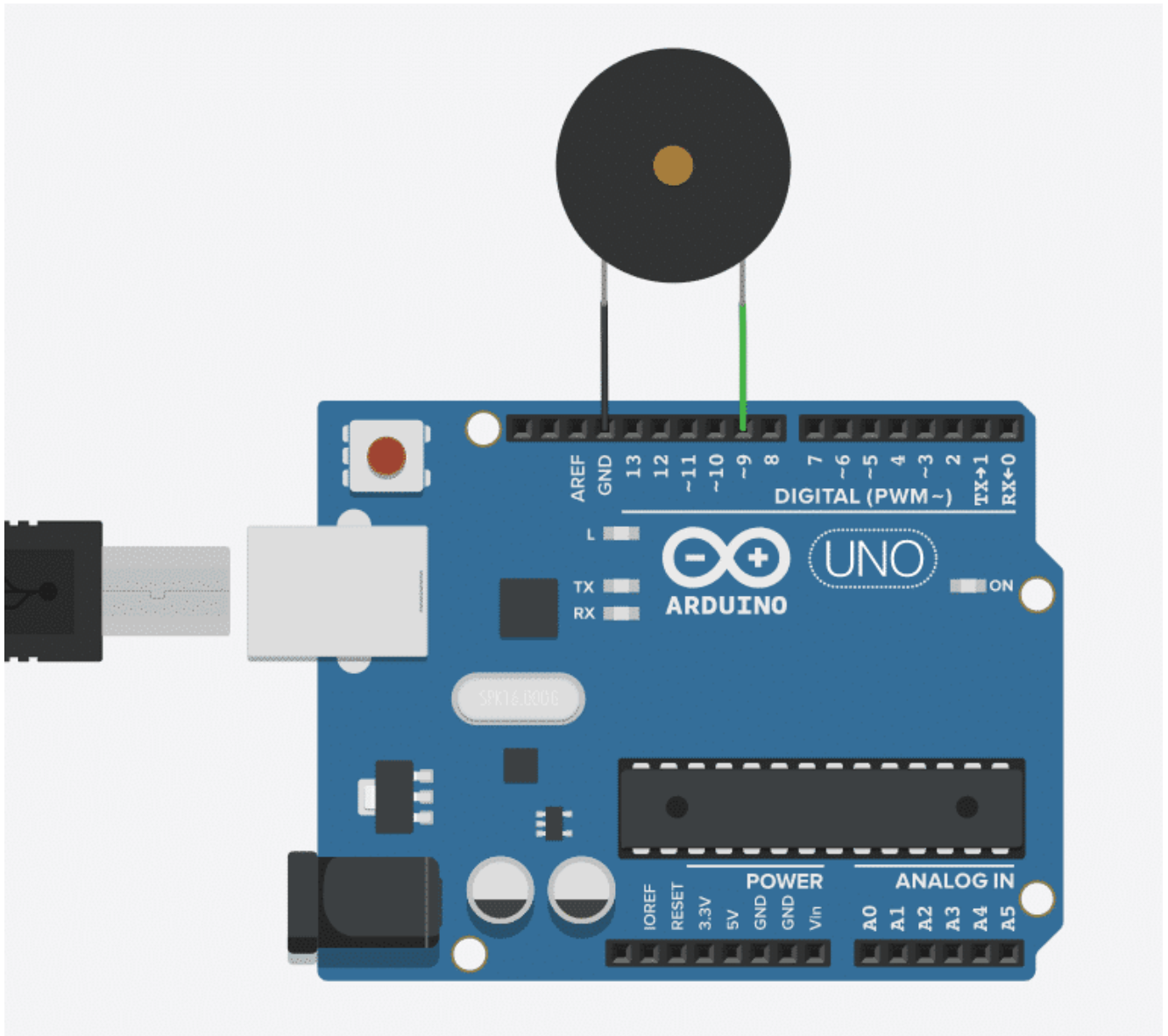
Aktiver vs. Passiver Piezo

Es gibt zwei Arten von Piezos: Aktive – hier wird einfach nur Strom angelegt, damit sie einen **einzigen vordefinierten** Ton abgeben. Passive: Diese Piezos eignen sich hingegen, verschiedene Töne, also auch Melodien zu erzeugen.

Wenn du also mehr als nur einen einzigen Ton erzeugen möchtest, **achte darauf, dir einen passiven Piezo zuzulegen**. Im Folgenden sprechen wir ausschließlich über passive Piezos.

Den Piezo am Arduino anschließen

Da dein Piezo nur über zwei Pins verfügt, benötigst du auch nur zwei Kabel, um ihn an deinem Arduino anzuschließen. Schließe zunächst einen der Pins – welcher ist egal – an GND an. Den anderen Pin verbindest du mit einem Digitalpin des Arduinos. Im folgenden Aufbau ist das Pin 9:



Screenshot: Tinkercad

Und das war auch schon alles. Zeit für etwas Musik!

Mit der Funktion `tone()` Töne erzeugen

Um Töne und Melodien mit verschieden langen Noten zu spielen, benötigst du nur die Funktion `tone()`. Du kannst auf diese Funktion direkt in deiner Arduino IDE zugreifen und musst hierfür nichts installieren oder vorbereiten.

Die Funktion rufst du immer mit mindestens zwei Parametern auf: dem Pin, an dem der Piezo angeschlossen ist und die

Tonhöhe in Hertz.

Wenn dein Piezo also wie oben zu sehen an Digitalpin 9 angeschlossen ist und du den Kammerton A mit seinen 440 Hertz spielen möchtest, wäre der Code hierfür

```
tone(9, 440);
```

Dein Piezo spielt diesen Ton dann so lange, bis er den Befehl erhält, etwas anderes zu tun – z.B. einen anderen Ton auszugeben.

Du hast allerdings die Möglichkeit, die Tonlänge direkt über einen dritten Parameter vorzugeben. Soll der Piezo den Ton A genau 1000 Millisekunden spielen, trägst du folgendes in deinen Sketch ein:

```
tone(9, 440, 1000);
```

Wenn du deinen Piezo ganz ausschalten möchtest, erreichst du das ganz einfach mit der Funktion **noTone()**.

```
noTone(9);
```

Noten und Melodien spielen

Jeder Note entspricht eine bestimmte Frequenz. Wie du oben gesehen hast, ist die Frequenz des Kammertons A genau 440Hz. Das A in der nächsten Oktave hat die Frequenz 880Hz.

Eine sehr umfangreiche Sammlung von Tönen und ihren entsprechenden Frequenzen [findest du hier auf Github](#).

Zu jeder guten Melodie gehören auch Pausen. **Diese kannst du wie gewohnt mit der Funktion delay() zwischen die Töne setzen.**

Instrumente bauen

Natürlich ist es eine Sache, alle Töne und ihre Abfolge im Sketch vorzugeben und dann abspielen zu lassen. Spannender wird es jedoch, wenn du sie selbst spontan nach deinen Wünschen spielen kannst – also ein richtiges Instrument baust.

Hier findest du zwei Arduino Projekte, mit denen du deinen Arduino als Instrument verwenden kannst:

[Arduino Orgel aus Überraschungseiern](#)

[Arduino Theremin mit A-Moll Pentatonik](#)