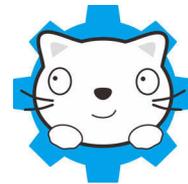


# Projekt 4-9: Sound-Machine

# 4 9

Was hältst du davon, wenn wir Scratch in diesem Kapitel dazu bewegen, etwas Musik zu spielen? Natürlich nicht von alleine, sondern über eine kleine selbstgebaute Tastatur. Scratch bringt von Haus aus schon einige Instrumente mit, über die wir nette Sounds erzeugen können. Wir werden über 12 Taster die einzelnen Sounds ansteuern und nicht einen einzigen digitalen Eingang dafür verwenden. Unmöglich? Sicherlich nicht! Du wirst dich möglicherweise fragen, wie das funktionieren soll, denn einige Taster müssen doch hinsichtlich ihres Status – gedrückt oder nicht gedrückt – abgefragt werden. Wie sollen die entsprechenden HIGH- oder LOW-Pegel zum Arduino übertragen werden, wenn nicht über die digitalen Eingänge? Werfen wir zuvor einen Blick auf die Tastatur.



## Die Sound-Machine

In der folgenden Abbildung siehst du auf der linken Seite die selbst hergestellte Tastatur mit den 12 kleinen Tastern und auf der rechten Seite das Arduino-Board:

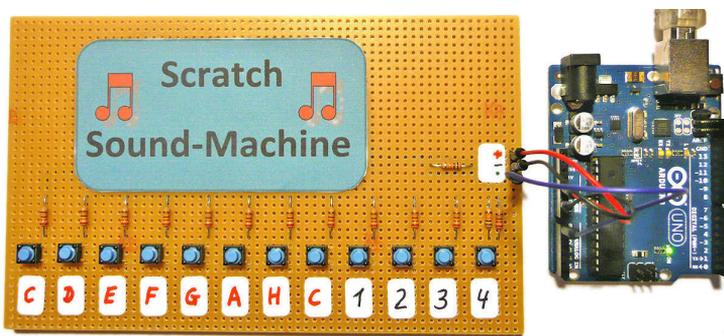


Abbildung 4-9-1: Die Sound-Machine-Tastatur und der Arduino

So wie es aussieht, ist keiner der digitalen Eingänge beschaltet, und es gibt lediglich drei Leitungen zwischen Tastatur und Arduino. Wenn du dir die Platine, auf der sich die Taster befinden, genauer anschaust, wirst du einige Widerstände erkennen. Und das ist genau der Punkt, auf den ich jetzt zu sprechen komme. Die Steuerung erfolgt über einen der analogen Eingänge des Arduino. Ich mache dir das mithilfe des folgenden Bildes etwas deutlicher:

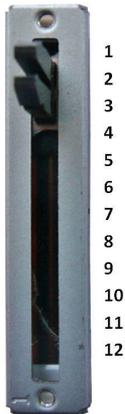


Abbildung 4-9-2: Ein Widerstandsregler

Wir haben es hier mit einem regelbaren Widerstand zu tun, der in Form eines Schiebereglers realisiert wurde. Du kannst seinen Widerstandswert vergrößern oder verkleinern, indem du den Regler nach oben oder nach unten schiebst. Rechts daneben befindet sich eine Skala, die die einzelnen Ziffern anzeigt, bei denen du den Regler positionieren musst, um einem bestimmten Widerstandswert zu erzielen. Da es hier keine festen Raster gibt, die an den markierten Stellen den Regler festhalten, ist diese Lösung natürlich sehr ungenau. Die Tastatur für die Sound-Machine arbeitet in ähnlicher Weise. Drückst du eine Taste, dann wird ein bestimmter Gesamtwiderstand aus einer diversen Zahl von Einzelwiderständen zusammengesetzt, der natürlich maßgeblich davon abhängt, welche Taste du drückst.

### **Jetzt Du!**

Du kannst die Schaltung auch auf einen Breadboard aufbauen und musst nicht die von mir gezeigte Platine im Eurokartenformat verwenden.

## **Benötigte Bauteile**

Für dieses Beispiel benötigen wir die folgenden Bauteile.

## Benötigte Bauteile



12 x Taster



12 x Widerstand 220



2 x Widerstand 10K



Mehrere flexible Steckbrücken in unterschiedlichen Farben und Längen

## Was sagt der Schaltplan?

An dieser Stelle möchte ich dir einen Schaltplan zeigen, der dir das Prinzip verdeutlicht:

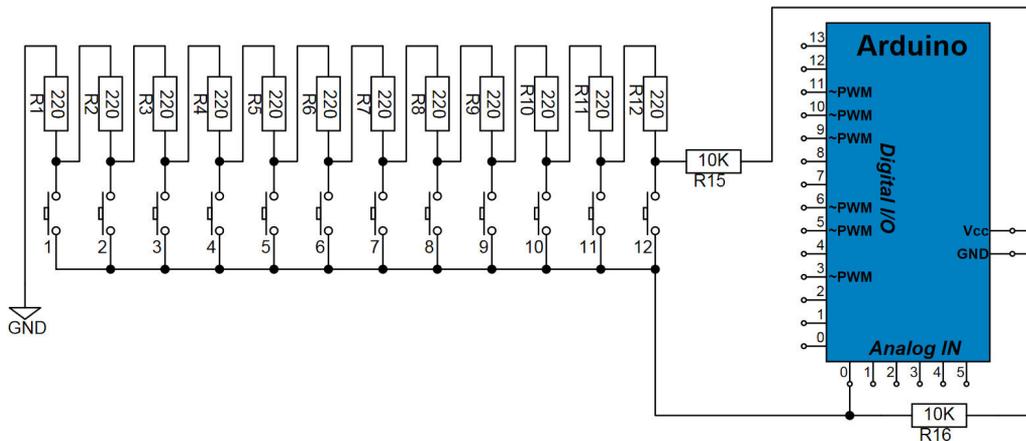


Abbildung 4-9-3: Der Schaltplan für die Sound-Machine

Die Widerstände mit dem Wert 220 Ohm bilden mit dem Widerstand R15 von 10K einen Spannungsteiler, bei dem je nach gedrückter Taste von rechts nach links gesehen ein Widerstand hinzugeschaltet wird. Wir haben Spannungsteiler schon im *Was ist Scratch*-Kapitel beim Thema *S4A und die Ports des Arduino* behandelt; dort kannst du ggf. noch einmal nachlesen.

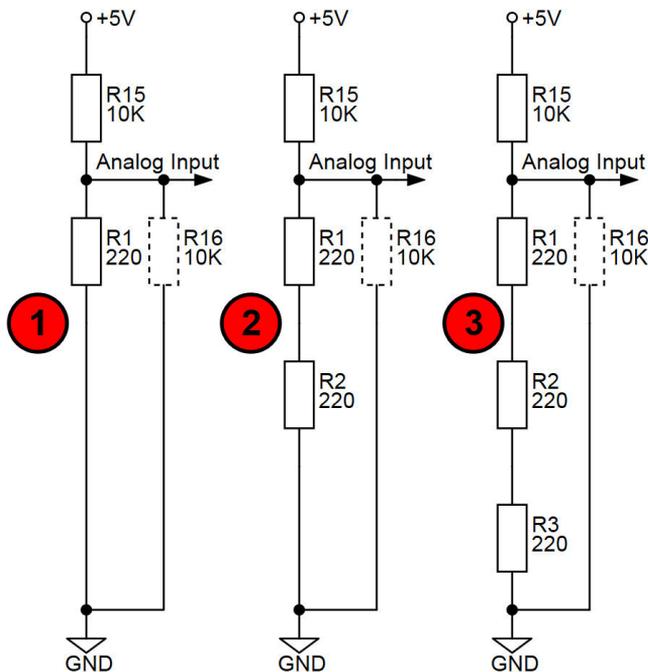


Abbildung 4-9-4: Das Ersatzschaltbild für die ersten drei Taster ( 1, 2 und 3)

Drückst du Taster 1, wird der erste 220-Ohm-Widerstand dem Spannungsteiler hinzugefügt, bei Taster 2 kommt ein weiterer 220-Ohm-Widerstand hinzu und so weiter und so fort. Je mehr untere Widerstände hinzugefügt werden, desto größer wird das Spannungspotenzial, das dem analogen Eingang des Arduino zugeführt wird.

Der untere Widerstand R16 im Schaltplan bzw. Ersatzschaltbild für die Sound-Machine arbeitet übrigens als Pull-down-Widerstand, um bei geöffneten Tastern einen definierten Pegel von 0V am analogen Eingang zu gewährleisten. In der nachfolgenden Tabelle siehst du die Werte, die dir Scratch auf der Bühne anzeigt, wenn du die einzelnen Taster nacheinander drückst.

Taste	1	2	3	4	5	6	7	8	9	10	11	12
Wert	15	32	48	64	79	93	108	122	136	149	162	175

Diese Werte können bei dir etwas abweichen, da die Widerstände eine bestimmte Bauteiltoleranz aufweisen. Teste es einfach aus und passe sie ggf. an, das sollte kein allzu großes Problem darstellen. Du kannst erkennen, dass die gemessene Spannung von rechts nach links kontinuierlich zunimmt, wobei die Differenz zwischen benachbarten Werten im Schnitt 15 beträgt. Über die Variable *Value*, die wir gleich noch anlegen werden, habe ich mir die einzelnen Werte auf der Bühne anzeigen lassen.

setze Value auf lese analogen Pin (A) 0

## Das Scratch-Skript

Das Scratch-Skript habe ich in mehrere Teile gesplittet, so dass es ein wenig übersichtlicher wird.

### Skript

Die folgenden Variablen müssen zuvor angelegt werden, wobei ich auf ihre Bedeutung gleich näher eingehe:



Abbildung 4-9-5: Die benötigten Variablen

Es ist lediglich für die Variable *Value* sinnvoll, sich ihren Wert bei verschiedenen Tastendrücken auf der Bühne anzeigen zu lassen, so dass ich auch nur hier das Häkchen gesetzt habe.

### Initialisierung der Variablen



Abbildung 4-9-6: Die Initialisierung der Variablen

Was passiert bei der Ausführung der einzelnen Blöcke:

- A Über den *Setze-auf*-Block wird die Variable *threshold* mit dem Wert 5 belegt. Diese Variable wird im Skript u. a. für die eindeutige Identifizierung eines Tastendrucks benötigt. Ich hatte schon die Bauteiltoleranzen bei Widerständen erwähnt; der Wert von *threshold* wird dem ermittelten Wert am analogen Eingang hinzugefügt bzw. abgezogen, um damit einen gewissen Toleranzbereich zu schaffen. Wir kommen gleich noch darauf zu sprechen.

- B Über den *Setze-auf*-Block wird die Variable *beatsInstrument* mit dem Wert 0,5 versehen und bestimmt, wie lange ein Instrument klingen soll. Ändere diesen Wert, um die Auswirkungen zu sehen.
- C Über den *Setze-auf*-Block wird die Variable *beatsDrum* mit dem Wert 0,5 versehen und bestimmt, wie lange das Schlagzeug klingen soll. Ändere diesen Wert, um die Auswirkungen zu erkennen.
- D Über den *Setze-Instrument-auf*-Block wird das gewünschte Instrument ausgewählt. Der Wert 6 steht für den Bass. Klickst du auf die Ziffer innerhalb des Blocks, erscheint eine Auswahlliste, so dass du dir um die nichtssagenden Zahlen keine Sorgen machen musst.

## Spiele die Töne

Das folgende Scratch-Skript spielt die Töne eines gewählten Instruments an:

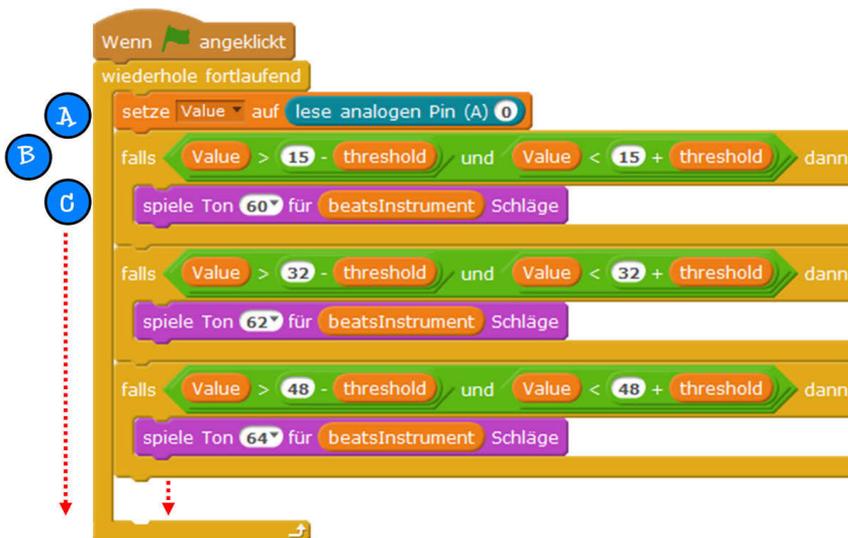


Abbildung 4-9-7: Das Spielen der Töne

Was passiert bei der Ausführung der einzelnen Blöcke:

- A Über den *Setze-auf*-Block wird die Variable *Value* durch Abfrage des analogen Wertes am Eingang A0 mithilfe des *Lese-analogen-Pin*-Blocks initialisiert. Die Variable wird jetzt in allen nachfolgenden Blöcken ausgewertet, um darüber einen entsprechenden Ton zu spielen.
- B Mithilfe des *Falls-dann*-Blocks und der nachfolgenden Bedingung wird ermittelt, welcher Ton gespielt werden soll.



Der Wert 15 ist aus der Wertetabelle entnommen, alle weiteren *Falls-dann*-Blöcke müssen lediglich mit den nachfolgenden Werten bis 122 fortgesetzt werden. Die Werte 136, 149, 162, und 175 sind für die Drums reserviert und werden gleich erläutert.

- Ⓒ Über den *Spieler-Ton*-Block wird das zuvor ausgewählte Instrument aus der Variablen-Initialisierung mit der Tonhöhe (hier 60: Middle C) bzw. Spieldauer, die in der Variablen *beatsInstrument* festgelegt wurde, angespielt.

## Spieler die Drums

Das folgende Scratch-Skript spielt die Drums an:

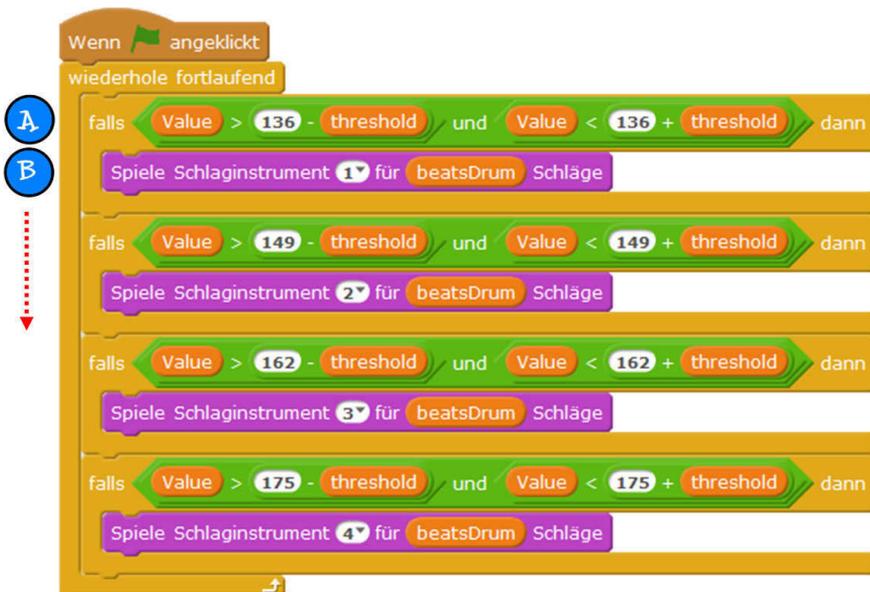


Abbildung 4-9-8: Das Spielen der Drums

Was passiert bei der Ausführung der einzelnen Blöcke:

- Ⓐ Über den *Falls-dann*-Block mit der nachfolgenden Bedingung wird ermittelt, ob der nachfolgende Block in B ausgeführt werden soll.
- Ⓑ Über den *Spieler-Schlaginstrument*-Block wird über den Wert 1 die *Snare-Drum* mit der Dauer, die in der Variablen *beatsDrum* hinterlegt ist, angespielt. Klickst du auf die Ziffer 1, werden dir alle zur Verfügung stehenden Drums in einer Auswahlliste angezeigt. Alle nachfolgenden *Falls-dann*-Blöcke werden in gleicher Weise abgearbeitet, um darüber die *Bass-Drum*, *Side-Stick* und *Crash-Cymbal* anzuspielden.

Alle hier gezeigten Einstellungen bzw. Vorgaben hinsichtlich der Instrumente kannst du natürlich deinen Bedürfnissen anpassen.