



In dieser Anwendung wirst du ein Programm entwickeln, das dich die aus dem Lautsprecher kommenden Töne über den Lichtsensor steuern lässt, indem du die Hand über ihn bewegst – Handmusik.

Lernziele:

- Helligkeitswerte in Töne konvertieren
- Den „12. Wurzel aus 2“-Zusammenhang zwischen Musiknoten und Frequenzen wiederholen

Schreibe ein Programm, das die BRIGHTNESS des Lichtsensors abliest und einen Ton abhängig vom Helligkeitswert abspielt. Für diesen Ton gibt es zwei Möglichkeiten:

- Spiele eine Frequenz im hörbaren Bereich (etwa 100 Hz bis 1000 Hz).
- Spiele eine Musiknote (einen der harmonischen Töne, die auf einem Musikinstrument gespielt werden).

Bei der ersten Option wirst du nur etwas wie Geräusche oder Lärm erzeugen. Mit der anderen Option wird etwas wie Musik entstehen, aber die Mathematik dahinter ist etwas komplizierter.

Dieses Programm lässt den TI-Innovator™ Hub wie einen *Theremin* funktionieren.

Wir beginnen:

1. Beginne ein neues Programm und nenne es *anwendung3*.
2. Füge ein **Disp**, und den Text „Handmusik!“ zwischen Anführungszeichen an wie gezeigt.
3. Füge ein **While...EndWhile**-Schleife zum Ablesen der Helligkeit an mit: **Send “READ BRIGHTNESS”** und speichere die Variable mit **Get b**.
4. Hänge die Anweisung zum Abspielen eines Tons dran.
 - Wir verwenden die Variable **b** für den Helligkeitswert (BRIGHTNESS) und die Variable **f** für den Ton (SOUND).

```

*anwendung3 5/10
local b,f,f#
Disp "Handmusik!"
b:=2
While b>1
  Send "READ BRIGHTNESS "
  Get b
  Send "SET SOUND eval(f)"
  Wait 0.2
EndWhile
  
```

Es ist nun deine Aufgabe, den Teil, der den Helligkeitswert in einen hörbaren *Ton* oder in eine *Musiknote* umwandelt, einzufügen.

Nimm für den Ton eine Frequenz zwischen 100 Hz und 1000 Hz (oder einen anderen Bereich deiner Wahl).

Für Musiknoten wähle für den Bereich den Startwert A1 (55 Hz) und gehe 50 Noten weiter. (Beziehe dich auf das Programm *ton2* in Lektion 2, Übung 3, das 12 Noten einer Oktave abspielen lässt.)

Für die Musiknoten musst du deinen Wert in eine ganze Zahl umwandeln, so dass die „Notennummer“ richtig ausgedrückt wird. Dazu kannst du entweder die **int()**-Funktion oder die **round(,0)**-Funktion verwenden.

- int(x)→x** ergibt die größte ganze Zahl < x.
- round(x,0)→x** rundet x auf die nächste ganze Zahl.

```

anwendung3()
Handmusik!
51. 25. 233.
56. 27. 262.
100. 49. 932.
60. 29. 294.
100. 49. 932.
100. 49. 932.
  
```

anwend	Get b
n:=int	
f:=55	
Disp b	
Send "	
Wait 0	
EndWhil	



Hinweis:

Wenn nur Töne erzeugt werden sollen, müssen die Schüler den Wert B aus $[0,100)$ in einen Wert F aus $[100,1000]$ umwandeln.

B	F
0	100
100	1000

Berechne die Steigung der Geraden, die diese beiden Punkte verbindet und schreibe die Gleichung für F als Funktion von B.

$$m = (1000 - 100)/(100 - 0) = 9$$

daher:

$$F = 9 \cdot B + 100$$

Für die Musiknoten musst du dich an die $F \cdot 2^{(1/12)}$ -Eigenschaft der Notenintervalle erinnern. A1 = 55Hz und wir wollen 50 Noten erzeugen. Die erste Note bekommt die Nummer 0 und die letzte die Nummer 49. (Programmierer beginnen eine Zählung immer mit 0). Denke auch daran, dass eine Notenummer nur eine ganze Zahl sein kann. Daher müssen wir entweder **int()** oder **round()** verwenden. In zwei Schritten machen wir die Rechnung klar:

$N = \text{int}(49 \cdot B/100)$	Nummer der Note
$F = 55 \cdot 2^{(N/12)}$	Notenfrequenz