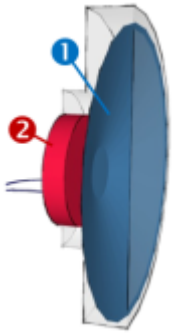
Töne und Funktionen

Hier lernst du, wie du mit dem Arduino und einem Lautsprecher Töne erzeugen und Tonfolgen abspielen kannst.



Viele Leute meinen, dass ein Lautsprecher sofort einen Ton von sich gebe, wenn er an eine Stromversorgung angeschlossen wird. Das ist aber falsch: Ein Lautsprecher besteht nämlich nur aus einer elastisch aufgehängten (1) Membran und einem (2) Elektromagnet. Fließt Elektrizität durch diesen Elektromagnet, wird die Membran angezogen, fließt keine Elektrizität, fällt die Membran wieder zurück. Ein Ton entsteht, indem die Membran schnell abwechselnd angezogen und wieder los gelassen wird, also die Stromversorgung abwechselnd ein- und ausgeschaltet wird.

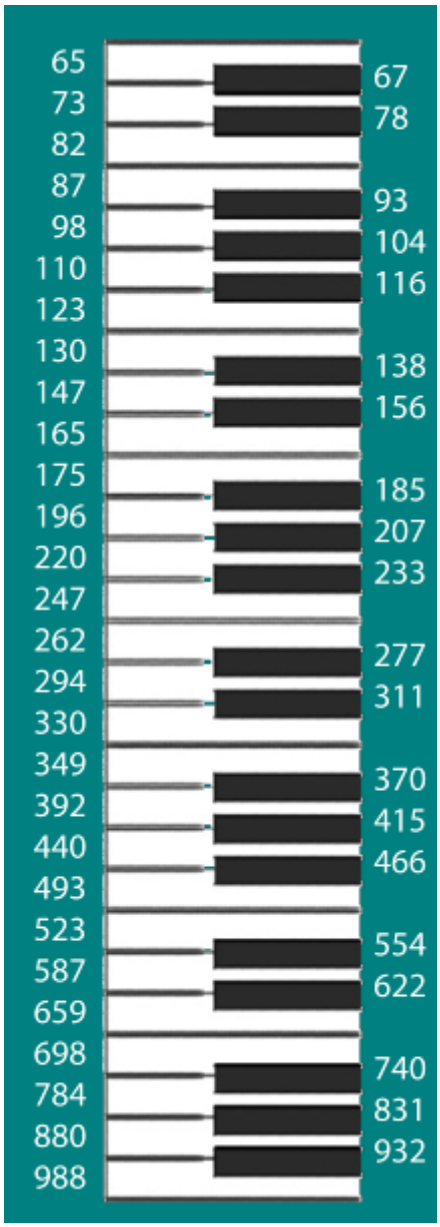
Um einen Ton zu erzeugen, kann der Lautsprecher also mit einem seiner Anschlüsse (egal welcher) zum Beispiel an Pin 11 und mit dem anderen an den Minuspol angeschlossen werden. Pin 11 muss schnell abwechselnd HIGH und LOW geschaltet werden. Weil man Töne häufig braucht, gibt es eine Anweisung, die sich um das schnelle Ein- und Aus-schalten kümmert, selbst wenn der Mikrocontroller gerade etwas anderes macht.

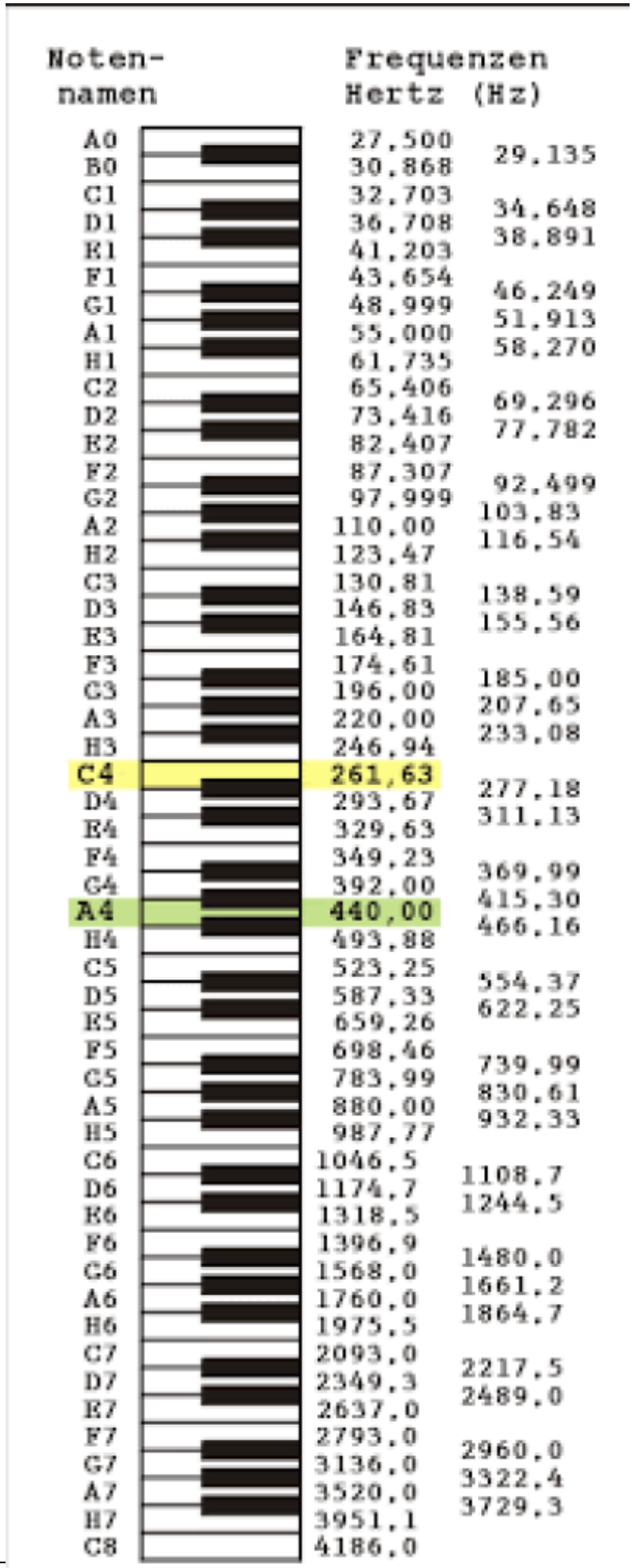
Der Programmcode sieht dann so aus:

```
void setup() {
}
void loop() {
 tone(11,440);
 delay(1000);
 noTone(11);
 delay(3000);
}
```

Diese Anweisung sorgt dafür, dass Pin 11 mit einer Frequenz von 440 Hz (Hertz) immer wieder HIGH- und LOW-geschaltet wird. Die Anweisung Tone sorgt übrigens vorher selbst dafür, dass Pin 11 in den pinMode OUTPUT geschaltet wird.

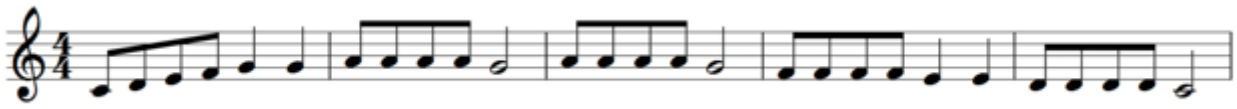
Diese Anweisung schaltet Pin 11 wieder dauerhaft auf LOW. Also ist der Ton aus.


65
73
82
87
98
110
123
130
147
165
175
196
220
247
262
294
330
349
392
440
493
523
587
659
698
784
880
988
67
78
93
104
116
138
156
185
207
233
277
311
370
415
466
554
622
740
831
932


Noten-
namen
Frequenzen
Hertz (Hz)
A0 27,500
B0 30,868
C1 32,703
D1 36,708
E1 41,203
F1 43,654
G1 48,999
A1 55,000
H1 61,735
C2 65,406
D2 73,416
E2 82,407
F2 87,307
G2 97,999
A2 110,00
H2 123,47
C3 130,81
D3 146,83
E3 164,81
F3 174,61
G3 196,00
A3 220,00
H3 246,94
C4 261,63
D4 293,67
E4 329,63
F4 349,23
G4 392,00
A4 440,00
H4 493,88
C5 523,25
D5 587,33
E5 659,26
F5 698,46
G5 783,99
A5 880,00
H5 987,77
C6 1046,5
D6 1174,7
E6 1318,5
F6 1396,9
G6 1568,0
A6 1760,0
H6 1975,5
C7 2093,0
D7 2349,3
E7 2637,0
F7 2793,0
G7 3136,0
A7 3520,0
H7 3951,1
C8 4186,0
29,135
34,648
38,891
46,249
51,913
58,270
69,296
77,782
92,499
103,83
116,54
138,59
155,56
185,00
207,65
233,08
277,18
311,13
369,99
415,30
466,16
554,37
622,25
739,99
830,61
932,33
1108,7
1244,5
1480,0
1661,2
1864,7
2217,5
2489,0
2960,0
3322,4
3729,3

Aufgabe 5.1

Schließe einen lautsprecher an einen Pin an und programmiere ein Lied. Wenn dir wirklich nichts besseres einfällt, kannst du „Alle meine Entchen nehmen“. Die Klaviertastatur rechts kann dir helfen, die Frequenzen zu den Noten herauszufinden.



Hinweis: Wenn zwei Töne direkt aufeinanderfolgen, benötigst du keine noTone Anweisung dazwischen. Es geht dann auch `tone(11,200); delay(120); tone(11,400); delay(500);tone(11,1000);…`

Unterprogramme

Programmteile, die mehrfach benötigt werden (wie z. B. ein Refrain), muss man nicht mehrfach programmieren. Man legt dafür ein sogenanntes Unterprogramm an, das man wie eine ganz gewöhnliche Anweisung überall dort aufruft, wo es ausgeführt werden soll.

Namensgebung: Für die Namen von Unterprogrammen kann man beliebige selbst gewählte Worte verwenden, die keine Leerzeichen und keine Sonderzeichen (wie Umlaute oder Symbole) und als erstes Zeichen auch keine Zahl enthalten. Möglich wären also oma, opa oder hundl6, unmöglich sind aber 16hund, rüpel oder a#. Dabei kommt es auf Groß- und Kleinschreibung an. Natürlich darf man keine Bezeichner verwenden, die schon als Befehle in der Programmiersprache verwendet werden.

```
void setup(){
}
void loop(){
  …
  refrain();
  …
  refrain();
  …
}
void refrain(){
  tone(11,220);
  delay(300);
  …
}
```

Am Ende fühlt sich ein Unterprogramm wie eine selbst geschriebene Anweisung an, hier zum Beispiel mit dem selbst gewählten Namen refrain(). Immer, wenn du refrain() schreibst, wird dein selbst definiertes Unterprogramm aufgerufen und ausgeführt.

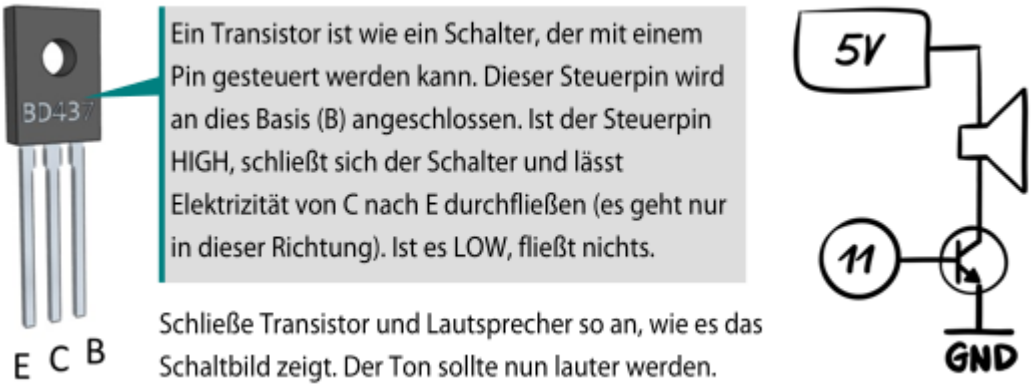
Hier siehst du, wie das Unterprogramm mit dem Namen refrain() definiert wird. Es sieht genau so aus wie setup oder loop, hat aber einen selbst gewählten Namen und natürlich einen eigenen Zweck.

Wie viele Zeilen kannst du in deinem Musikstück durch Unterprogramme sparen?

Aufgabe 5.2

Vielleicht hat auch dein Musikstück einen Takt oder einen Refrain, der sich wiederholt? Probiere, diesen Teil als Unterprogramm zu schreiben.

Ein Transistor als Verstärker



Dein Lautsprecher ist relativ leise, weil der Arduino durch seine Pins nur geringe elektrische Leistungen leiten kann. Höhere Leistung kannst du am Pin `V_in` beziehen, das aber nicht programmierbar sondern einfach immer an ist Mit einem Transistor kannst du trotzdem `V_in` für den Lautsprecher nutzen.

Die Anschlüsse des Transistors heißen Basis, Kollektor und Emitter. Der Transistor ist das technische Gerät das am häufigsten auf der Erde hergestellt wird. Das liegt auch daran, dass alleine in einem modernen Computerprozessor mehr als 1 Milliarde Transistoren verbaut sind. Im Mikro-controller auf dem Arduino sind es etwa 100000 - Transistoren können also sehr klein gebaut werden

ardulb1