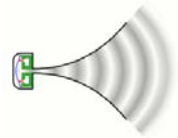
1. Vom Ton zur Melodie

Öffne in 02Digital den Sketch **toneMelody**.
Verkabele die Hardware und lasse den Sketch laufen.
Gestalte nun deine eigene Melodie.

2. Gewürfelte Töne

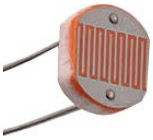
Es sollen sehr kurze Töne nach dem Zufallsprinzip **random(x,y)** abgespielt werden. Der Frequenzumfang beläuft sich von z.B. 300Hz bis 1500Hz. Erzeuge einen Sketch und teste ihn (bitte nur kurz!! ☺).
Beachte: Mit **tone(x,y)** läuft ein Ton an Pin(x) mit der Frequenz(y) so lange, bis die Anweisung **noTone(x)** oder eine neue Tonanweisung kommt.

3. Akustische Sirene



Programmiere einen Arduino so, dass er zunächst ein anschwellender Sirenenton zu hören ist.
Erweitere den Sketch nun geeignet um einen „richtigen" Sirenenton zu erzeugen (300 bis 800Hz).....nicht nerven! ☺

4. Töne und noch mehr Töne



Hier arbeiten wir mit einem LDR-Widerstand (**L**ight **D**ependend **R**esistor), dessen Wert umso kleiner wird, je mehr Licht auf ihn trifft.
Öffne den Beispiel-Sketch **tonePitchFollower** (Tonhöhenfolger) in 02Digital. Baue die Schaltung laut Anweisung auf. Der 4,7K-Ohm Widerstand hat die Farbcodierung *gelb-violett-rot*.
Ersetze zunächst in der Anweisung **tone(9, thisPitch, 10);** **„thisPitch"** durch **„sensorReading"**. Lasse den Sketch laufen, verdunkle den Fotowiderstand und beobachte dabei auch den seriellen Monitor.
Verwende nun den originalen Sketch und überlege, was die **map**-Anweisung bewirkt.

5. Lichtgesteuerte Ampel

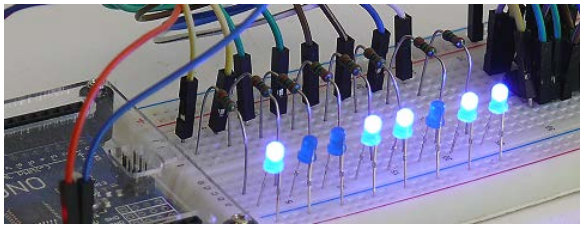
Nur ein paar kleine Änderungen und Zusätze und schon

wird aus dem tonePitchFollower eine helligkeits-gesteuerte Verkehrsampel. Man kann auch beides kombinieren. Dann wäre die Ampel sogar blindengerecht.
Der Sketch soll folgendes bewirken:
Bei Dunkelheit soll die Ampel (3-Farben-LED) rot leuchten, bei Dämmerung gelb und wenn es hell ist grün.

Installiere auch hier den Seriellen Monitor für den analogen Eingang A0. Durch Beobachten der Werte beim Verdunkeln des LDR kannst du die Übergänge (rot/gelb/grün) sinnvoll festlegen.

6. Einführung des Shift-Registers

Die seriell-parallel-Umsetzung ist eine ganz zentrale Aktion für die Programmierung. Nur so ist es möglich z.B. mit nur 3 Datenpins des Arduino beliebig viele Ziffern, später auch Grafiken, in ein Display zu



transportieren.
Öffne dazu die separate PDF: „74HC595 8 Bit Schieberegister mit Latch".

a) Der **shiftregister-sketch-1** befindet sich auf dem Stick. Teste ihn und nimm Veränderungen dran vor.

b) Erzeuge damit ein schnelles „Lauflicht". Hier sollten nur wenige zusammenhängende Bits aktiv sein, z.B. 10000000.

c) Erweitere den Sketch so, dass die Geschwindigkeit des Lauflichtes langsam zu- oder abnimmt.

d) Vereinfache den Sketch, so dass ein im Array vorgegebenes Bitmuster nach dem Hochladen des Sketches sofort von den LEDs angezeigt wird. Dazu muss man das Register nicht löschen und die Übergabe an das Latch muss nur noch erfolgen nachdem ..... überlege selbst!

7. Spielen mit dem Shift-Register und neue Methoden

Jetzt wird´s komfortabel: **shiftOut()** und andere Bequemlichkeiten. Lade den **shiftregister-sketch-2** und teste ihn.

a) Verändere ihn so, dass rückwärts gezählt wird.

b) Erweitere den Sketch so, dass im seriellen Monitor zusätzlich der Wert der Variablen **counter** erst als Binärzahl und ca. 5-10 Leerzeichen dahinter als Dezimalzahl angegeben wird. New Line nicht vergessen!
Binäre Anzeige: Serial.print(counter, BIN),
Dezimale Anzeige: Serial.print(counter)