1. Ultraschall-Sensor, Entfernungen messen

Mit unserem Ultraschall-Sensor (SR04) soll die Entfernung gemessen und auf dem TM1637 Modul angezeigt werden.
Den Sketch einer einfachen Realisierung besprechen wir.

Hier ein paar Erläuterungen: Wir verwenden eine neue Anweisung zum Auslesen der Echodauer in Micro-sekunden: Syntax **`pulseIn(pin, value)`** wobei **`pin`** die Nummer des Pins angibt, welcher die Zeit auslesen soll (z.B. „Echo") und **`value`** HIGH oder LOW sein kann. Im ersten Fall wird die Zeit in µs gemessen, in welcher der Pin auf HIGH liegt. Wir definieren „zeit" als „long"-Variable und setzen sie mit **`pulseIn`** gleich.

Mit Hilfe der Schallgeschwindigkeit können wir dann die Distanz d (Strecke) berechnen. Es ergab sich die Formel: distanz=zeit*0,01715, Zeit in µs und distanz in cm.

2. NewPing Library (Ultraschall-Messung)

Mit der **NewPing-Library** sollen die Sonarwerte etwas besser werden. Öffne diese Library und teste den einfachen Sketch zur Messung der Entfernung. Verwende ein Delay von ca. 500ms, stelle die MAX-DISTANCE auf 350 (cm) und teste, ob die Ergebnisse spürbar besser als bei 1.) sind.

3. Einparkhilfe programmieren

Mit Hilfe des Ultraschallsensors können wir eine realistische Einparkhilfe programmieren. Lediglich die Abstände wählen wir kleiner als in der Realität. Die folgende Aufgabe ist mit Hilfe des Sketches von 1) zu lösen. Beachte: **NewPing** lässt die Verwendung von **tone()** leider nicht zu.

a) Das TM1637-Display ist weiterhin mit folgenden Eigenschaften einzubinden:
* Für d>60cm soll die Anzeige mit 20% der maximalen Helligkeit leuchten.

** Für d<60cm soll die Helligkeit 100% betragen.

*** Für d<15cm soll die Anzeige zusätzlich blinken.

b) Danach: Wenn ein Gegenstand (z.B. geparktes Auto oder eine Wand) mehr als 60cm vom Sensor entfernt ist soll kein Ton entstehen. Ab einer Distanz von d<60cm soll ein Ton generiert werden, dessen Frequenz mit kleiner werdender Distanz immer höher wird (**map**

verwenden!). Ab einer Distanz von d<15cm soll der Ton in einen Signalton (=Warnung) übergehen oder du steuerst damit unser Sprachmodul ISD 1820 an, auf welchem du vorher eine geeignete Nachricht aufnimmst (z.B. „vorsicht, gleich krachts" oder eine Lego-Mindstorms Klangdatei).

Info: Nochmal: Was ist eine map-Funktion?
Mit Hilfe dieser Funktion können wir z.B. Sensorwerte so abändern, dass sie für unseren Zweck passen. Wenn z.B. ein Temperaturfühler Werte zwischen 0 und 1023 ausgibt und wir wissen, dass die Grenze 20 und 100 ist, kann map das für uns umrechnen. Syntax:

```
y = map(x, 0, 1023, 20, 100);
```

x ist die Variable, die transformiert werden soll.