1. Neue Methoden, Lauflichter und Ping-Pong

Lade vom Stick den **shiftregister-sketch-3**. Hier werden einzelne Bits gesetzt. Wir deuten den Sketch gemeinsam.

a) Erweitere den Sketch so, dass das Bitschieben erst von links nach rechts und danach von rechts nach links verläuft. Verändere die Laufgeschwindigkeit, der Effekt soll möglichst „wirkungsvoll" sein. Mit den Methoden des Sketch 3 **bitSet()** und **shiftOut()** ist das relativ einfach realisierbar.

b) Erweitere nun den Sketch so, dass synchron mit dem Lauflicht kurze Töne mit an- bzw. absteigender Frequenz, je nach Laufrichtung, zu hören sind.

2. Die erste größere Library (TM 1637)

Nun verwenden wir das Anzeige-Modul mit vier Stück 7-Segment Anzeigen. Das ist bereits so komplex, dass wir ohne Library kein Segment zum Leuchten bringen.

a) Öffne die Beispiel-Sketche, lade diese und „spiele" damit. Wenn du alles durchprobiert hast (das dauert eine Weile) kannst du (hoffentlich) folgende Aufgaben lösen:

a) Das Display soll dein Geburtsdatum (MM,JJ) anzeigen: Nacheinander 5 Sekunden hell, dann 5 Sekunden blinkend und schließlich 5 Sekunden dunkler. Danach soll das Datum (oder „HALLO") als Laufschrift erscheinen.

b) Der Sketch „Snake" soll mindestens 20 Runden alleine ablaufen, ohne Vorspann. Eventueller Zusatz: Davor soll eine Textansage per Sprachmodul zu hören sein, z.B. „the snake is running 20 times" oder halt was anderes.

3. Ultraschall-Sensor, Entfernungen messen

Mit unserem Ultraschall-Sensor (SR04) soll die Entfernung gemessen und zunächst auf dem Seriellen-Monitor 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 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.

Entfernungen auf dem Display anzeigen

Verwende nun das 4x7-Segment-Display um die Entfernung direkt anzuzeigen. Deaktiviere den SM. Sketch speichern!

4. 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 (zunächst mit der Anzeige nur auf dem seriellen Monitor).
Baue die 7-Segment-Anzeige in den Sketch ein, verwende ein Delay von ca. 500ms, stelle die MAX-DISTANCE auf 350 (cm) und teste, ob die Ergebnisse spürbar besser als bei 3.) sind. Den seriellen Monitor benötigen wir dann nicht mehr.

5. 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 3) 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).