

1. Neue Methoden, Lauflichter und Ping-Pong

Kopiere **Sketch 3** aus den Extraseiten zum Schieberegister. Hier werden einzelne Bits gesetzt. Wir werden den Sketch gemeinsam deuten.

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. Im Gegensatz von Aufg. 6e), Blatt 4, ist das mit den Methoden des Sketch 3 (bitSet, shiftOut) relativ einfach realisierbar.

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

c) Wer immer noch Zeit hat, kann zusätzlich die Laufgeschwindigkeit wieder so programmieren, dass auch sie zeitlich wächst oder fällt.

2. Brightness Control

Im **Sketch 4** findest du eine Möglichkeit, zusätzlich die Helligkeit des LEDs aus Aufg. 1 über PWM zu steuern.

a) Lade den Sketch auf den Arduino und teste ihn.

b) Schreibe den Sketch so um, dass er ohne ausgelagerte Methoden auskommt (testen!).

3. Das 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-Sketches, lade diese und „spiele“ damit. Wenn du alles durchprobiert hast (das dauert eine Weile) kannst du (hoffentlich) folgende Aufgaben lösen: 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. Davor soll eine Textansage per Sprachmodul zu hören sein, z.B. „the snake is running 20 times“ oder halt was anderes.

4. Würfelspiel programmieren

Wer Zeit hat, kann versuchen, das in den Beispielen (TM1637) vorhandene Würfelspiel so abzuändern, dass zwei wirkliche Spieler würfeln. Dazu benötigt man nur einen Taster. Im Sketch muss dann per Zufall nach dem Drücken des Tasters eine Zahl zwischen 1 und 6 erzeugt werden. Das Würfelspiel kann man auch mit Geräuschen noch etwas aufpäppeln!

5. 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. Versuche danach den Sketch selbst zu schreiben. Hier die Kurzanweisung:

- Definition der Variablen und Setup wie üblich.....
- Im Loop oder vor dem Setup die Zeitdauer des Echoimpulses und die Distanz mit **long** deklarieren
- Den Triggerimpuls (5µs LOW, 10µs HIGH) erzeugen
- Die Dauer des Echo's mit **pulseIn** auslesen (in µs)
- Die Länge der Distanz aus der Echodauer berechnen
- Diesen Wert im Seriellen-Monitor anzeigen und ein **delay** zwischen 100ms und 1000ms bis zur nächsten Messung einfügen

Entfernungen auf dem Display anzeigen

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

6. NewPing Library (Ultraschall-Messung)

Mit dem 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 **5.)** sind. Den seriellen Monitor benötigen wir dann nicht mehr.

7. 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 5.) zu lösen. Beachte: **NewPing** lässt die Verwendung von **tone()** leider nicht zu.

a) Das TM1637-Display ist weiterhin mit folgenden Eigenschaften einzubinden:

- Für $d > 60\text{cm}$ soll die Anzeige mit 10% der maximalen Helligkeit leuchten.
- Für $d < 60\text{cm}$ soll die Helligkeit 100% betragen.
- Für $d < 15\text{cm}$ soll die Anzeige zusätzlich blinken.

b) Zusätzlich: 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 $d < 60\text{cm}$ soll ein Ton generiert werden, dessen Frequenz mit kleiner werdender Distanz immer höher wird (**map** verwenden!). Ab einer Distanz von $d < 15\text{cm}$ soll der Ton in einen Signalton (=Warnung) übergehen.