

1. Laufflichter und neue Methoden

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.

Wer möchte, kann die Lauf-Geschwindigkeit wieder so programmieren, dass sie zeitlich wächst oder fällt.

b) Initialisiere wieder den Seriellen Monitor, der den Bitwert und den Dezimalwert der Variablen „leds“ anzeigen soll. Jetzt sollte das Programm wieder langsam laufen.

2. Brightness Control

Im **Sketch 4** findest du eine Möglichkeit, zusätzlich die Helligkeit des LEDs aus Aufg. 1 über PWM zu steuern. Wir haben den Ablauf erläutert. Dennoch: Versuche dir vorzustellen, diesen Sketch einem Freund, der „C“ halbwegs beherrscht, zu erklären. Schaffst du das? Falls ja, lasse den Sketch nun laufen, falls nein, frage nach!

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 alleine ablaufen, ohne Vorspann und mindestens 20 Runden. Davor soll eine Textansage per Booster-Box und 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 ausschmücken!

In Vorbereitung:

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 mit **long** deklarieren
- Den Triggerimpuls (5µs LOW, 10µs HIGH) erzeugen
- Die Dauer des Echo's mit **pulseln** 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.

6. Einparkhilfe programmieren

Mit Hilfe des Ultraschallsensors können wir eine realistische Einparkhilfe programmieren. Lediglich die Abstände wählen wir kleiner als in der Realität. Der zu erstellende Sketch soll folgendes bewirken: Wenn ein Gegenstand (z.B. geparktes Auto oder eine Wand) mehr als 60cm vom Sensor entfernt ist soll kein Ton entstehen. Ab einem Abstand von <60cm soll ein Signalton generiert werden, dessen Ton-Pause-Intervall mit kleiner werdender Distanz auch immer kleiner wird. Ab einem Abstand von <15cm soll der Ton in einen Dauerton übergehen. Verwende den „normalen“ Lautsprecher, nicht die Booster-Box!

7. Jetzt wird es bunt!

Man benötigt die 3-Farben-LED, den Shield-Button und einen Lautsprecher. Gesucht ist ein Sketch der folgendes bewirkt:

Nach jedem Drücken des Buttons leuchtet die LED in einer anderen Farbe in der Reihenfolge rot—grün—blau. Nach dem 4.-mal Drücken sollen die LED's aus sein. Bei jedem Druck auf den Button soll zusätzlich ein ultrakurzer Kontroll-Ton entstehen.

1. Erweiterung: Nun sollen auch Mischfarben auf Tastendruck entstehen, also z.B. rot + blau usw.

2. Erweiterung: Die Frequenz des kurzen Tones soll sich bei jeder Farbe ändern.