

## 1. LED & Key, großes Anzeigenmodul

Bevor wir das Library (TM1638) verwenden, wollen wir ein bisschen in die Ansteuerung des Modules reinschnuppern. Dazu verwenden wir **shiftOut** und müssen berücksichtigen, dass hier alles etwas komplizierter ist. Das erste Byte ist ein **Command-Byte**, welches z.B. den Helligkeitswert der Dateneingabe setzt. Danach kommen die Daten, für jede Anzeige mit immer zwei Bytes, im ersten werden die Segmente angesteuert (letzte 7 Bit) und das erste Bit setzt den Dezimalpunkt. Im zweiten Byte wird u.a. die LED oben mit Hilfe des niederwertigsten Bit gesetzt.

**a)** Öffne den Sketch **TM1638-1** (auf dem Stick). Dieser ist ausführlich dokumentiert. Mit jedem Druck auf die Reset-Taste wird die nächste Ziffer gesetzt. Lösche zunächst das Display. Teste die Eingabe für einige Ziffern, so dass auf dem Display z.B. die Zahl 6666 steht, danach soll 23.412 erscheinen und 5 LEDs oben sollen leuchten.

**b)** Aktiviere die for-Schleife und erkläre das Verhalten. Gib Dezimalpunkte ein und (de-)aktiviere die LEDs.

**c) Für Profis:** Verwende das **FONT-Array** für die Anzeige der Ziffern 1 bis 9 und lade diese mit Hilfe einer Schleife. Erweiterung: Setze dabei einen Dezimalpunkt und evtl. jede 2. LED.

## 2. TM1638 Master-Library

Diese Library ist schon ein Hammer und teilweise für Anfänger schwer verständlich. Der Erzeuger wollte wohl zeigen, was er drauf hat. Die Erklärungen sind recht dürftig.

**a)** Teste ein paar Beispiel-Sketches aus der Library. Vorsicht: Im **functions\_example** Sketch werden die Pins mit 3, 2, 5 deklariert. Ändere das auf 8, 9, 7 ab, sonst müsstest du die Kabel vertauschen.

**b)** Öffne den Sketch **TM1638-zaehler** (siehe Stick) lasse ihn laufen und versuche diesen zu verstehen. Danach solltest du die berechnete Distanz des Ultraschall-Sketches auf dem Display anzeigen können. Dazu ist der NewPing-Sketch geeignet zu erweitern.

## 3. Taster verwenden, Keyboard programmieren

Die 8 Taster auf dem Modul **LED & KEY** lassen sich universell verwenden. Lade dazu folgenden Sketch, der selbsterklärend ist und beobachte den SM. Die Variable „**keys**“ lässt sich nun beliebig in jedem Sketch verwenden (TM1638 muss dazu natürlich inkludiert sein). Erläutere zunächst den Sketch:

```
#include <TM1638.h>
TM1638 module(8, 9, 7);

void setup() {
  Serial.begin(9600);
}

void loop() {
  byte keys = module.getButtons();
  Serial.println(keys);
  delay(300);
}
```

**a)** Erweitere den Sketch, so dass ein 300ms langer Ton nach dem Drücken des 4. Tasters entsteht.

**b)** Hier gibt es Schokolade zu gewinnen!! Wenn du nun noch **<pitches.h>** inkludierst kannst du die 8 Taster so programmieren, dass sie einem Ausschnitt einer (Klavier-)Tastatur entsprechen. Dann lassen sich darauf richtige Melodien spielen! Das Abfragen der Variablen **keys** könnten man mit einer if-Abfrage gestalten oder einfacher mit **switch/case statements** Beispiel:

```
switch (var) {
  case 1: //do something when var equals 1
    break;
  case 2: //do something when var equals 2
    break;
  default: // if nothing else matches, do the default
    // default is optional
    break;
}
```

**c) LEDs aktivieren:** Das Drücken der Taster lässt sich mit folgender Anweisung optisch kontrollieren:

`module.setLEDs(keys);`

Mit `module.setLEDs(128);` würde man z.B. die letzte der 8 LEDs bedienen. Was würde geschehen, wenn man 128 durch 126 ersetzt? Ausprobieren!

Kleiner Test: Ersetze in `setLEDs „keys“` durch „2\*keys“.

Überlege vorher, was dann passiert.

**Nur für Profis:** Wenn man den Taster loslässt, geht die LED natürlich wieder aus. Ändere den Sketch so, dass eine einmal gesetzte LED an bleibt.

## 4. Laufschrift und mehr

In zwei weiteren Sketches wird eine Laufschrift initiiert sowie die kombinierte Darstellung von Text und Zahl-Variable gezeigt. Diese Sketches verwenden wir im Sinne einer selbst erstellten Library. Damit haben wir die wichtigsten Möglichkeiten des TM1638-Moduls ausgeschöpft.