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 0 bis 9 und lade diese mit Hilfe einer Schleife. Erweiterung: Setze dabei einen Dezimalpunkt und evtl. jede 2. LED.

d) Nur für super Profis: Die Eingabe der anzuzeigenden Zahl kann man weiter vereinfachen. Das Löschen der Anzeige sollte kein großes Problem sein. Die Eingabe der Ziffer, des Dezimalpunktes und der LED könntest du mit einer Methode z.B. **void printZi(z, DP, LED);** gestalten. **z** wäre jeweils eine Dezimalziffer, die ein passendes Byte aus dem FONT-Array auswählt. **DP** (Dezimalpunkt) und **LED** sind **bool-Variable** (**0** oder **1** bzw. **false** oder **true**). Mit **bitSet** kannst du in die erste Stelle des ersten Daten-bytes eine „1" setzen, wenn dort der DP erscheinen soll. Ähnliches gilt für LED im 2. Datenbyte.

2. TM1638 Master-Library

Diese Library ist schon ein Hammer und 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, 4 deklariert. Ändere das auf 8, 9, 7 ab, sonst müsstest du die Kabel anders anschließen.

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

3. Taster verwenden, Keyboard programmieren

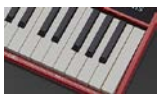
Die 8 Taster auf dem Modul LED & KEY lassen sich universell verwenden. Lade dazu folgenden Sketch, der selbsterklärend ist und beobachte den SM. Die Variable „keys" lässt sich nun beliebig in jedem Sketch verwenden (TM1638 muss dazu natürlich inkludiert sein). Erläutere zunächst den Sketch (DIO=Daten In-/Output, CLK=Clock STB=Strobe):

```
#include <TM1638.h>
TM1638 module(8, 9, 7); //8= DIO, 9= CLK, 7= STB

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

void loop() {
byte keys = module.getButtons();
Serial.print(keys);
Serial.print("……binaer:");
Serial.println(keys, BIN);
delay(300);
}
```

Hier erhältst du eine dezimale und binäre Ausgabe auf dem SM. Drücke auch mehrere Tasten gleichzeitig und beobachte die Ausgabe

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! Die beste Melodie wird prämiert! Das Abfragen der Variablen keys könnte man mit einer if-Abfrage gestalten oder einfacher mit **switch-case statements** Beispiel:

```
switch (var) {
  case 1:   //mache irgend etwas, wenn die Variable den Wert 1 hat
    break;
  case 2:   // mache etwas anderes, wenn der Wert 2 ist
    break;
  default:  // wenn nichts zutrifft, mache das was hier steht
    break;   // default ist optional, jedoch sinnvoll!
  }
```

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

module.setLEDs(keys);

Mit module.setLEDs(128); würde man z.B. die letzte der 8 LEDs bedienen. Überlege vorher, was passiert wenn man

- 128 durch 68 ersetzt?
- in set.LEDs „keys" durch „2*keys" ersetzt?

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.

Fortsetzung auf Blatt 8!