

1. Laufschrift und andere Tricks

Wir sind zunächst immer noch beim Modul **LED & KEY**. In zwei weiteren Sketches wird eine Laufschrift initiiert sowie die kombinierte Darstellung von Text und Zahl-Variable gezeigt. Diese Sketche verwenden wir als fertige Programme.

Damit haben wir die wichtigsten Anwendungs-Möglichkeiten des TM1638-Moduls ausgeschöpft.

Kurzinfo zu den Sketchen:

Laufschrift: Der Text kann im Prinzip beliebig lang sein. Die Blank's vor und hinter dem Text löschen die Anzeigen. Das wirkt besser, muss aber nicht sein.

Text + Variable: Das Display kann eigentlich entweder nur Text oder nur Zahlen anzeigen. Um beides gleichzeitig anzuzeigen muss die Variable (Zahlen) in formatierten Text umgewandelt werden. **SECS** steht hier als Beispiel für 4 Textzeichen. **%03d** ist eine Formatierungsanweisung für die Variable. Die **0** besagt, dass führende Nullen angezeigt werden, **3** steht für 3 Ziffern, **d** für dezimale Darstellung. **millis()** ist eine innere Zeitbasis des Arduino, die mit dem Start die Zeit in Millisekunden angibt. Diese wird mit **Reset** wieder auf Null gesetzt. Sie dient hier als „Beispielvariable“.

Vorbereitung zum Roboter Aufbau

Da die Motoren nicht absolut gleich laufen, müssen wir den Bewegungszustand feststellen können. Dazu dient ein **Encoder**. Was ist das?

Zwischeninfo, Begriffe klären:

Code (codieren): Verfahren, welches die Symbole einer Nachricht in eine andere Form bringt ohne den Informationsgehalt einzuschränken. Bsp. Binärcode, ASCII-Code usw.

Decoder: Gerät zur Entschlüsselung codierter Signale.

Encoder: Ist ein Sensor für Drehwinkel, der in der Regel digitale Ausgangssignale liefert, die am anderen Ende der Sensorleitung im Auswertegerät decodiert werden müssen. Das macht hier unser Arduino.

2. Der Rotary-Encoder



Nach diesem Prinzip funktioniert auch das Scroll-Rad einer Maus. Abhängig von der Raddrehung werden Impulse abgegeben aus denen man die Drehrichtung und den Drehwinkel bestimmen kann. Siehe Extrablatt.

a) Verbinde den Encoder mit der Betriebsspannung und die Ausgänge **CLK** und **DT** jeweils mit einer LED (z.B. 3-Farben-LED). Beobachte ob bei ganz langsamen Drehen eine Zeitverzögerung (z.B. blau-rot) zu erkennen ist.

b) Lade den Sketch **Rotary-Encoder** (ist auf dem Stick) und teste ihn. Hier wird ein Interrupt verwendet!

Info: Was sind Interrupts ?

Durch die Verwendung eines Interrupts wird der gerade laufende Code buchstäblich sofort unterbrochen, egal ob er eine Schleife abarbeitet oder sonst was tut.

Direkt danach wird der normale Programmablauf an genau der Stelle fortgesetzt an der er unterbrochen wurde.

Interrupts müssen in der Regel aktiviert werden. Syntax:

attachInterrupt(INTER, ISR, MODUS)

INTER: Nummer des Interrupts (z.B. 0=Pin2).

ISR: Die Interrupt **S**ervice **R**outine ist eine beliebige Funktion die ausgeführt werden soll, sobald ein Ereignis eintritt. Diese Funktion benötigt keinen Rückgabewert. In dieser Funktion werden weder `delay()` noch `millis()` funktionieren.

MODUS: Der Modus definiert, wann der Alarm oder das Ereignis ausgelöst werden soll:

LOW: Löst aus wenn der Wert LOW ist

CHANGE: Löst aus wenn sich der Wert am Pin ändert

RISING: Löst aus wenn der Wert des Pins von LOW auf HIGH wechselt

FALLING: Löst aus wenn der Wert des Pins von HIGH auf LOW wechselt

Hier eine Liste der externen Interrupts unserer Arduino Boards:

Uno

- INTER: 0 = Digitaler Pin 2
- INTER: 1 = Digitaler Pin 3

Mega2560 wie Uno und zusätzlich

- INTER: 2 = Digitaler Pin 21
- INTER: 3 = Digitaler Pin 20
- INTER: 4 = Digitaler Pin 19
- INTER: 5 = Digitaler Pin 18

Weitere Info: Grundsätzlich gilt: **volatile** (flüchtig) wird nur benutzt, wenn man eine globale Variable benötigt, die in einer Interrupt Routine verändert werden soll.

c) Erweitere den Sketch so, dass die Zahl auf dem Display des kleinen TM1637- oder des großen **LED & KEY** Moduls angezeigt wird [`display.clear()` usw. nicht vergessen.....]. Mit Hilfe des Buttons (SW) soll die Anzeige wieder zurück gesetzt werden. Für Profis: Integriere auch Text und (oder) eine Laufschrift zu Beginn.

d) Nur Für Profis: Gestalte den Sketch so, dass beim Drehen im Seriellen Monitor nacheinander das Alphabet (nur Großbuchstaben) angezeigt wird. Zum Kopieren: `caracterFont[26] = {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}`

Für hyper-Profis: Erweitere den Sketch so, dass der eingestellte Buchstabe nach dem Drücken des Buttons z.B. in einem Array gespeichert wird. Mit dem Seriellen Monitor kann man das Array auslesen und so ganze Wörter oder Sätze bilden. Diese Methode wird auch in der Praxis angewendet!