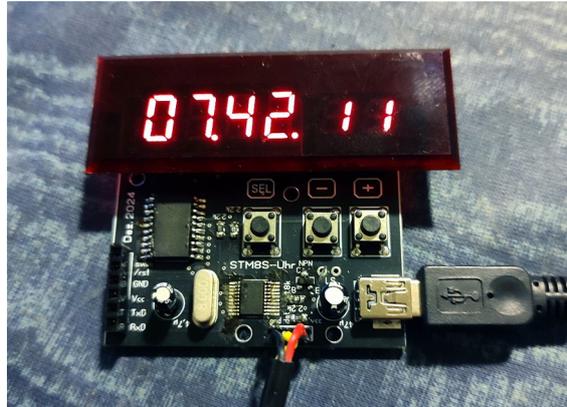


# Digitaluhr

mit STM8Sxx3, TM1637, DS18B20 und 6-Digit 7-Segmentanzeige  
R. Seelig



Die hier vorgestellte Uhr ist aus einer Diskussion im Forum [www.mikrocontroller.net](http://www.mikrocontroller.net) entstanden innerhalb dieser eine Uhr mit 7-Segmentanzeigen und hoher Ganggenauigkeit nachgefragt wurde. Da es unzählige Selbstbauuhren bereits im Netz gibt, stellt diese Uhr die gefühlte einmillionste Version dar.

Dennoch wartet diese Uhr mit ein paar wie ich finde Bemerkenswerten Dingen auf:

- Datumsfunktion
- Temperaturmessung mittels DS18B20
- Helligkeitsregulierung der Anzeige
- Umschaltbarer Anzeigenmodus von permanenter Uhrzeit und wechselnder Anzeige von Temperatur, Datum und Uhrzeit
- Kalibriermöglichkeit bei grob ungenauen Quarzen

Bei dieser Uhr wurde großen Wert auf gute Bedienbarkeit bei absolut geringstmöglichen Bauteilekosten gelegt. Aus diesem Grund viel die Wahl des verwendeten Mikrocontrollers auf einen STM8S003 (ein STM8S103 funktioniert auch) sowie einem TM1637 Anzeigetreiber. Bei [www.lcsc.com](http://www.lcsc.com) kostet der Controller (Stand Januar 2025) 29 Cent, der Treiber 15 Cent.

## Bedienung

Die Bedienung der Uhr erfolgt mittels den Tasten:

- SEL
- -
- +

Mittels der SEL-Taste gelangt man in das Uhrenmenu. Nach betätigen der Taste erlischt die aktuelle Anzeige und es erscheint der erste Menüpunkt. Folgende Menüpunkte stehen zur Verfügung:

- Set
- Mod
- CAL
- HELL

Mit den Tasten "+" und "-" können die einzelnen Menüpunkte angewählt werden. Ein Betätigen der SEL-Taste wählt den entsprechenden Menüpunkt aus.

### SEt

Hier kann die Uhr eingestellt werden. Nacheinander wird das Jahr, der Monat, der Tag, die Stunde, die Minute und die Sekunde abgefragt. Jeder einzelne Wert kann mittels den beiden Tasten "+" und "-" verändert werden. Die Tastenabfrage geschieht dynamisch, d.h. ein längeres Drücken läßt die entsprechende Eingabe schneller inkrementieren bzw. dekrementieren. Jede Eingabe ist mit der SEL-Taste zu bestätigen.

### Mod

"Mod" stellt den Anzeigemodus ein. Der Wert "0" repräsentiert hier eine permanente Uhrzeitanzeige, der Wert "1" stellt den wechselnden Anzeigemodus ein. In diesem Modus werden in regelmäßigen Abständen der Temperaturwert des DS18B20 und anschließend das Datum angezeigt. Die Eingabe ist mit der SEL-Taste zu bestätigen.

### CAL

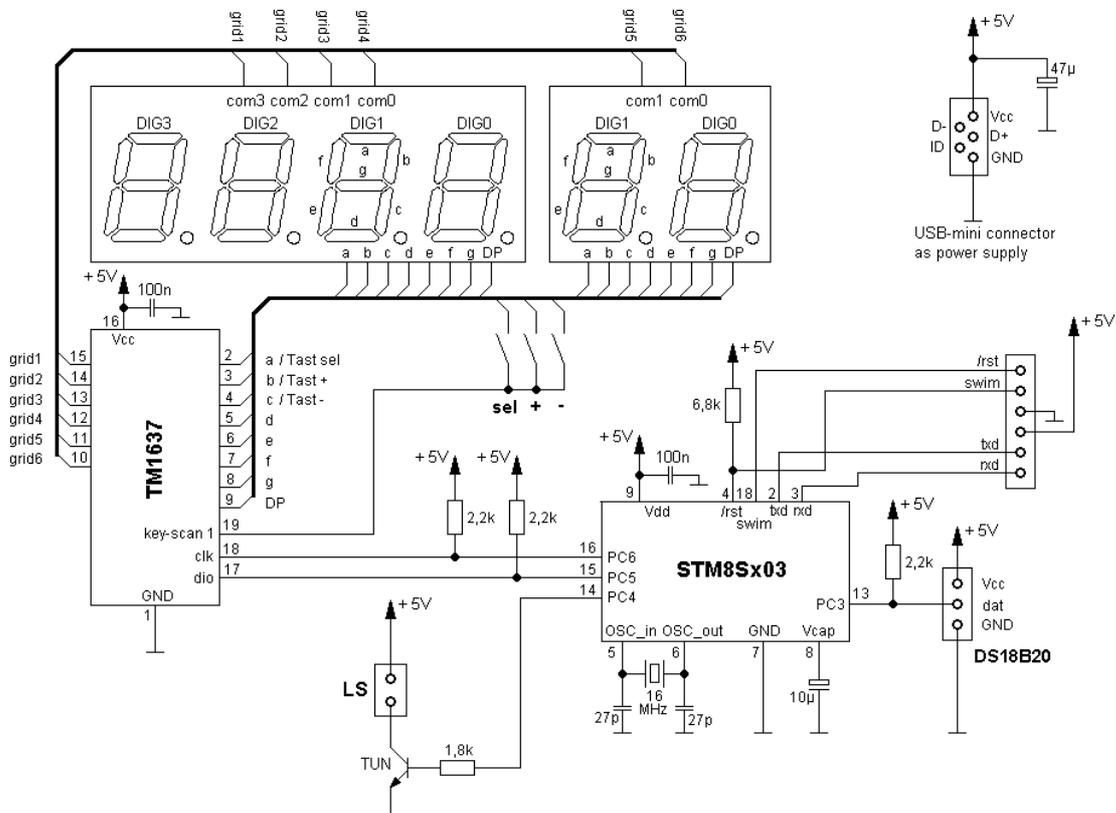
"CAL" stellt eine Kalibriermöglichkeit für grob falsch gehende Quarze ein. Wird hier ein Wert größer als 0 eingegeben, so wird an jedem Tag um 00:01.30 die Uhr um diesen Betrag korrigiert. Die Eingabe ist mit der SEL-Taste zu bestätigen.

Diese Funktion ist entstanden, da ein erster Entwurf (mit einem ATmega auf Arduino-Board) Gangungenauigkeiten von ca. 3 Sekunden am Tag hatte. Wie sich herausgestellt hat, ist der vom Autor verwendete Quarz deutlich besser, so dass die Uhr deutlich weniger als 1 Sekunde am Tag abweicht. Hier besteht somit ein Bedarf der Nacharbeit um eine Kalibriermöglichkeit zu realisieren, die bspw. 1 Sekunde in 5 Tagen korrigiert.

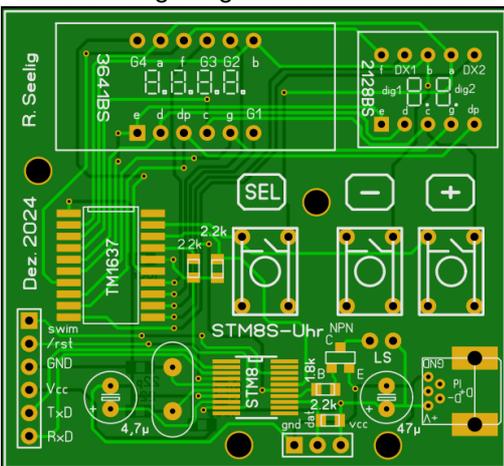
### HELL

"HELL" stellt die Helligkeit der Anzeige ein. Eingabemöglichkeiten sind hier von 0..7. Die Eingabe ist mit der SEL-Taste zu bestätigen.

## STM8Sx003 : 7-Segment Digitaluhr mit TM1637 Treiber und DS18B20 Temperatursensorauswertung



Die Schaltung ist ein einfacher Mikrocontrolleraufbau. Der Mikrocontroller steuert über 2 Leitungen, Clock- und Datenleitung, den Anzeigentreiber, an den die 7-Segmentanzeigen mit gemeinsamer Anode angeschlossen sind. Diese besteht aus 2 unterschiedlich großen Anzeigen, bei denen die Sekunden auf der kleineren angezeigt werden. Im Entwurf wurden 3641BS für die 4-stellige Anzeige, 2821BS für die 2-stellige Anzeige verwendet. Für einen eventuellen Lautsprecheranschluss kann ein universeller NPN-Transistor verwendet werden (die Software allerdings unterstützt diesen Anschluss noch nicht und der geneigte „Bastler“ kann evtl. die Uhr in der Software um eine Weckfunktion erweitern).



Die Uhr wird über eine Mini-USB Buchse mit Spannung versorgt, eine Spannungsregelung erfolgt nicht, so dass das Netzteil (evtl. altes Handy-Ladegerät) diese vornehmen muß.

Eine Buchsenleiste enthält die Signale für /rst, swim, UART-Transmit und UART-Receive. Über diesen Anschluss kann der Controller mit seiner Firmware geflasht werden. Als Programmiergerät kann ein preiswerter STLINK-Clone verwendet werden.

Desweiteren ist es möglich, den Controller mit einem Bootloader wie bspw. aus <https://www.mikrocontroller.net/topic/425431#5745227> zu flashen.

Die Gerberfiles der Platine sind im Ordner /uhr\_tm1637/docs/stm8\_uhr\_gerber.zip zu finden.

## Programm der Uhr

Das Main-Programm der Uhr benötigt 2 weitere Softwaresourcen, my\_time.c / my\_time.h für Uhrzeit und Datumsberechnungen, sowie tm16xx.c / tm16xx.h für das Ansteuern des 7-Segment Anzeigetreibers.

Die Programmierung der Uhr (bzw. des „Uhrwerks“) geht **nicht** den klassischen Weg über das Hochzählen von Variablen für Sekunden, Minuten und Stunden innerhalb eines Interrupts (bei Überlauf der Sekunden von 59 nach 60 die Minuten erhöhen und bei Überlaufen der Minuten dann die Stunden).

Bei dieser Uhr wird in einem Interrupt lediglich eine 32-Bit Variable kontinuierlich hochgezählt. Der Inhalt dieser Variable repräsentiert die Anzahl der Sekunden, die seit einem gegebenen Referenzdatum, hier der 01.01.1970 um 00.00:00, vergangen sind.

Der Überlauf dieser Variable wird am 06.02.2106 um 6.28:15 Uhr erfolgen :-)

Dieses macht die Programmteile innerhalb des Interruptvektors sehr schlank und vereinfacht eventuelle Rechnungen mit Uhrzeiten und Datum (für bspw. einen Wecker) stark. Zudem ist es so auch möglich, mit dieser Uhr zusätzlich auch das Datum anzuzeigen.

Das Programm bedient sich hierbei einer Struktur aus „my\_time.h“

```
struct MYDATE_TIME
{
    uint16_t year;
    uint8_t  month;
    uint8_t  day;
    uint8_t  hour;
    uint8_t  min;
    uint8_t  sec;
};
```

sowie einer globalen 32-Bit Variablen

```
volatile uint32_t sekcount;
```

die den Zeitstempel relativ zum Datum des 01.01.1970, 00:00.00 enthält.

Aufgrund der Tatsache, dass mit eine erste Uhr (anderer Controller und andere Takterzeugung) eine relativ hohe Gangungenauigkeit von knapp mehr als 3 Sekunden am Tag aufgetreten ist, wurde dem Programm eine Kalibriermöglichkeit hinzugefügt, mittels derer alle 24 Stunden um 00:00.30 die Uhr um die Anzahl der an einem Tag falschen Anzahl Sekunden korrigiert werden kann.

Bei dem Quarz des Muster der hier verwendeten Uhr traten jedoch Gangungenauigkeiten von weit kleiner als 1 Sekunde am Tag auf, so dass über einen anderen Kalibrieralgorithmus nachgedacht werden sollte.

Die Uhr selbst kann mit einem 16 MHz oder mit einem 8 MHz Quarz betrieben werden. Hierfür ist in

```
#undef F_CPU
#define F_CPU 8000000UL
```

der korrekte Quarzwert anzugeben.

Die Funktionen der Source für den Anzeigetreiber sowie der Source für „my\_time“ können in den Textdateien im Ordner /uhr\_tm1637/docs/ nachgelesen werden.

Ein Absetzen des „make“ Befehls übersetzt das Uhrenprogramm (ein im System installierter SDCC Compiler wird hier vorausgesetzt), ein „make flash“ programmiert den Chip bei angeschlossenem Programmer. Das Makefile selbst ist „selbsterklärend“ und kann, besonders in Bezug auf den verwendeten Programmer, entsprechend abgeändert werden.