

Lichtuhr-Sketch

Marco Steffens

24. März 2016

Inhaltsverzeichnis

1	Der Sketch	1
1.1	Änderung der Zeichen-Darstellung	5
1.2	Motor-Poti	5

1 Der Sketch

Die Fläche für die Anzeige, also das *Display*, ist im Sketch ein zweidimensionales Array. Die einzelnen Zellen sind also etwa wie bei einer Tabelle angeordnet:

[illegible]

Die ZEILENzahl entspricht der Anzahl der übereinander angeordneten LEDs.

Die SPALTENZahl ergibt sich aus der Darstellung der einzelnen Ziffern. Dazu später mehr.

Die anzuzeigende Uhrzeit („12:34 Uhr“) besteht aus einer zweistelligen Stunden- (H1, H2) und einer zweistelligen Minutenzahl (M1, M2), außerdem gibt es den Doppelpunkt als Trennzeichen zwischen Stunden und Minuten. Und die jeweils zwei Stellen von Stunden und Minuten werden durch einen kleinen Leerraum getrennt. Insgesamt ergibt sich daraus folgende Darstellung:

(H1) (Leerraum) (H2) (Trennzeichen) (M1) (Leerraum) (M2)

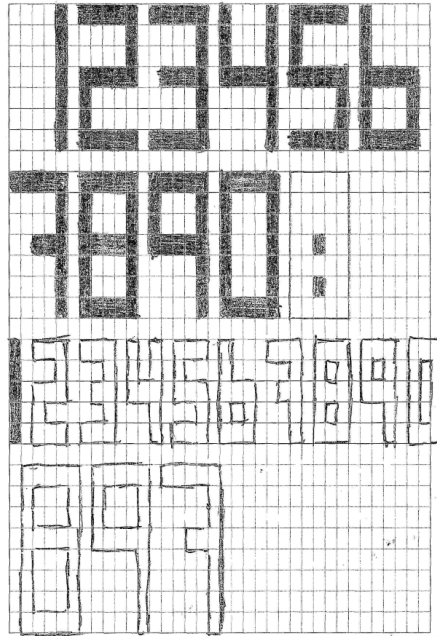
oder eben für "12:34 Uhr":

(1) () (2) (:) (3) () (4)

Die einzelnen Ziffern (H1, H2, M1, M2) liegen natürlich im Bereich 0 bis 9.

Die Ziffern werden im Sketch als *DIGIT* bezeichnet, der Leerraum als *SPACE* und das Trennzeichen als *DELIMITER*.

Diese drei *Zeichenarten* sind ebenfalls jeweils als zweidimensionales Array definiert. Die *HÖHE* dieser Zeichen entspricht dabei eigentlich wieder der Anzahl der LEDs, bzw. sie ergibt sich aus der gewählten Darstellung. Man kann sich das leicht mit kariertem (oder noch besser rautiertem) Papier ausmalen.



Durch die gestauchte Darstellung der Zahlen auf dem Display wären Zahlen mit einer Höhe von 8 sehr verformt gewesen, außerdem ist die Darstellung solcher Zahlen mit einer ungeraden Anzahl LEDs übereinander einfach symmetrischer... Deshalb werden im Sketch nur 7 LEDs für die Anzeige verwendet, obwohl 8 im Prototyp vorhanden sind. Die achte Zeile taucht im Code auf, ist aber immer leer, bleibt also unbelichtet. Die SPALTENZAHL ergibt sich dann auch einfach aus den gewählten Proportionen.

Beim Testen verschiedener Darstellungen hat sich gezeigt, dass eine eher *breite* Zeichenform (5 breit statt 3 oder 4) mit großen Leerräumen eine bessere Darstellung ergibt.

Die Anzeigenbreite ist ja fix, durch die breitere Form wird also nur die *Strichstärke* der Ziffern schmaler. Der Sketch ist aber so geschrieben, dass sich das verhältnismäßig einfach ändern lässt, was beispielsweise durch einen längeren Poti oder durch sonstwie geänderter Hardware notwendig sein könnte. Erstmal sind die DIGITs aber jedenfalls alle 5 Spalten breit. Das Array für die Zeichen enthält an den Stellen, an denen die LED angeschaltet werden soll, eine 1, sonst eine 0. Die Ziffer 8 beispielsweise sieht also wie folgt aus:

```
{ 1, 1, 1, 1, 1 }
{ 1, 0, 0, 0, 1 }
{ 1, 0, 0, 0, 1 }
{ 1, 1, 1, 1, 1 }
{ 1, 0, 0, 0, 1 }
{ 1, 0, 0, 0, 1 }
{ 1, 1, 1, 1, 1 }
{ 0, 0, 0, 0, 0 }
```

Das Leerzeichen ist drei *Spalten* breit und enthält überall eine Null:

```
{ 0, 0, 0 }
{ 0, 0, 0 }
{ 0, 0, 0 }
{ 0, 0, 0 }
{ 0, 0, 0 }
{ 0, 0, 0 }
{ 0, 0, 0 }
{ 0, 0, 0 }
```

Das Trennzeichen ist 5 *Spalten* breit:

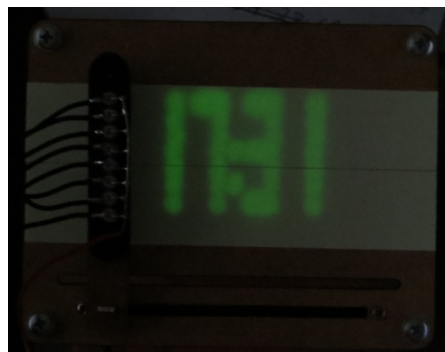
```
{ 0, 0, 0, 0, 0 }
{ 0, 0, 0, 0, 0 }
{ 0, 0, 0, 0, 0 }
{ 0, 0, 1, 0, 0 }
{ 0, 0, 0, 0, 0 }
{ 0, 0, 1, 0, 0 }
{ 0, 0, 0, 0, 0 }
{ 0, 0, 0, 0, 0 }
```

Somit ergibt sich für die Breite aller anzuzeigenden Zeichen eine Spaltenzahl von 31. Nämlich:

$$(1)()(2)(:)(3)()(4) \\ 5+ 3+ 5+ 5+ 5+ 3+ 5 = 31$$

[illegible][illegible]

Und wenn das schnell hintereinander erfolgt, und der LED-Arm einigermaßen langsam bewegt wird, dann ergibt sich die Anzeige der Uhrzeit.



1.1 Änderung der Zeichen-Darstellung

Ich habe mit Zeichen rumprobiert, die immer 7 Zeilen hoch waren. Die Ziffern haben ich aber in der Breite 3, 4 und 5 ausprobiert. Für den Prototypen hat sich eine Breite von 5 als diejenige herausgestellt, bei der die Anzeige am saubersten ist. Aber das wird sich mit anderer Hardware vermutlich ändern. Der Sketch ist so aufgebaut, dass sich die Zeichen auch relativ einfach anpassen lassen. Allerdings nur *relativ* einfach...

Wichtig dazu ist, dass sich aus den 1024 Poti-Schritten minus dem doppelten Offset und geteilt durch die Anzahl der „Spalten“ im Uhrzeit-Array ein ganzzahliger Wert ohne Rest ergibt. Und dann müssen im Sketch noch die entsprechenden Konstanten angepasst werden:

```
byte arrDisplay[8][31];

const int DIGIT_WIDTH = 5;
const int DIGIT_HEIGHT = 8;
const int DISPLAY_WIDTH = 31;
```

Außerdem müssen natürlich die Zeichen-Arrays geändert werden.

Das *Display*, also die beleuchtete Fläche, ist beim Prototypen verhältnismäßig klein. Die einzelnen LEDs haben einen Durchmesser von etwa 4mm, der LED-Arm ist mit 8 LEDs also 3,2cm lang. Die Breite zwischen dem LED-Arm in Anfangs- bzw. Endposition beträgt etwa 4,8cm, das wären dann 12 LEDs nebeneinander. Man könnte also sagen, dass die Anzeigefläche etwa 8x12 LEDs groß ist. Wobei man in der Breite natürlich stauchen und so mehr Platz schaffen kann. Trotzdem, wenn man das mal auf kariertem oder rautiertem Papier aufmalt, ergibt sich relativ wenig Platz und deshalb halt auch sehr schmale, gestauchte Ziffern. Durch eine entsprechende *Linse* vor den LEDs könnte man da noch was rausholen – ich würde einfach ein Gitter mit rechteckigen bzw. ovalen Öffnungen nehmen. (Deshalb das rautierte Papier statt kariertem.)

1.2 Motor-Poti

Das Motor-Poti wird beim Setup auf die Anfangsposition verschoben. Anschließend fährt das Poti etwa alle 5 Sekunden einmal hin und her. Die Zeit ist natürlich änderbar. Während das Poti fährt, läuft die `loop()`-Funktion durch, im Ruhezustand wird einmal pro Sekunde geschaut, ob schon 5 Sekunden um sind. *Das ist aber alles noch beta-Stadium.*