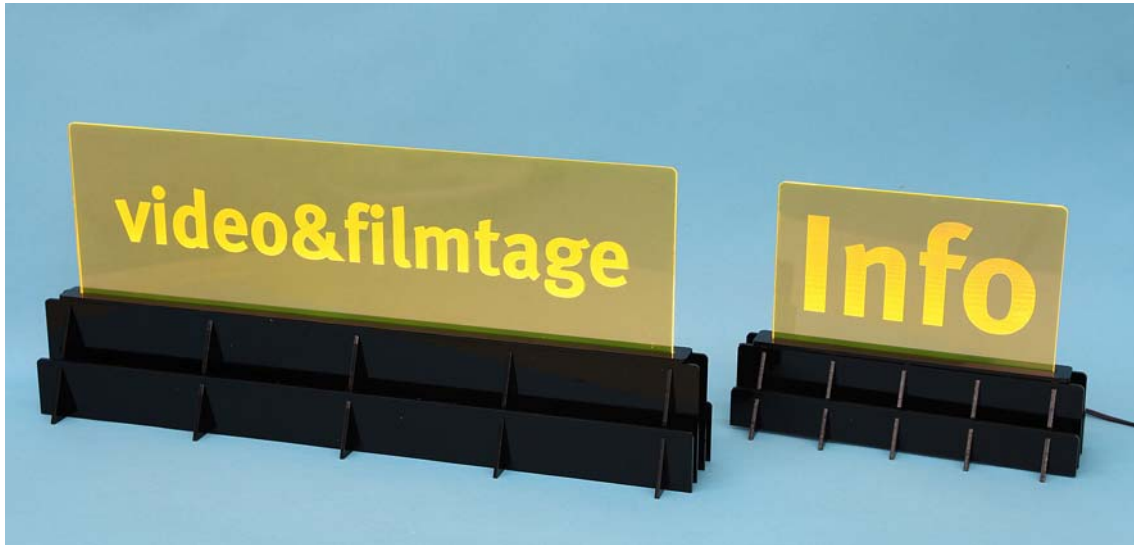


Leuchtschilder aus Plexiglas



Ein **Leuchtschild** besteht aus einem Sockel (Keilstumpf) und einer beleuchtbaren Schriftplatte aus Plexiglas mit eingraviertem Schriftzug (oder Logo). Die Teile des Leuchtschilds können am Lasercutter geschnitten bzw. graviert werden. Der Sockel besteht aus undurchsichtigem Plexiglas, die Schriftplatte aus fluoreszierendem Plexiglas.



Verwendete Materialien

Sockel: Plexiglas, GS 3mm Schwarz 9H01

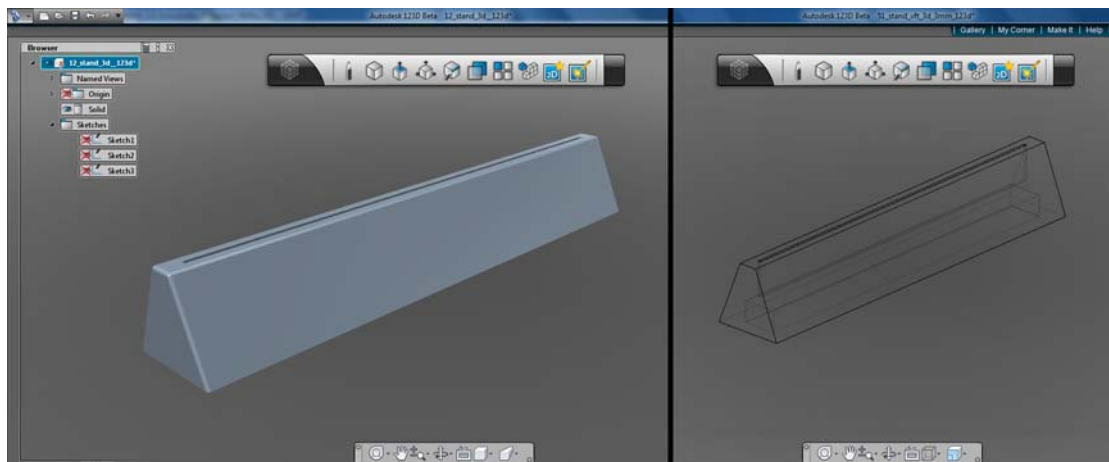
Schriftplatte: Plexiglas, GS 3mm fluoreszierend Orange 2C01

Im konkreten Fall habe ich 2 Versionen hergestellt, eine kleinere (Breite des Schriftplatte ca. 20cm) und eine größere Version (Breite ca. 45cm).

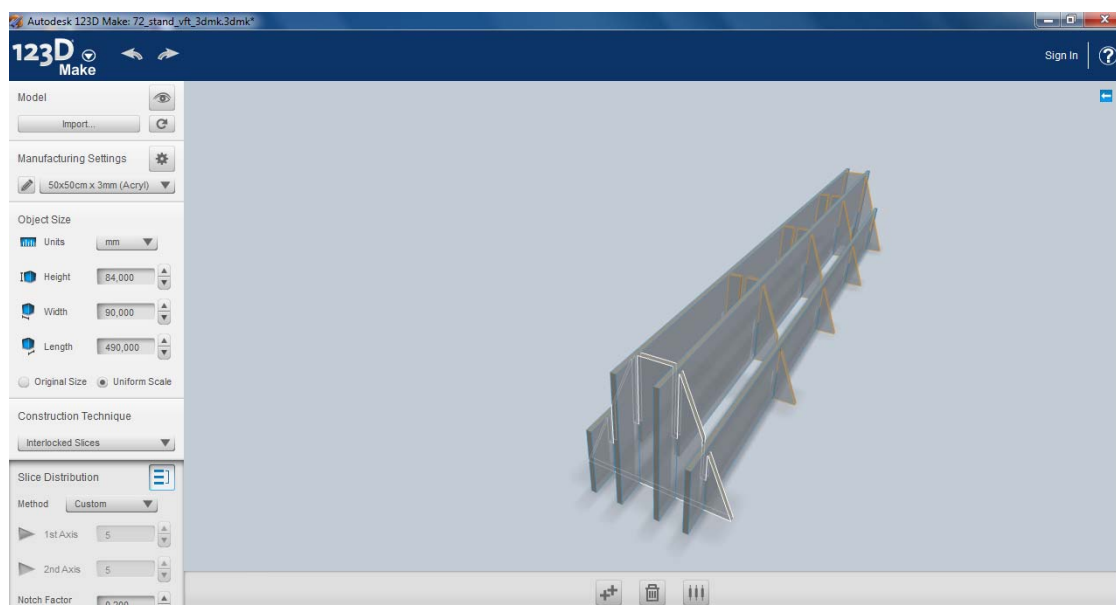
Der Sockel

Die Grundform des Sockels wurde mit **Autodesk 123D Beta** konstruiert und im Format STL gespeichert/exportiert (dazu muss man eingeloggt sein).

[Als Beispiel siehe Datei: 70_leuchtschild_sockel_gross__123d.123d]



Die weitere Verarbeitung erfolgte mit **Autodesk 123D Make**:

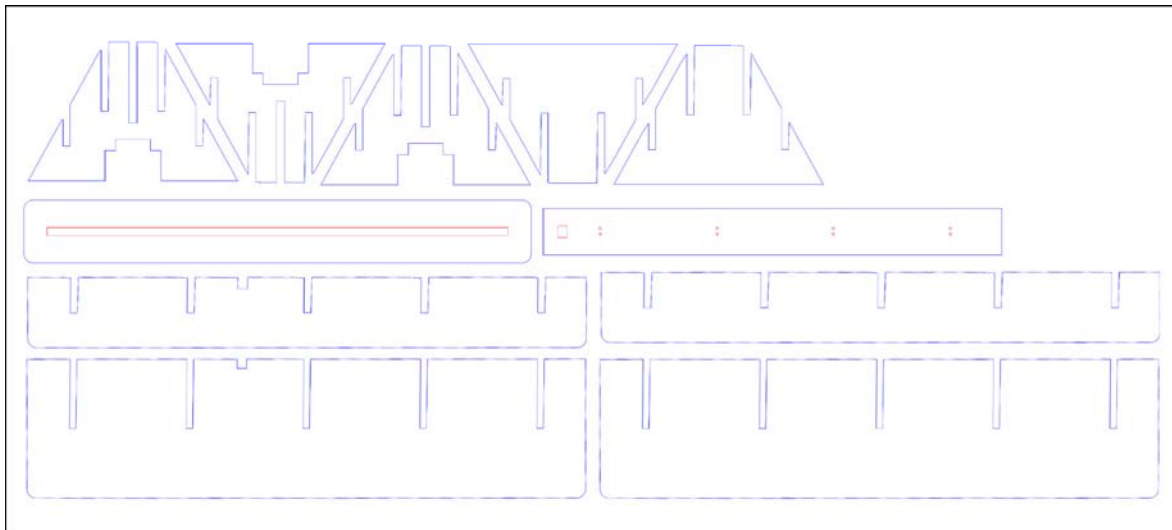


[Als Beispiel siehe Datei: 71_leuchtschild_sockel_gross_3mm_acryl__3dmk.3dmk]

Die einzelnen Bereiche am linken Bildschirmrand der Reihe nach durchgehen:
STL-Datei importieren.
Material festlegen (3mm Acryl).
Objektgröße überprüfen und falls nötig ändern.
Construction Technique: Interlocked Slices.
Unter Slice Distribution die gewünschte Anzahl und Positionierung der "Slices" wählen.
Die gewünschten Ebenenwinkel wählen. Diese Winkel genau horizontal und vertikal auszurichten ist dabei nicht immer einfach.
Exportieren als EPS unter "Get Plans".

Weiterbearbeitung mit CorelDraw - Vorbereitungen für Lasercut

Die von 123D-Make exportierte EPS-Datei wurde in **CorelDraw** importiert.
Um Material zu sparen wurde die Anordnung der Einzelteile optimiert und die (automatisch von 123D-Make generierten) Beschriftungen wurden entfernt.
Zwei zusätzlich benötigte Teile habe ich hinzugefügt: Die oberen und unteren Abdeckungen des „Leuchtkastens“ (siehe Bild zweite Reihe).
Die "Dreiecke" habe ich auch noch für die Verwendung von selbstklebenden LED-Streifen optimiert und eine Kabeldurchführung an den langen Teilen hinzugefügt.
Im Bild zu sehen sind die Teile für den kleinere Sockel.



[Kleiner Sockel siehe: 55_leuchtschild_sockel_klein__cdr_4laser.cdr
55_leuchtschild_sockel_klein__eps_4laser.eps]

[Großer Sockel siehe: 75_leuchtschild_sockel_gross__cdr_4laser.cdr
75_leuchtschild_sockel_gross__eps_4laser.eps]

Der Sockel wurde mit den Einstellungen laut Wiki für 3mm Plexiglas am Lasercutter geschnitten:

Acrylic 3mm cut: Speed: 1,1 Power: 100 PPI: 1000 Air: yes Fokus: -1,5mm

Die Teile des Sockels wurden zusammengefügt und mit Acrylkleber (Acrifix) an mehreren Stellen (innen) verklebt. Die beiden Abdeckungen wurden aber erst später befestigt (verklebt), nachdem die Beleuchtung angebracht worden war.

Die Schriftplatte

Die Schriftplatte wurde direkt in CorelDraw entworfen. Den Schriftzug (mit einer speziellen Schriftart, die auf den Computern im HAPPYLAB nicht installiert ist) habe ich zuhause mit einem Grafikprogramm in Originalgröße mit 500 dpi erstellt und im TIF-Format gespeichert. Die TIF-Datei kann CorelDraw gut importieren.



Die Materialwahl fiel auf fluoreszierendes Plexiglas, weil dadurch die Leuchtwirkung intensiver ist. Es kann aber natürlich auch jedes andere durchsichtige Plexiglas verwendet werden.

Ich habe mehrere Versionen des Schriftzuges hergestellt, dabei ist mir folgendes aufgefallen:

Schriftzug spiegeln

Die Wirkung der gravierten Schrift ist viel besser, wenn die Schrift auf der Rückseite angebracht ist. Dazu ist es nötig die Schrift (bzw. das Logo) zu spiegeln.

Tief gravieren

Der gravierte Schriftzug (oder das Logo) sollte recht tief in das Material eingraviert sein, deshalb habe ich nicht die Einstellungen laut Wiki für Plexiglasgravur verwendet, sondern habe experimentiert und bin auf folgende Werte gekommen:

Speed: 40 Power: 100 PPI: 500 Air: no Fokus: -1,5mm

Damit reicht die Gravur ca. 0,5mm tief in das Material. Möglicherweise wäre es interessant die Geschwindigkeit noch stärker zu reduzieren und damit eine noch tiefere Gravur zu erzielen. Die Schutzfolie auf der Oberseite des Schriftschildes wurde entfernt, um das Gravieren zu ermöglichen.

Hochkant für weniger Streifenbildung

Während des Graviervorganges bewegt sich der "Laserkopf" ja, ähnlich dem Druckkopf eines Tintenstrahldruckers, schnell von links nach rechts und wieder zurück und der Schriftzug wird von oben nach unten graviert. Nun ist mir aufgefallen, dass es zu unregelmäßiger Streifenbildung in den recht großen gravierten Flächen des fetten Schriftzuges kommt.



Unregelmäßige Streifenbildung bei "horizontaler Gravur"

Ich bin diesem Phänomen nicht sehr tief auf den Grund gegangen (es sieht ja auch durchaus interessant aus...), ich habe aber den Eindruck, dass es etwas mit der Länge des (je nach Buchstabenkombination unterschiedlich langen) Laserfahrweges zu tun hat. Höchstwahrscheinlich kühlt das Material bei längeren Fahrwegen auch stärker ab und die Streifenbildung wird dadurch begünstigt.

Jedenfalls kommt es zu weniger Streifenbildung, wenn das Schild im Lasercutter um 90 Grad gedreht wird und der Laserkopf somit jeden Buchstaben einzeln abarbeitet und nicht den gesamten Schriftzug auf einmal.



Reduzierte Streifenbildung bei "vertikaler Gravur"

Zum Schneiden wurden die Einstellungen laut Wiki für 3mm Plexiglas verwendet (siehe oben).

Das Schriftschild wird nicht fix mit dem Sockel verbunden, sondern nur hineingesteckt, so kann es leicht gewechselt und zum Transport entfernt werden.

[Kleine Schriftplatte siehe: 56_leuchtschild_schriftplatte_klein__cdr_4laser.cdr
56_leuchtschild_schriftplatte_klein__eps_4laser.eps]

[Große Schriftplatte siehe: 76_leuchtschild_schriftplatte_gross__cdr_4laser.cdr
76_leuchtschild_schriftplatte_gross__eps_4laser.eps]

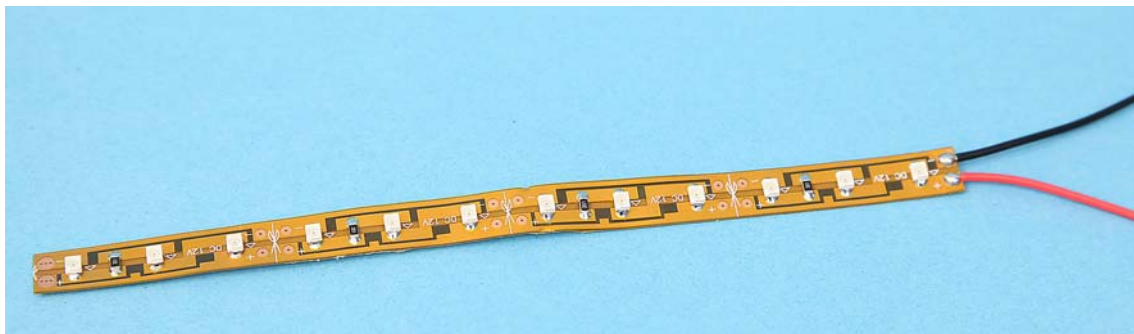
Beleuchtung

Die Beleuchtung des Schildes erfolgt von unten. Im Sockel ist Platz für etwas Technik, im größeren Socken natürlich mehr als im kleineren. Es gibt mehrere Möglichkeiten das Schild zu beleuchten.

Selbstklebende LED-Streifen und Steckernetzteil

Das ist die unaufwändigste Form der Beleuchtung.

Selbstklebende LED-Streifen gibt es in unterschiedlichen Ausführungen. Viele sind mit 12 Volt zu betreiben, ein kleines 12-Volt-Netzteil (300 - 500 mA) reicht aus.



LED-Streifen für 12 Volt

Batterie- oder Akkuspeisung

Wer das Schild unabhängig vom Netzstrom betreiben will, kann es mit Batterien oder Akkus bestücken. Eventuell funktioniert das auch recht gut mit den LED-Streifen: LiPo-Akkus mit 11,1 Volt wären dafür zum Beispiel geeignet.

Eine andere Möglichkeit: **Einzel-LEDs**. Wichtig ist dabei: jede LED braucht einen Vorwiderstand, der den Strom auf ein verträgliches Maß reduziert.

Beispiel:

LED Weiß, 3mm, 3400mcd, 60°, 20mA, 3.2V, Nichia NSPW315DS
3 x NiMH AA-Akkus, 1.2V, 2000mAh

Die Akkus haben also ca. 3,6 Volt, die LED will aber nur 3,2 Volt.
Am Vorwiderstand muss also 0,4 Volt bei 20mA (0,02A) abfallen.

$$0,4V \times 0,02A = 20 \text{ Ohm}$$

Der Vorwiderstand wäre also ca. 20 Ohm.

Platz für Akkus ist eher im Sockel des größeren Schildes. Das schmalere Schild betreibe ich mit Steckernetzteil und ein Kabel führt von hinten zum Sockel.

Blinkmuster

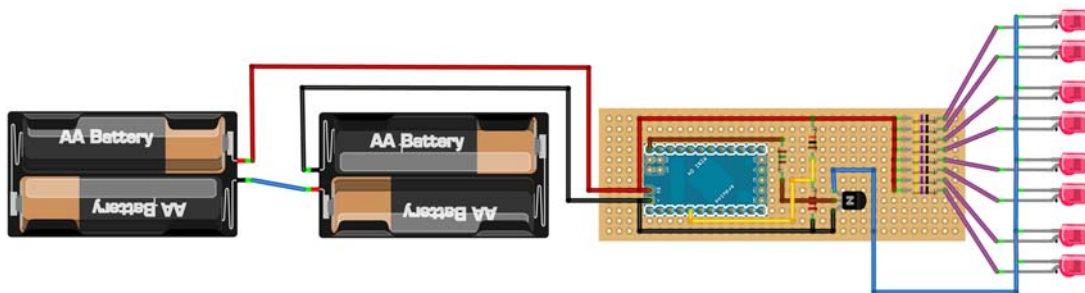
Die größte Aufmerksamkeit bekommt das Schild natürlich wenn es in irgendeiner Form blinkt. Es gibt verschiedene Möglichkeiten ein Blinkmuster („Leuchtschema“) zu steuern. Eine davon ist der Einsatz eines Mikrocontrollers.

Variante Arduino

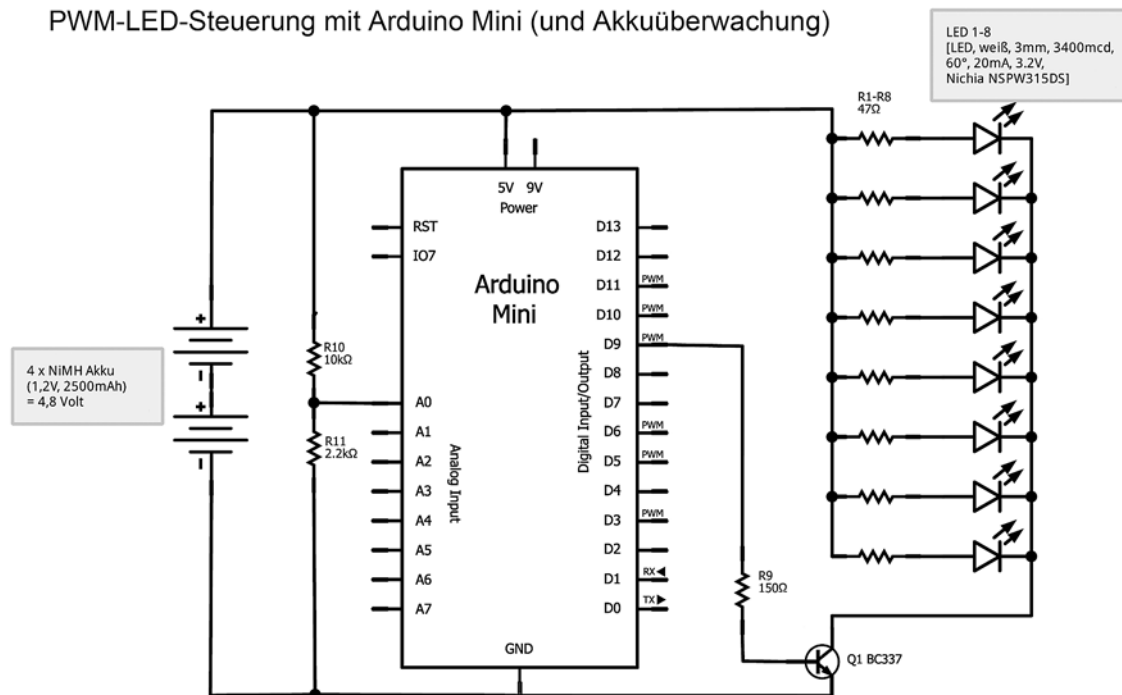
Das Arduino Controller-Board ist einfach und schnell zu programmieren, es gibt eine große Community. Bei diesem Projekt habe ich mich für den Arduino Mini entschieden, da er weniger Strom verbraucht als seine "größere Brüder", ausserdem ist er kleiner, der Platz reicht aber auch, um z.B. einen Arduino UNO oder Leonardo unterzubringen.

Das Arduino-Board wird mit ca. 5 Volt betrieben. 4 Stück NimH-Akkus (Größe AA) mit je 1,2 Volt liefern ca. 4,8 Volt, das ist nahe genug dran. Mit 4 normalen Alkali-Batterien (je 1,5 Volt) sollte dieses Board aber eher nicht betrieben werden, das wären 6 Volt, der Mikrocontroller verträgt laut Datenblatt aber maximal 5,5 Volt, das könnte also schon zu viel sein... (Natürlich könnte ein Spannungsregler verwendet werden.)

PWM-LED-Steuerung mit Arduino Mini (und Akkuüberwachung)



PWM-LED-Steuerung mit Arduino Mini (und Akkuüberwachung)



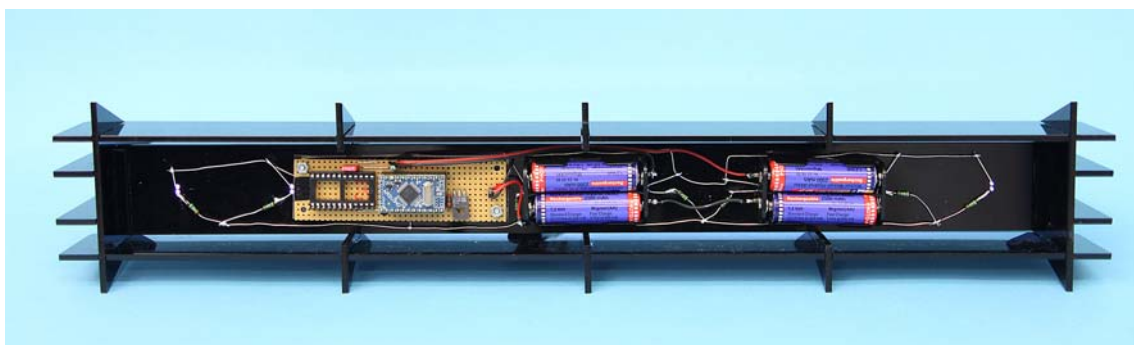
Die Schaltung braucht nur wenige externe Bauteile.

Damit die Akkus nicht tiefentladen und dadurch geschädigt werden, wird die Akkuspannung regelmäßig (unter Last) gemessen. Falls die Spannung unter 4 Volt sinkt wird abgeschaltet.

Die Widerstände R10 und R11 werden für diese Messungen gebraucht.

R1 bis R8 sind die Vorwiderstände der LEDs. Falls andere LEDs mit anderer Spannung verwendet werden, müssten diese Widerstände entsprechend angepasst werden.

Es können auch mehr als 8 LEDs zum Einsatz kommen. Der Transistor Q1 schafft mindestens 500mA ohne Probleme.



Unterseite des größeren Leuchtschildes, betriebsbereit, mit Arduino Mini.

[Code für Arduino siehe: `pwm_led_pulse_with_bat_check.ino`]

Näheres über das Arduino Controller-Board siehe <http://arduino.cc/>

Variante PIC 12F683

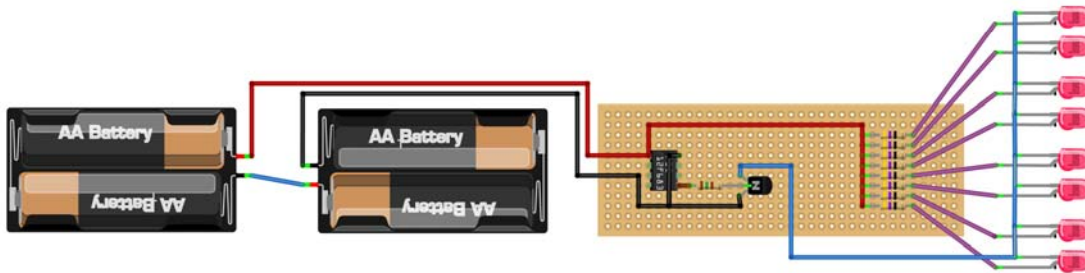
Der Mikrocontroller 12F683 der Firma Microchip kostet weniger als zwei Euro und hat alles, was zum Steuern des Leuchtschildes nötig ist, sogar ein Anschluss mit PWM (Pulsweitenmodulation) ist vorhanden. Es kann also auch ein Leuchtmuster mit sanfter Ein- und Ausblende programmiert werden.

Auch hier können 4 NiMH-Akkus zum Einsatz kommen oder ein kleines preisgünstiges Netzteil mit 5 Volt.

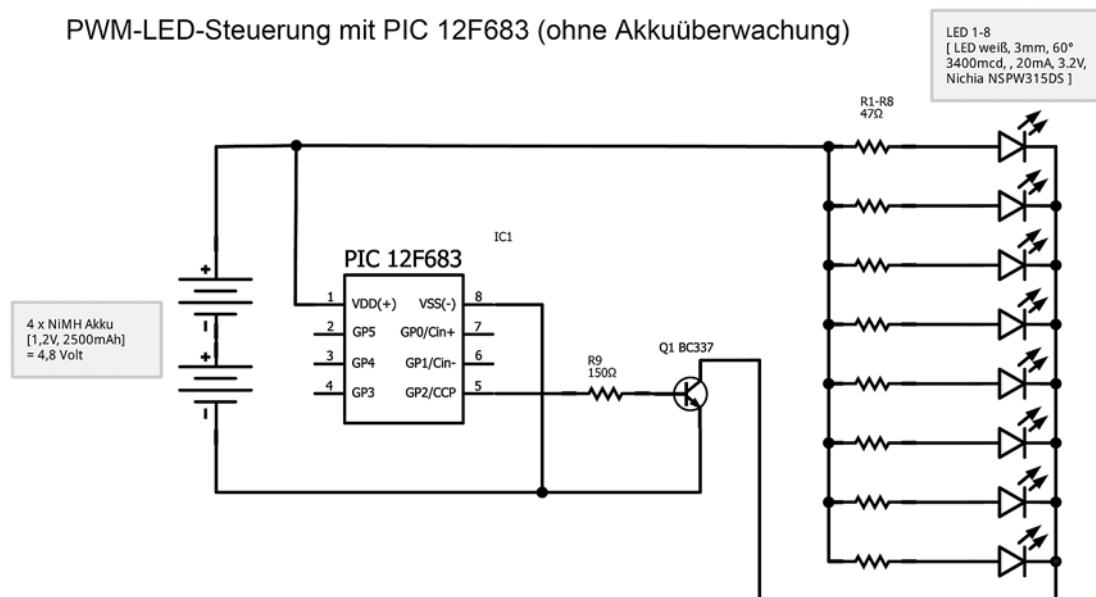
Da die Messung der Akkuspannung beim PIC12F683 einige zusätzliche Bauteile braucht, gibt es hier auch eine ganz einfache Schaltungs- und Programm-Variante ohne Akkuüberwachung. Eine Akkuladung reicht ja für 10 oder mehr Stunden (siehe weiter unten) bzw. kann ja ein Steckernetzteil (z.B. 5 Volt, 300-500mA) verwendet werden. Wer also mit möglichst wenigen Bauteilen auskommen möchte, kann diesen Code und diesen Schaltplan verwenden.

[Assembler-Code für PIC 12F683 siehe: pwm_long_pulse_12f683.asm]

PWM-LED-Steuerung mit PIC 12F683 (ohne Akkuüberwachung)



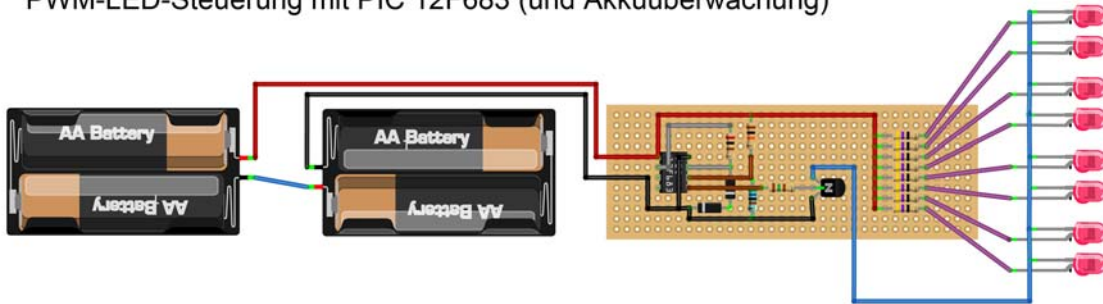
PWM-LED-Steuerung mit PIC 12F683 (ohne Akkuüberwachung)



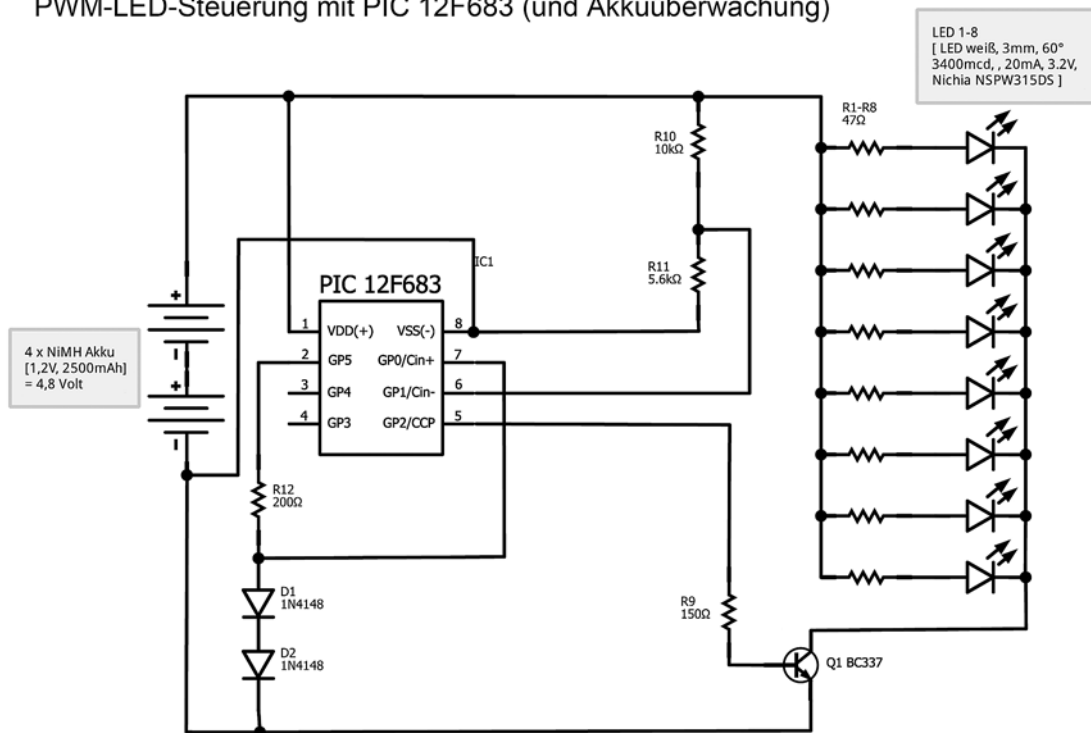
Und hier die "Luxusvariante" mit Akkuüberwachung und automatischer Abschaltung bei niedriger Akkuspannung.

[Assembler-Code für PIC 12F683 siehe: pwm_long_pulse_with_bat-watch_12f683.asm]

PWM-LED-Steuerung mit PIC 12F683 (und Akkuüberwachung)

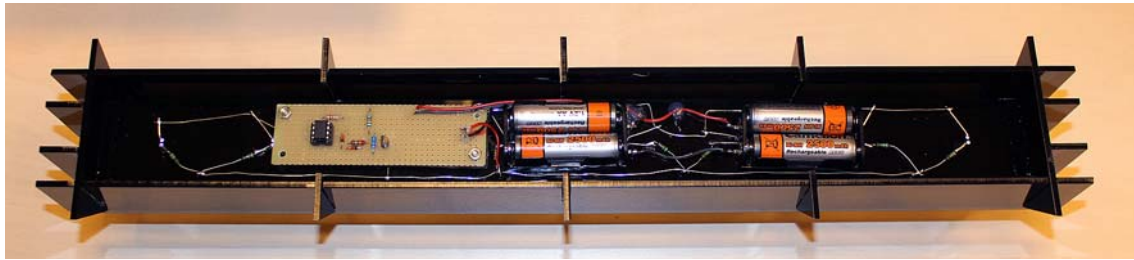


PWM-LED-Steuerung mit PIC 12F683 (und Akkuüberwachung)



Die beiden Dioden D1 und D2 erzeugen eine Referenzspannung von ca. 1,4 Volt, die relativ unabhängig von der Akkuspannung ist. Diese Referenz wird dem nicht-invertierenden Eingang des Komparators zugeführt (Pin 7). Ausserdem wird mit der Akkuspannung über R10 und R11 der invertierende Eingang des Komparators gespeist (Pin 6). Bei vollen Akkus ist die Spannung an Pin 6 höher als an Pin 7. Mit sinkender Akkuspannung sinkt aber auch die Spannung am Pin 6, die Spannung an Pin 7 bleibt aber einigermaßen konstant. Sinkt die Akkuspannung unter 4Volt (unter Last) wird die Spannung an Pin 6 kleiner als an Pin 7 und der Komparator "kippt". Das Programm fragt regelmäßig der Zustand des Komparators ab

und kann so erkennen, wann der Akku leer ist. In diesem Fall werden LEDs und Mikrocontroller abgeschaltet.



Unterseite des größeren Leuchtschildes, betriebsbereit, mit PIC 12F683.

Um ein Programm auf einen PIC-Mikrocontroller zu bekommen wird spezielle Software und Hardware benötigt.

Die Software, ein Entwicklungstool (z.B. MPLAB-IDE), kann kostenlos von der Microchip-Website heruntergeladen werden <http://www.microchip.com/mplab8>

Die Hardware - ein Programmiergerät auch "Brenner" genannt - gibt es vom Hersteller des Mikrocontrollers (z.B. PICkit3), von anderen Herstellern oder zum Selberbasteln (z.B. <http://www.sprut.de/>).

Laufzeit einer Akkuladung unter Realbedingungen

Ich habe, wie oben beschrieben, 4 x NiMH AA-Akkus, 1.2V, ca. 2500mAh verwendet.

Beim größeren Schild sind 8 LEDs des oben genannten Typs verbaut.

Ich habe bei der Arduino-Variante folgendes gemessen:

Gesamtstromaufnahme bei 100% Leuchtkraft ca. 250mA

Gesamtstromaufnahme bei 0% Leuchtkraft ca. 25mA

Bei einer mehrtägigen Veranstaltung kam das "Long-Pulse"-Leuchtschema zum Einsatz.

Dabei leuchtet das Schild etwa 60% der Zeit mit voller Leuchtkraft in der verbleibenden Zeit wird ab- und wieder aufgeblendet. Die durchschnittliche Stromaufnahme schätze ich auf ca. 210mA. Rechnerisch sollte eine Akkuladung also für mehr als 10 Stunden reichen. Das dürfte auch stimmen: Die längste durchgehende Einsatzdauer war etwa 11 Stunden. Nach dieser Zeit war die Spannung auf ca. 4,2V gesunken (also auf ca. 1,05V pro Zelle), die Akkus waren also noch nicht "ganz leer". Das Schild leuchtete dann zwar nicht mehr ganz so hell wie mit vollen Akkus, aber das ist „verkraftbar“. Wer möchte, könnte natürlich Maßnahmen treffen, um den LED-Strom konstanter zu halten.

Eine Schutzschaltung, damit die Akkus nicht tiefentladen werden ist zum Teil in den Programmen für die Mikrocontroller enthalten. Wird die Akkuspannung zu niedrig und geht unter Last gegen 4 Volt (also 1V je Zelle) wird abgeschaltet, um die Akkus nicht zu schädigen.

Viel Spaß beim Nachbauen!

Verbesserungen sind willkommen!

Udo Somma