

Solarzellen, Supercaps und ein kleiner Mikrocontroller

Ein Arduino-kompatibles Solarlicht für den Garten

Kennen Sie das auch, die vielen Dinge, die Sie immer schon einmal ausprobieren wollten? Supercaps hatte ich zum Beispiel schon länger auf dem Radar, Solarmodule wollte ich ebenfalls einmal irgendwo einbauen. Und kurz nachdem im Internet eine Anleitung zur Programmierung von ATtiny85-Controllern über die Arduino-Umgebung aufgetaucht war, lagen auch von diesen kleinen Controllern 10 Stück im Schrank. Für die tägliche Arbeit war all dies leider nicht zu gebrauchen, ein „Privatprojekt“ musste her: ein Arduino-kompatibles Solarlicht mit Supercaps für den Garten.

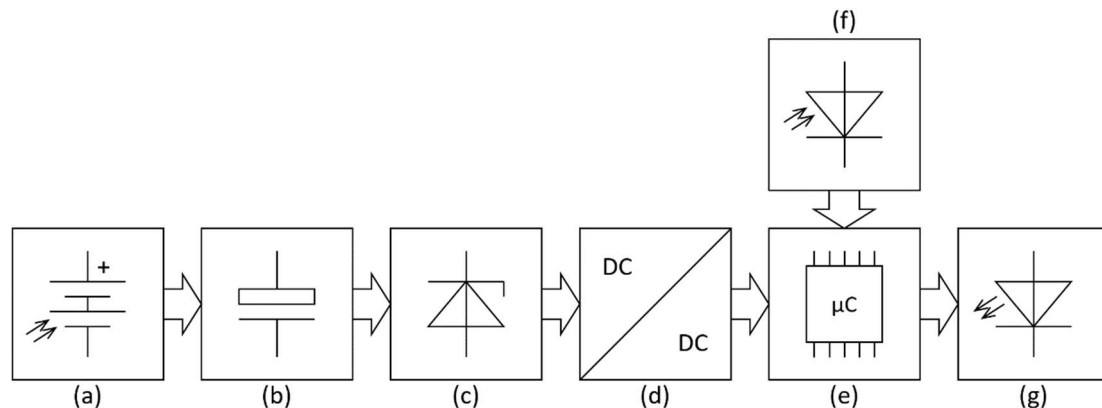


Abbildung 1 - Struktur des Gesamtsystems

Abbildung 1 zeigt die Struktur des Gesamtsystems: Über ein Solarmodul (a) werden zwei parallel geschaltete Supercaps (b) geladen, die sich durch eine erhebliche Kapazität, hier insgesamt 100 Farad, bei kleiner Baugröße auszeichnen. Allerdings muss die geringe Spannungsfestigkeit der Kondensatoren von nur jeweils 2,7 Volt beachtet werden, ein Schutz gegen Überladung (c) ist unverzichtbar. Ein Step-Up-Wandler (d) wandelt die vom Ladezustand der Kondensatoren abhängige Spannung in eine Gleichspannung von 5 Volt zur Versorgung des Mikrocontrollers (e). Dieser erfasst die aktuelle Helligkeit über eine Fotodiode (f) und steuert schließlich – sofern es dunkel genug ist – drei Leuchtdioden (g) an.

Beschreibung der Schaltung

Die Schaltung des Stromversorgungsteils – vom Solarmodul bis zum Step-Up-Wandler – ist in Abbildung 2 dargestellt: SZ1 liefert bei voller Beleuchtung einen Strom von 200 mA bei einer Spannung von 4 Volt und lädt so die beiden Supercaps C1 und C2 auf. Die Diode D1 ist notwendig, damit sich die geladenen Kondensatoren bei Dunkelheit nicht wieder über das Solarmodul entladen. Sind die Supercaps auf ca. 2,5 Volt aufgeladen, leitet die aus R1-R3, IC2 und T1 bestehende Begrenzungsschaltung den in der Regel weiterhin fließenden „Solarstrom“ ab und verhindert eine Überladung von C1 und C2. Die Begrenzungsschaltung ist dem Datenblatt der 2,5-Volt-Referenz TL431 [1] entnommen, im Solarlicht wird indes die stromsparendere 1,24-Volt-Referenz TLV431 eingesetzt. Ein kleiner Step-Up-Wandler auf Basis des recht bekannten NCP1402 setzt die je nach Ladezustand variierende Spannung an den Supercaps auf 5 Volt herauf. Dieser Wandler arbeitet mit einer Schaltfrequenz von max. 200 kHz und kann bei einer Ausgangsspannung von 5 Volt einen Strom von bis zu 200 mA liefern, für einen kleinen Mikrocontroller und drei Leuchtdioden sollte dies genügen. Vorteilhaft ist, dass der Wandler selbst bei einer Eingangsspannung von 0,3 Volt noch arbeitet, wenn auch die Ausgangsspannung unter Belastung dann schon etwas absinkt: So ist es möglich, die in den Supercaps gespeicherte Energie fast vollständig auszunutzen.

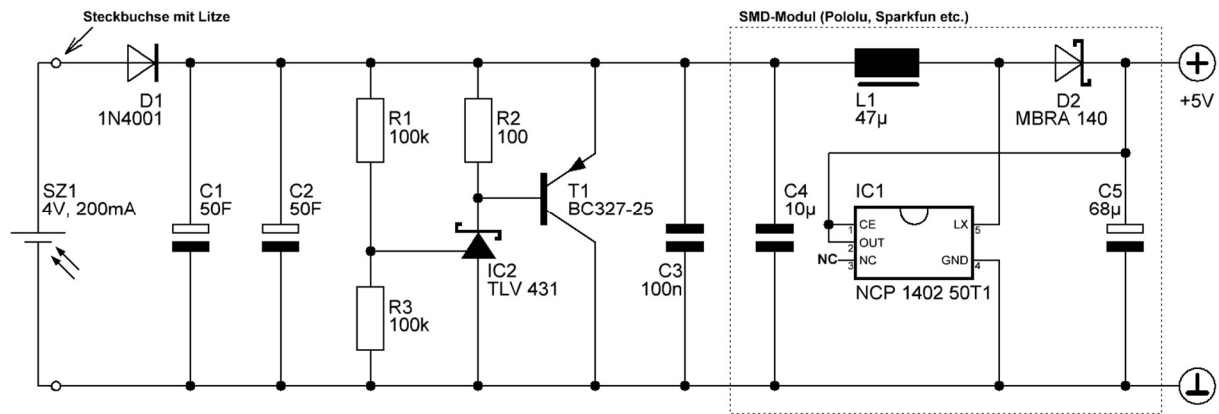


Abbildung 2 - Schaltung des Stromversorgungsteils

Der Mikrocontroller ATtiny85, ein 8-Bit-Controller der Firma Atmel, wird mit seinem intern generierten Takt von 1 MHz betrieben. Er verfügt unter anderem über einen 10-Bit-Analog-Digital-Wandler mit mehreren Eingängen. An einen dieser Eingänge ist die Fotodiode D3 zur Helligkeitsmessung angeschlossen. Mehrere PWM-Ausgänge erleichtern die Programmierung „fließender“ Farbübergänge zwischen den angeschlossenen Leuchtdioden. 8 KB Programmspeicher und 512 Byte SRAM sind mehr als ausreichend für das hier realisierte Solarlicht. Die Ausgänge des ATtiny85 können genug Strom liefern, um Standard-Leuchtdioden direkt anzusteuern. Zu beachten ist lediglich, dass aus allen Ausgängen in Summe nicht mehr als 60 mA fließen dürfen [2] – deutlich mehr, als hier benötigt wird. Abbildung 3 zeigt die Schaltung des Mikrocontrollerteils.

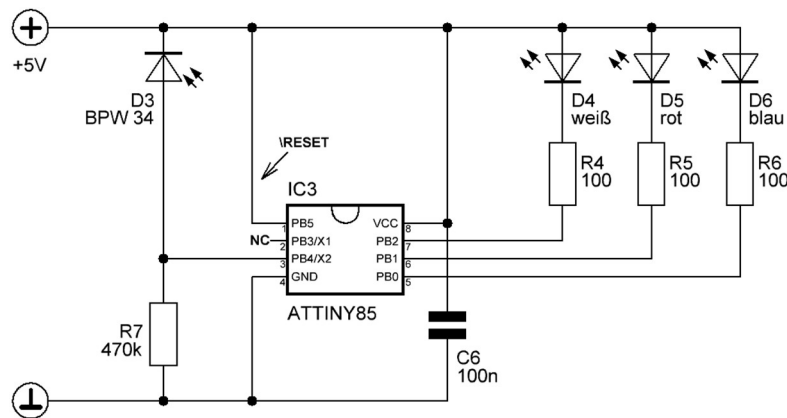


Abbildung 3 - Schaltung des Mikrocontrollerteils

Programmierung des Mikrocontrollers

Die Firmware des Solarlights ist auf GitHub [4] frei verfügbar. Es handelt sich letztlich um einen Arduino-Sketch, der wie auf der Webseite der „High-Low Tech Group“ [3] beschrieben kompiliert und auf den ATtiny85-Mikrocontroller aufgespielt werden kann. Zur Programmierung sind ein Arduino Uno, eine Steckplatine sowie einige Drähte erforderlich. Da die Betriebsspannung des Mikrocontrollers beim Auf- und Entladen des Solarlights stark schwankt, ist es sinnvoll, die Brownout-Erkennung des Mikrocontrollers zu aktivieren: Sinkt die Betriebsspannung unter 2,7 Volt, wird der Mikrocontroller automatisch im Reset-Zustand gehalten, bis sich die Betriebsspannung wieder erholt hat. Abstürze des Mikrocontrollers als Folge von Spannungseinbrüchen werden so verhindert.

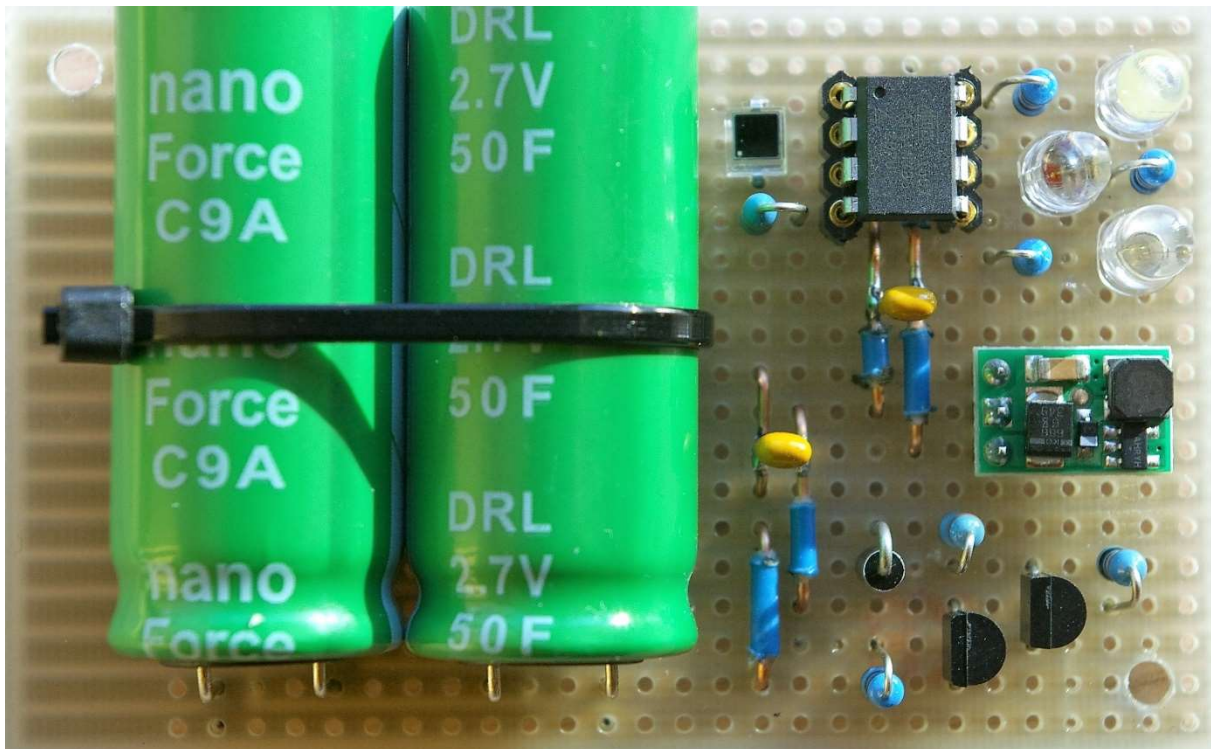


Abbildung 4 - Platine von oben

Zur Aktivierung der Brownout-Erkennung sind im Anschluss an das Aufspielen der Firmware die Fuse-Bits des Mikrocontrollers anzupassen. Dies erfolgt mit dem Programm „avrdude“, welches standardmäßig zusammen mit der Arduino-Software auf dem PC installiert wird. Befindet sich die Arduino-Software wie üblich im Ordner „C:\Program Files (x86)\Arduino“, sind dazu die folgenden Befehle über die Eingabeaufforderung auszuführen [5]. Der zweite Befehlsaufruf („avrdude...“) ist dabei in einer einzelnen Zeile einzugeben:

```
cd "C:\Program Files (x86)\Arduino\hardware\tools\avr\bin"
avrdude -p t45 -c stk500v1 -P com4 -b 19200 -v -C ..\etc\avrdude.conf
-U lfuse:w:0x62:m -U hfuse:w:0xdd:m -U efuse:w:0xff:m
```

Aufbau und Inbetriebnahme

Das Solarlicht besteht im Wesentlichen aus einigen wenigen Standardbauteilen, entsprechend einfach gestaltet sich der Aufbau auf einer 80 x 50 mm großen Streifenrasterplatine. Als Leuchtdioden sollten superhelle Typen (5 mm) mit einem Nennstrom von 20 mA ausgewählt werden. Der Step-Up-Wandler (NCP1402) ist zwar nur als SMD-Bauelement im SOT-23-Gehäuse verfügbar, bei verschiedenen Anbietern gibt es jedoch komplette Module [6], welche über eine dreipolige Stiftleiste leicht mit der Streifenrasterplatine verbunden werden können.

Abbildung 4 zeigt die Oberseite, Abbildung 5 die Unterseite der fertig aufgebauten Platine. Eine Steckbuchse mit kurzer zweipoliger Litze verbindet diese mit dem Solarmodul, siehe Abbildung 6. Weitere Fotos vom Aufbau des Solarlichts finden sich auf der bereits erwähnten GitHub-Seite. Dort liegt auch ein detaillierter Schaltplan des Solarlichts, auf dem die genauen Typen der Supercaps, des Solarmoduls sowie eines passenden Gehäuses angegeben sind.

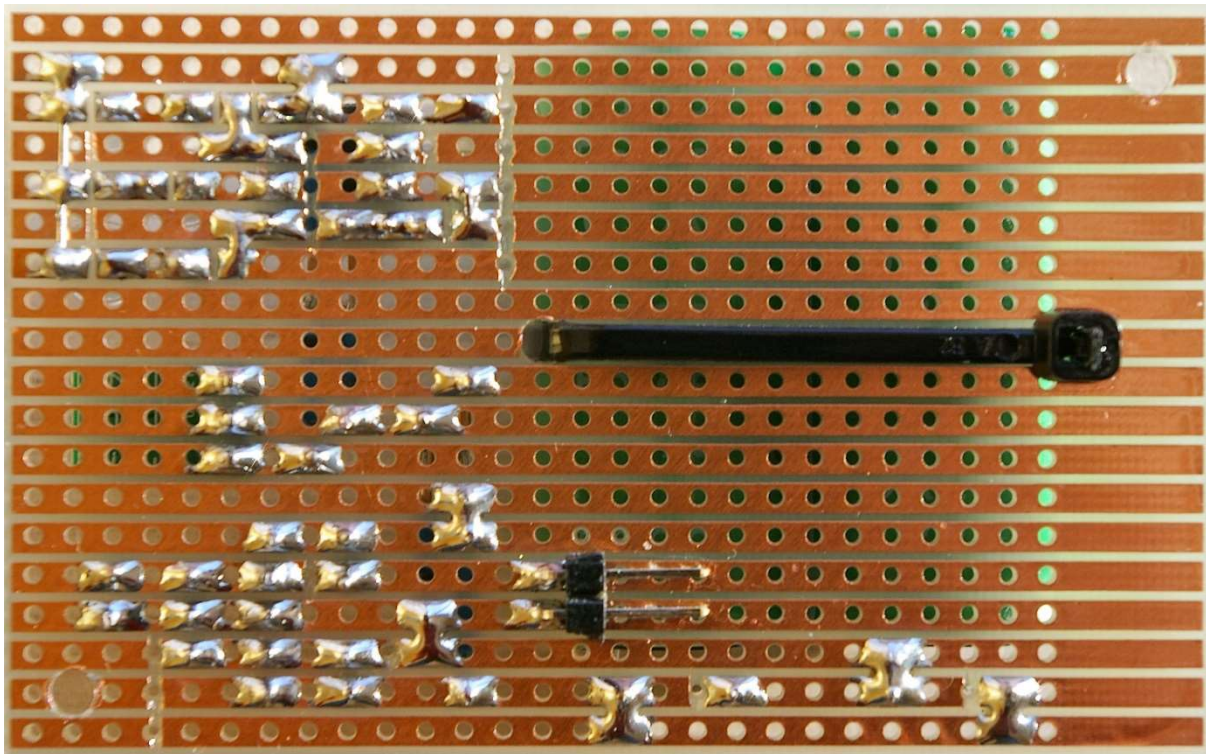


Abbildung 5 - Platine von unten

Lessons Learned

Nur selten gelingt es, ein Projekt auf Anhieb ohne Fehler zum Laufen zu bringen. Beim Solarlicht war das nicht anders:

1. Ursprünglich war zur Stromversorgung eine Parallelschaltung zweier kleinerer Solarmodule [7] mit einem Nennstrom von 35 mA bei einer Spannung von 4 Volt vorgesehen, um das Solarlicht in eine alte Colaflasche (0,5 Liter) einbauen zu können. Diese Solarmodule lieferten allerdings weniger Strom als erwartet, die Supercaps wurden an trüben Tagen nicht vollständig aufgeladen.
2. Für die ersten Tests des Solarlichts hatte ich einen Step-Up-Wandler auf Basis des „klassischen“ LT1073 [8] direkt auf die Streifenrasterplatine gelötet, leider ohne Erfolg. Wenn nämlich nach einem Kaltstart die Ladespannung der Supercaps ca. 0,7 Volt erreicht, öffnet der LT1073 seinen internen Schalttransistor gerade weit genug, um ein weiteres Aufladen sicher zu verhindern: Wenn im Datenblatt die Funktion eines Bauteils ab 1 Volt garantiert ist, so bedeutet dies offensichtlich nicht, dass das betreffende Bauteil sich unterhalb dieser Spannung rein passiv verhält...
3. Das Solarmodul wird bei der Montage des Solarlichts auf die Außenseite des transparenten Gehäuses aus Polycarbonat großflächig aufgeklebt. Zunächst hatte ich dazu Sanitärsilikon aus dem Baumarkt verwendet, welches allerdings auf glattem Polycarbonat nur schlecht haftet. Bessere Ergebnisse erzielt man mit Montagekleber, der speziell für Kunststoffe geeignet ist.

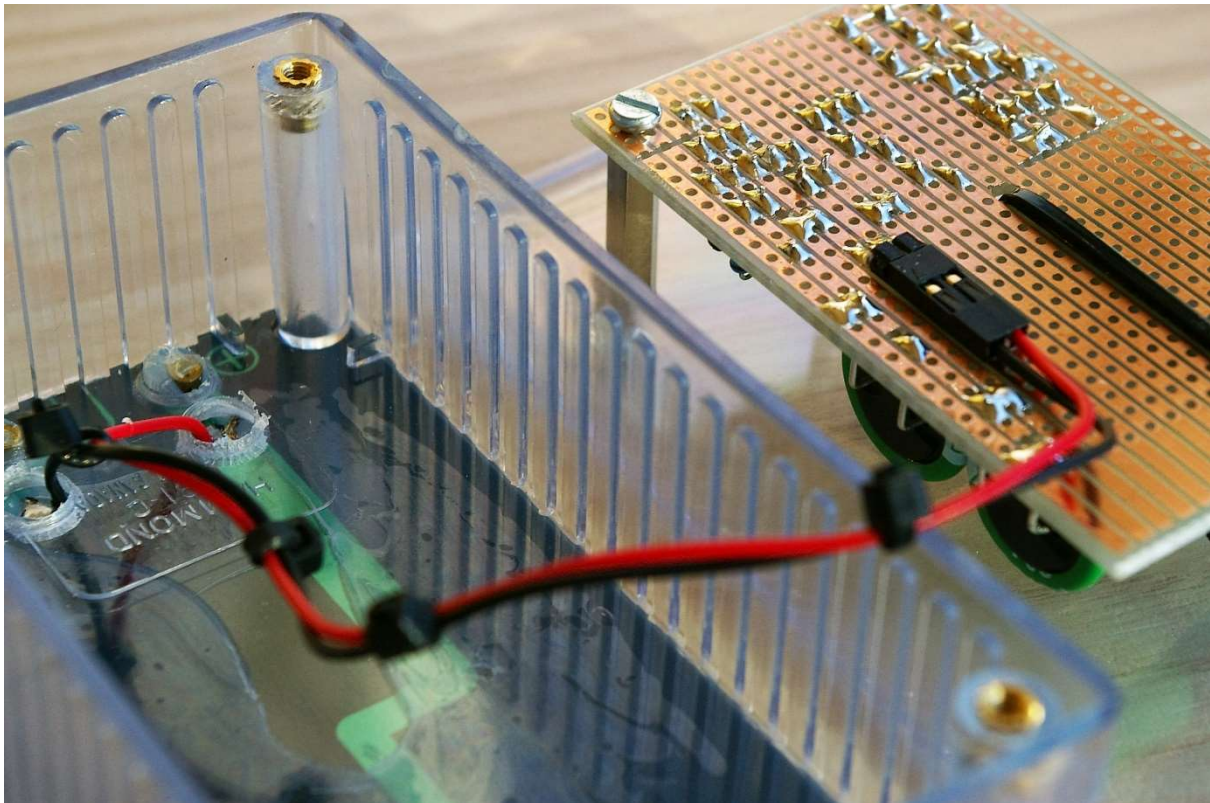


Abbildung 6 - Anschluss des Solarmoduls

Quellen und Links

- [1] Texas Instruments, Datenblatt TL431, <http://www.ti.com/lit/ds/symlink/tl431.pdf>, Abbildung 31: „High-Current Shunt Regulator“
- [2] Atmel Corporation, ATtiny85, http://www.atmel.com/Images/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf, Seite 162, Fußnote 5: „The sum of all IOH, for all ports, should not exceed 60 mA.“
- [3] MIT Media Lab, High-Low Tech Group, <http://highlowtech.org/?p=1695>: „Programming an ATtiny w/ Arduino 1.6 (or 1.0)“
- [4] GitHub-Projekt zum Solarlicht: <https://github.com/t4n/arduino-solar/wiki>
- [5] Zur Berechnung der benötigten Fuse-Werte wurde der „Engbedded Atmel AVR Fuse Calculator“ genutzt: <http://www.engbedded.com/fusecalc/>
- [6] Passende NCP1402-Module mit einer Ausgangsspannung von 5 Volt sind zum Beispiel bei Pololu (<http://www.pololu.com>, item #: 798) oder Sparkfun (<http://www.sparkfun.com>, PRT-10968) erhältlich.
- [7] Monokristallines Solarmodul YH-39X35 (<http://www.conrad.de>, Bestellnummer 191308)
- [8] Linear Technology, LT1073, <http://cds.linear.com/docs/en/datasheet/1073fa.pdf>

Der vorliegende Artikel „Solarzellen, Supercaps und ein kleiner Mikrocontroller“ wurde im August 2016 im Online-Magazin myembedded.de veröffentlicht, www.myembedded.de/2016/08/11/solarzellen-supercaps-und-ein-kleiner-mikrocontroller-2/



Abbildung 7 - Das fertig aufgebaute Solarlicht