

Do It Yourself: Fahrrad-Rücklicht

Niklas Ebell

1. Januar 2016

Inhaltsverzeichnis

1	Einführung	4
2	Vorbereitungen	4
2.1	Maschinen und Geräte	4
2.2	Software	4
3	Umsetzung	5
3.1	Elektronik	5
3.1.1	Platine	5
3.1.2	Energieversorgung	6
3.1.3	Programmierung der Schaltung	7
3.2	Mechanik	8
3.2.1	Gehäuse	8
4	Zusammenfassung	8

Abbildungsverzeichnis

1	Schaltplan-Layout	6
2	Frontseite des Platinenlayout	7
3	Rückseite des Platinenlayout	8
4	3D-Modell des Hauptgehäuses	9

1 Einführung

Unter Einbeziehung möglichst vieler Produktionsmaschinen im FabLab (Quelle anzeigen) soll in meinem übersichtlich gehaltenen Projekt ein LED-Rücklicht zur Befestigung an einem Fahrrad entworfen werden.

Neben dem Gehäusebau wird ein besonderes Augenmerk auf die Schaltungstechnik mit Mikrocontroller auf einer selbst entwickelten Platine gelegt. Am Ende des Projektes soll ein einsatzbereites Fahrradrücklicht entwickelt worden sein, sowie Kenntnisse in der Elektronik erlangt werden, darunter das Platinenlayout, computergestützte 3D-Modellation, 3D-Druck, SMD-Löten sowie Mikrocontroller-Programmierung.

2 Vorbereitungen

2.1 Maschinen und Geräte

Es werden folgende Gerätschaften zur Herstellung des Fahrradlichtes benötigt:

- 1) Papier und Stift für erste Zeichnungen
- 2) Drucker für Belichtungsvorlage
- 2) Belichtungsapparat für Platinenlayout
- 3) Ätznatron zum Platinenentwickeln
- 4) Ätzbecken mit Ätzlösung
- 5) Verzinnungsbecken/Lösung für Platine
- 6) Bohrer
- 7) Lötkolben/Lötstation
- 8) Programmer für Mikrocontroller Atmel
- 9) 3D-Drucker

2.2 Software

Heutzutage sind computergestützte Programme zur Erstellung von Platinen oder 3D-Gebilden kaum mehr aus dem Alltag wegzudenken. Wenn immer möglich nutze ich auch OpenSource-Software. Bei Software, die keine OpenSource ist, findet sich nach Recherche meist oft eine gleichwertige Software aus dem OpenSource-Bereich.

Auch ich nutze einige Hilfen, wie die folgende Liste zeigt:

- 1) Schaltplanerstellung /Platinenlayout mit KiCad
- 2) 3D-Modellation mit SolidEdge ST8
- 3) Mikrocontroller-Programmierung mit der Arduino IDE
- 4) Dokumentation mit Texmaker und Miketex

3 Umsetzung

3.1 Elektronik

Im Bereich der Elektronik müssen verschiedene Themen betrachtet werden. Nachdem der Funktionsumfang des Fahrradlichtes feststeht, müssen die benötigten und aufeinander abgestimmten Elektronik-Komponenten ausgesucht und beschafft werden. Auch dem Thema der Energieversorgung muss sich gewidmet werden. Verschiedene Elektronik-Komponenten müssen unter Umständen mit unterschiedlichen Spannungsniveaus versorgt werden.

Folgende Bauelemente werden für die von mir verwendete Schaltung benötigt:

- 1) 3 helle rote LEDs
- 2) 3 Transistoren vom Typ B548C
- 3) 1 ISP-Connector (6-Pin)
- 4) 1 Mikrocontroller vom Typ Atmel Attiny45V
- 5) 1 Batteriehalterung??
- 6) Kondensator 100nF
- 7) Batterien/Akku?

3.1.1 Platine

In einer Software wie KiCad für Schaltplan-Layouts werden die Komponenten angeordnet und miteinander verschaltet. Den Komponenten werden dann physische Modelle/Gehäuse zugeordnet, um einen Aufbau der Schaltung auf einer Platine zu ermöglichen.

Abbildung 1 zeigt das Schaltplan-Layout in KiCad. Ein Mikrocontroller von Atmel vom Typ Attiny45V in SOIC-Ausführung ist die Basis dieser Fahrradlicht-Schaltung. Über 3 I/O-Pins des Mikrocontrollers werden über Transistoren vom Typ B548C drei helle rote LEDs angesteuert. Diese sind direkt an eine Energieversorgung angeschlossen. Da der Mikrocontroller nur einen max. Strom von 20mA pro Pin ausgeben kann, wird hier die Ansteuerung der LEDs über Transistoren realisiert. Somit muss nur ein kleiner Ansteuerungsstrom an der Basis des Transistor anliegen um einen höheren Strom direkt aus der Energieversorgung an die LEDs zu leiten.

Außerdem ist eine ISP-Steckverbindung an den Mikrocontroller angeschlossen. Diese Verbindung mit 6 Pins ist der Standard-Stecker zur Programmierung des Mikrocontroller-Chips für die In-System-Programmierung und erlaubt somit das einfache Anschließen eines Programmers an die Schaltung.

Zur Glättung der Eingangsspannung und Reduzierung von Störungen ist ein 100nF-Kondensator an die Versorgungsspannung angeschlossen.

Die Bauteile werden dann auf einer 'virtuellen' Platine angeordnet und die Leiterbahnen verlegt. Dabei können die Leiterbahnen entweder manuell verlegt werden oder bei einfachen Schaltungen die Autoroute-Funktion benutzt werden. Um ein leistungsfähigeres System für Leiterbahn-Berechnung zur Verfügung zu haben, habe ich den externen Leiterbahn-Editor 'FreeRoute' benutzt und das Leiterbahnen-Layout anschließend wieder zurück in KiCad importiert.

Abbildung 2 zeigt die Vorderseite des Platinenlayouts. Große Flächen auf der Platinen, die das gleiche Potential besitzen wurden ausgefüllt, um Ätzlösung zu sparen.

Abbildung 3 zeigt die zugehörige Rückseite der Platine. Es wird also auf insgesamt zwei Layern gearbeitet. Dadurch ergibt sich auch die Notwendigkeit, Löcher für Durchkontakt-

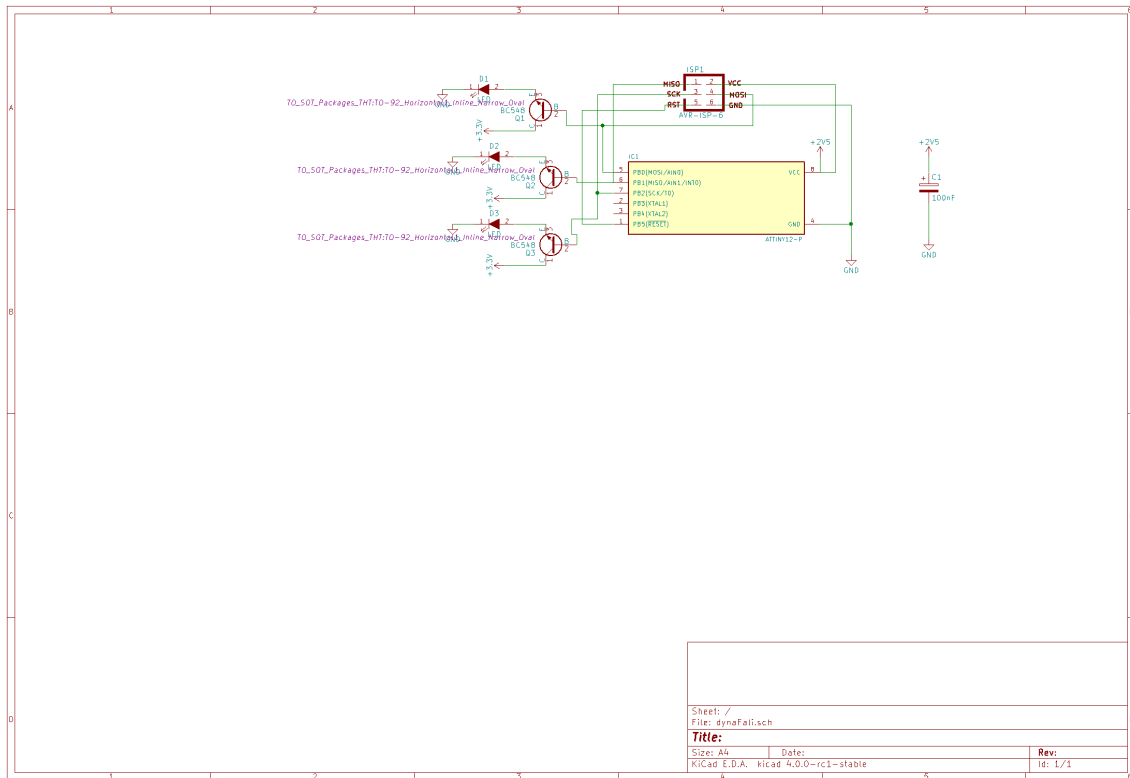


Abbildung 1: Schaltplan-Layout

tierungen in die Platine zu bohren. Ebenfalls werden Bohrungslöcher für die Befestigung von einigen Bauteilen, wie den LEDs benötigt.

Platine ätzen Platine bestücken Platine testen Platine einbauen

3.1.2 Energieversorgung

Ein Fahrradlicht am Fahrrad ist eine mobile Applikation. Somit wird auf der einen Seite eine hohe Lebensdauer von dem System erwartet. Auf der anderen Seite muss ein Fahrradlicht auch eine angemessene Leuchtkraft besitzen, damit es für den Einsatz im Straßenverkehr geeignet ist. Deshalb wird schon bei der Auswahl der Komponenten auf leistungsstarke, jedoch stromsparende Bauelemente Wert gelegt. Der Atmel Attiny45V kann z.B. auf niedrigen energiesparenden Frequenzen takten und hat einen großen Bereich der Betriebs-Eintrittsspannung. Außerdem ist der Mikrocontroller auf die Größe des Projektes angepasst. So reichen die sechs angebotenen I/O-Pins des Atmel Attiny45V allemal für die einfache Funktion des Fahrradlichtes mit LEDs. Die volle Leuchtkraft der LEDs wird durch den Einsatz der Transistoren gewährleistet. Der maximale Strom durch die LEDs wird somit nicht von den max. verfügbaren Ausgangsströmen der I/O-Pins des Mikrocontrollers limitiert, sondern durch die Leistungsstärke der Energieversorgungseinheit.

Auch das Programmieren kann energieeffizient ausgelegt werden. Dabei können zu allererst die Betriebsmodi geschickt gewählt werden. In einer Schaltung wie die des Fahrradlichtes wird die Lebensdauer von dem 'Energiehunger' der LEDs bestimmt. Somit kann

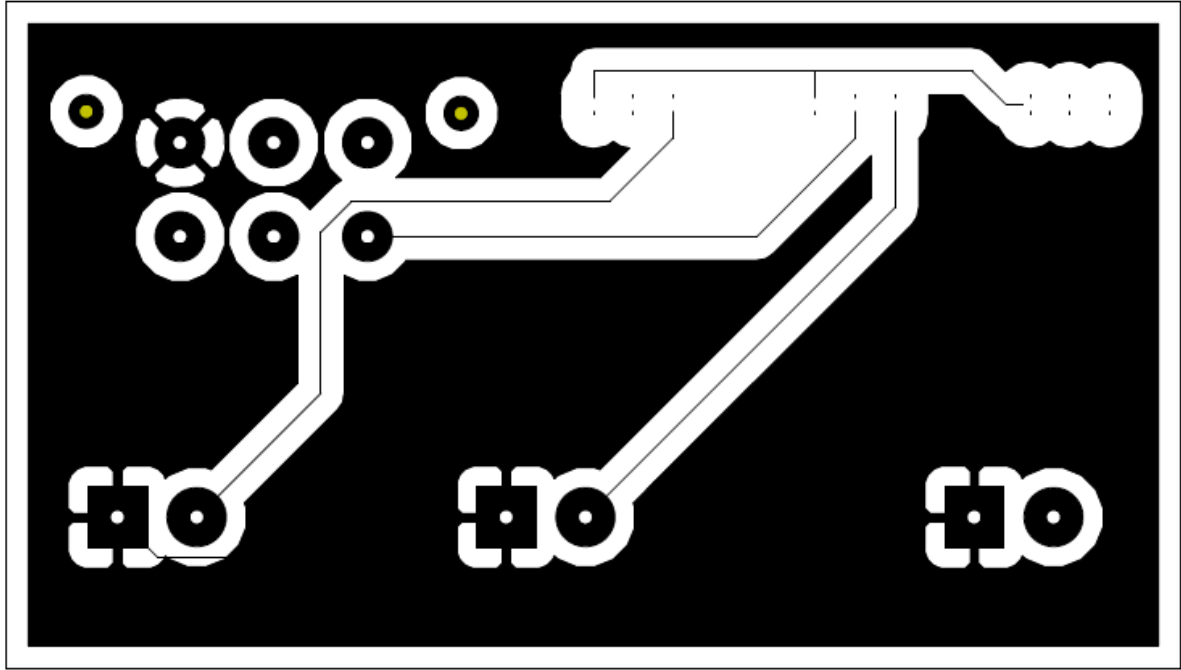


Abbildung 2: Frontseite des Platinenlayout

durch Betriebsmodi wir Blinken die Lebensdauer bis zum Wiederaufladen des Akkus verdoppelt werden. Ebenso ist es möglich den Mikrocontroller zwischen zwei Schaltzuständen in einen Schlafmodus zu befördern. Da dies keinen so großen Energieeinspareffekt bedeutet, jedoch einfach zu bewerkstelligen ist, findet der Schlafmodus bei unserer Programmierung der Schaltung Anwendung.

3.1.3 Programmierung der Schaltung

Die Programmierung des Mikrocontroller erfolgt durch das Programmiergerät Atmel AVR ISP MKII. Durch den Anschluss der sechs Pins des ISP-Interfaces auf der Platine mit der Programmer-Einheit lässt sich ein Bootloader und Sketch auf den Mikrocontroller übertragen. Der Sketch wird auf der Programmier-Plattform Arduino IDE für Windows geschrieben.

Das Fahrradlicht stellt drei Betriebsmodi zur Verfügung:

1. Betriebsmodus: normales Blinken aller drei LEDs
2. Betriebsmodus: schnelles Blinken aller drei LEDs
3. Betriebsmodus: normales Blinken der drei LEDs nacheinander

Der folgende auskommentierte Programmcode wird für das Fahrrad-Rücklicht auf den Mikrocontroller hochgeladen.

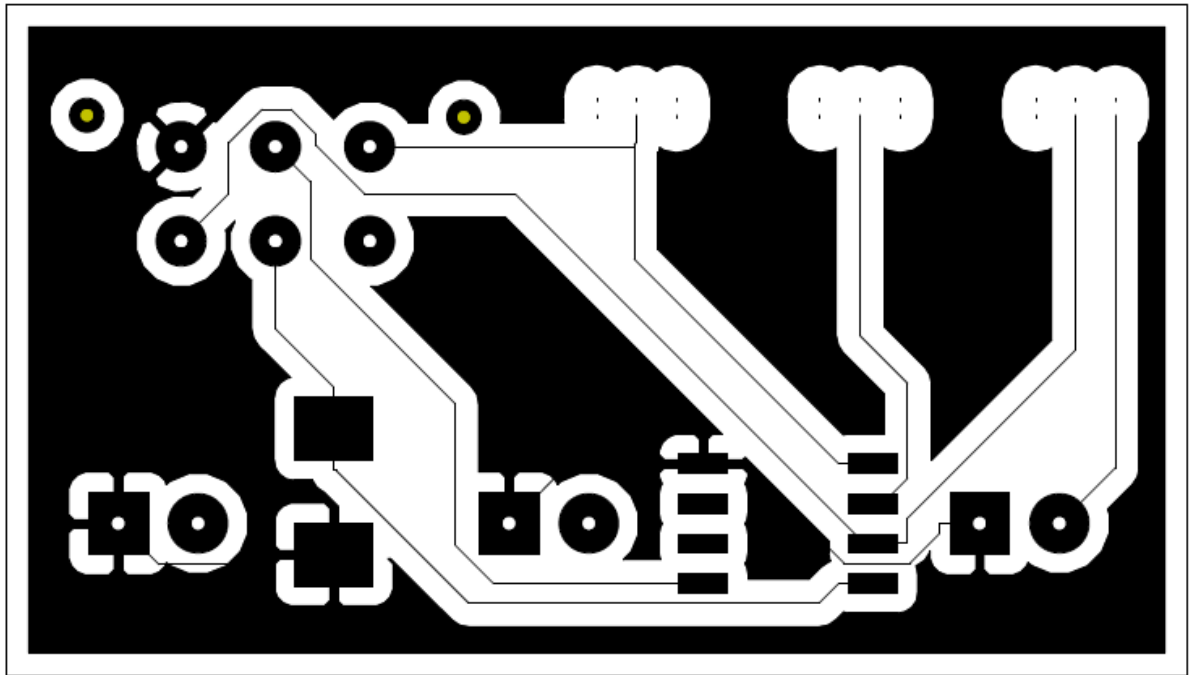


Abbildung 3: Rückseite des Platinenlayout

3.2 Mechanik

Der mechanische Teil dieses Projektes besteht hauptsächlich aus dem Gehäusebau, der die komplette Schaltung/Platine, die Energieversorgung und eine universelle Befestigungseinrichtung zum Fahrrad umfassen soll. Neben einer kleinen Skizze auf Papier wird hierbei hauptsächlich auf computergestützte Modellation zurückgegriffen. SolidEdge ST8 ist ein CAD-Programm von der Siemens AG zur Erstellung von 3D-Modellen.

3.2.1 Gehäuse

Das Gehäuse soll aus drei Teilen bestehen. Das Hauptgehäuse als Behausung für die Platine und Energieversorgung besteht aus zwei trennbaren Teilen, die zur Montage aneinander geschraubt werden. Das dritte Gehäuseteil ist die Befestigungseinrichtung für das Fahrrad.

Gehäuse drucken Platine in Gehäuse Gehäuse montieren Gehäuse befestigen

Abbildung 4 zeigt einen Export des in SolidEdge ST8 entworfenen Hauptgehäuses des Fahrradlichts. Die drei mittig angeordneten Löcher sind Aussparungen für die drei LEDs. Das einzelne Loch darunter soll später einen kleinen Druck-Knopf beherbergen, mit dem sich das Fahrradlicht in verschiedene Modi schalten lassen soll. Das Hauptgehäuse besteht aus zwei trennbaren Einzelteilen, damit ein Austausch der Batterien oder Reparaturarbeiten an der Platine gewährleistet werden können. Über Schraubverbindungen werden diese zwei Teile kraftschlüssig miteinander verbunden.

4 Zusammenfassung

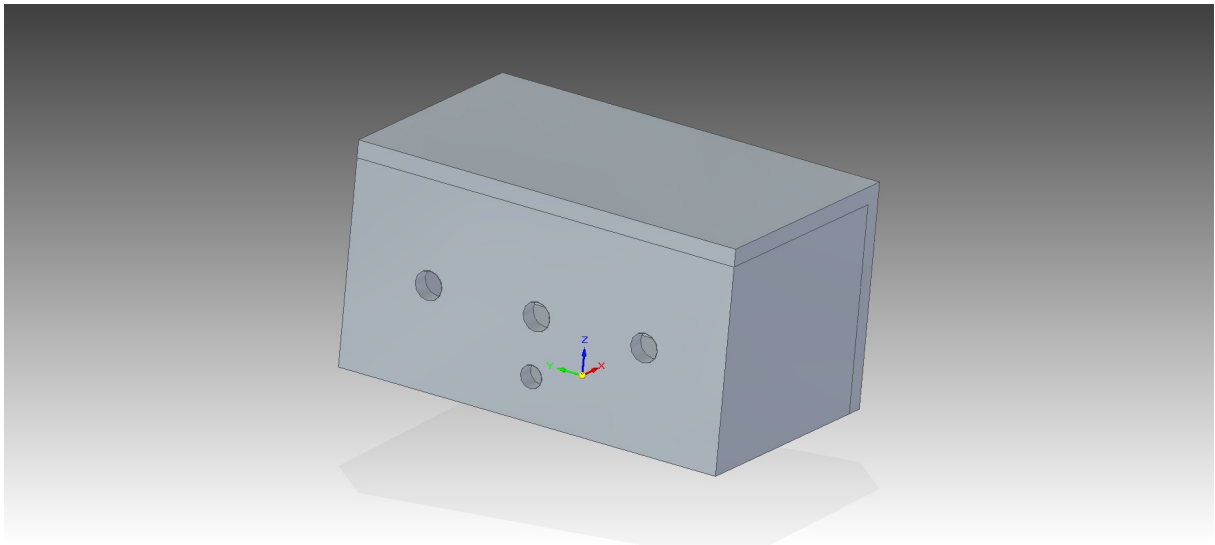


Abbildung 4: 3D-Modell des Hauptgehäuses