



Westsächsische Hochschule Zwickau
University of Applied Sciences

Fakultät Elektrotechnik

Entwicklung einer Ansteuerelektronik für Hochleistungs- LEDs

Als Diplomarbeit zur Erlangung des
akademischen Grades

Diplom-Ingenieur für Elektrotechnik (FH)

eingereicht von

Dietzsch, Stefan

geb. am 19.09.79

Betreuer: Prof. Dr.-Ing. Würfel

September 2011

Versicherung an Eides statt

gemäß SächsHG zur Anfertigung der Diplomarbeit mit dem Thema
Entwicklung einer Ansteuerelektronik für Hochleistungs- LEDs

Dietzsch, Stefan

.....

Name, Vorname

062048

.....

Matrikel

24176

.....

Kennnummer

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht.

Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Zwickau, den 14.09.2011

.....

Stefan Dietzsch

Konkrete Aufgabenstellung

Entwicklung einer Ansteuerelektronik für Hochleistungs- LEDs

Aufgrund der wachsenden Nachfrage nach elektrischer Energie und der steigenden Energiepreise, soll eine energieeffiziente Straßenbeleuchtung entwickelt werden. Dies trägt durch Ressourcenschonung gleichzeitig zum Umweltschutz bei. Die Grundlage hierfür sind Hochleistungs- LEDs, welche in die Leuchte integriert werden. Diese soll sich selbst ein- und ausschalten und je nach Umgebungsbedingung selbst dimmen. Zum Betreiben der LEDs ist die dafür notwendige Ansteuerelektronik zu entwickeln und zu programmieren. Dadurch soll ein um ein Vielfaches niedrigerer Energieverbrauch erzielt werden, als mit herkömmlichen Leuchtmitteln.

- **Recherchen zum Einsatz und Betrieb von Hochleistungs- LEDs**
- **Auswahl eines geeigneten LED-Treibers und der entsprechenden Beschaltung**
- **Schaltungsentwurf und Berechnung aller notwendigen Bauteile**
- **Leiterplattenentwurf und Leiterplattenfertigung**
- **Auswahl eines geeigneten Mikrocontrollers, zur Ansteuerung des Treiberschaltkreises**
- **Softwareentwicklung und Mikrocontrollerprogrammierung in C**
- **Testaufbau und Lichttechnische Messungen, sowie deren Auswertung**
- **Entwurf einer möglichen Leuchte, zum Einsatz im Labor**

Inhaltsverzeichnis

Abbildungsverzeichnis.....	IV
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
Kurzzeichenverzeichnis.....	VIII
Vorbetrachtungen.....	1
1 Einleitung.....	2
2 Stand der Technik.....	3
2.1 Geschichtlicher Hintergrund.....	3
2.2 Aufbau und Arbeitsweise von Leuchtdioden.....	4
2.3 Aufbau und Arbeitsweise der Treiberschaltung	5
2.4 Pulsweitenmodulation.....	6
3 Entwurf und Fertigung einer Platine zur Integration in eine Leuchte.....	11
3.1 Konzept zur Entwicklung einer Leuchte	11
3.2 Bauteilauswahl und Schaltungsentwurf.....	13
3.3 Entwicklung einer Testplatine	14
3.4 Versuche und Messungen an der Testplatine	17
3.5 Entwicklung der Hauptplatine	19
3.6 Signalerfassung mittels Mikrocontroller	20
3.7 Entwicklung und Fertigung der Hauptplatine.....	23
3.8 Kühlkörperberechnung	27
4 Softwareentwicklung und Mikrocontrollerprogrammierung	29
4.1 Entwicklung des Grundprogramms	29
4.2 Entwicklung des Hauptprogramms.....	30
5 Technische Messungen an der Leuchte	41
5.1 Elektrische Größen	41
5.2 Thermische Größen.....	44
5.3 Lichttechnische Größen	49
6 Zusammenfassung und Ausblick.....	53
Literaturverzeichnis	55
Anhang.....	X

Abbildungsverzeichnis

Abbildung 2.1	geschichtliche Entwicklung der LED	3
Abbildung 2.2	Funktionsschema einer LED.....	4
Abbildung 2.3	Schematische Darstellung eines Abwärtswandlers	6
Abbildung 2.4	Aufbau eines Rechtecksignals für PWM.....	7
Abbildung 2.5	Schematische Darstellung der Steuereinheit	7
Abbildung 2.6	Funktionsprinzip des Impulsbreitenmodulators	8
Abbildung 2.7	Einstellen des Tastverhältnisses mittels Compare- Wert.....	9
Abbildung 3.1	Konzept zur Entwicklung einer Leuchte	11
Abbildung 3.2	LED- Treiberschaltung für 10 W LEDs	14
Abbildung 3.3	Platinenlayout der LED- Treiberschaltung a.) und fertige Testplatine b.)	17
Abbildung 3.4	Ausgangsspannung in Abhängigkeit der Eingangsspannung	18
Abbildung 3.5	Ausgangsstrom in Abhängigkeit der Eingangsspannung	19
Abbildung 3.6	Schematische Darstellung der Funktionen des Mikrocontrollers.....	20
Abbildung 3.7	Widerstandsänderung am NTC in Abhängigkeit der Temperatur.....	22
Abbildung 3.8	Widerstandsänderung am NTC in Abhängigkeit der Temperatur.....	23
Abbildung 3.9	Platinenlayout der Unterseite mit SMD- Bauteilen	25
Abbildung 3.10	Platinenlayout der Oberseite mit bedrahteten Bauteilen	26
Abbildung 3.11	Oberseite und Unterseite der endgültigen Platine	27
Abbildung 3.12	Gemischtschaltung der einzelnen Wärmewiderstände	28
Abbildung 4.1	Ausschnitt aus dem Quellcode des Grundprogramms.....	29
Abbildung 4.2	Ausschnitt aus dem Quellcode der Hauptfunktion	30
Abbildung 4.3	Einstellen der Timer/Counter	31
Abbildung 4.4	Initialisieren des ADC	32
Abbildung 4.5	Temperaturüberwachung und Notabschaltung	33
Abbildung 4.6	ADC- Messung	34
Abbildung 4.7	Einschaltfunktion der LEDs	34
Abbildung 4.8	Aufbau der Zeitfunktion innerhalb der ISR.....	35
Abbildung 4.9	Einschalten der LEDs bei Erreichen der Dämmerung.....	36
Abbildung 4.10	Ausschalten der LEDs bei ausreichender Helligkeit	38
Abbildung 4.11	Hochdimmen beim Erkennen einer Bewegung	38
Abbildung 4.12	Ablauf der Wartezeit und Funktionsaufruf.....	39
Abbildung 4.13	Funktion zum Herunterdimmen, auf 12,5 %	39
Abbildung 4.14	Übergabe der verschiedenen Zeiten.....	40

Abbildung 5.1 Ausgangsspannung LED- Treiber 1 a.) und 2 b.), gedimmt auf 12,5 %	42
Abbildung 5.2 Ausgangsspannung LED- Treiber 1 a.) und 2 b.), bei 100 %	43
Abbildung 5.3 Spannungsverlauf an LED 1 a.) und LED 3 b.).....	43
Abbildung 5.4 Spannung am PWM- Ausgang des Mikrocontrollers, a.) 12,5 %, b.) 100 % ..	44
Abbildung 5.5 Temperaturverläufe an der Leuchte	45
Abbildung 5.6 Verteilung der Temperatursensoren	46
Abbildung 5.7 Temperaturverläufe von Kühlblech und MOSFET	47
Abbildung 5.8 Temperaturverläufe nach einer Stunde Erwärmung	48
Abbildung 5.9 Leuchtdichte bei 100 % und auf 12,5 % gedimmt.....	49
Abbildung 5.10 Gemessene Leuchtdichte bei 100 % und bei 12,5 %	49
Abbildung 5.11 Leuchtdichteskala	49
Abbildung 5.12 Messaufbau zur Leuchtstärkemessung	50
Abbildung 5.13 Beleuchtungsstärke im gedimmten Betrieb	51
Abbildung 5.14 Beleuchtungsstärke bei 100 % Leistung.....	52

Tabellenverzeichnis

Tabelle 3.1	Parameter des Buck Converters LM3409 HV	13
Tabelle 3.2	Parameter der verwendeten High- Power LEDs.....	14
Tabelle 3.3	Bauelemente zur Beschaltung des Buck Converters	15
Tabelle 4.1	Einstellen verschiedener Zeiten	40
Tabelle 5.1	Leistungsaufnahme	41

Abkürzungsverzeichnis

Abkürzung	Bezeichnung
Abb	Abbildung
AD	analog - digital
ADC	Analog - Digital - Converter
CPU	central processing unit
DSO	Digital-Speicher-Oszilloskop
EAGLE	Einfach Anzuwendender Grafischer Layout-Editor
EMV	Elektromagnetische Verträglichkeit
IC	Integrated circuit
I/O	In/Out
ISP	In-System Programming
ISR	Interrupt Service Routine
LED	Lichtemittierende Diode
MOSFET	Metal-Oxide-Semiconductor Field-Effect-Transistor
NTC	Negative Temperature Coefficient
OPV	Operationsverstärker
PWM	Pulsweitenmodulation
SMD	surface-mounted device

Kurzzeichenverzeichnis

Vormelzeichen	Bedeutung	Einheit
$\Delta \vartheta$	Temperaturänderung	K
η_{LED}	Wirkungsgrad der LED	%
η_{Buck}	Wirkungsgrad des Abwärtswandlers	%
$\eta_{Netzteil}$	Wirkungsgrad des Netzteils	%
Φ_{LED}	Lichtstrom der LED	lm
A	Kühlfläche	m ²
C1	Ausgangskapazität	F
C_{in1}, C_{in2}	Eingangskapazitäten	F
C_F	Kapazität zur Erzeugung des Ansteuerstromes für MOSFET	F
C_{off}	Kondensator zur Auszeitbestimmung	F
D1	Schottky Diode am Ausgang	
I_{ges}	Gesamtstrom der LED-Steuerung	A
I_{LED}	Strom durch die LED	A
I_{70}	Strom durch den NTC bei 70 °C	A
$I_{Netzteil}$	Strom durch das Netzteil	A
L1	Induktivität zur Aufrechterhaltung des Stromes am Ausgang	H
f_{PWM}	Frequenz der Pulsweitenmodulation	Hz
f_{CPU}	Taktfrequenz des Mikrocontrollers	Hz
Q1	Leistungs transistor am Abwärtswandler	
P_{LED}	Leistungsabgabe der LEDs	W
P_{ges}	gesamte Leistung	W
$P_{Netzteil}$	Leistungsaufnahme am Netzteil	W
P_{tot}	Verlustleistung	W
R_{NTC}	Widerstandswert des NTC-Widerstandes	Ω
R_{mess}	Messwiderstand zur Temperaturmessung	Ω
R_m	Widerstand bei mittlerer Temperatur (50 °C)	Ω
R_o	Widerstand bei oberer Temperatur (75 °C)	Ω
R_u	Widerstand bei unterer Temperatur (25 °C)	Ω
R_{off}	Widerstand zur Auszeitbestimmung	Ω
R_{SNS}	Messwiderstand	Ω

R_{thJC}	Wärmewiderstand Junction/Case	Ω
R_{thCH}	Wärmewiderstand Case/Heatsink	Ω
R_{thHA}	Wärmewiderstand Heatsink/Ambient	Ω
R_{UV1}	Widerstand zur Unterspannungskontrolle	Ω
R_{UV2}	Widerstand für Hysterese	Ω
T	Periodendauer	s
t_{ein}	Einschaltzeit	s
t_{aus}	Ausschaltzeit	s
t_{ges}	gesamnte Zeit zum hoch- und herunterzählen	s
U_A	Ausgangsspannung	V
U_E	Eingangsspannung	V
$U_{Netzteil}$	Ausgangsspannung am Netzteil	V
U_{NTC}	Spannung am NTC	V
U_{reff}	Referenzspannung	V
U_{St}	Steuerspannung	V
U_{SZ}	Sägezahnspannung	V
V_{in}	Eingangsspannung	V
V_{out}	Ausgangsspannung	V
GND	Ground	

Vorbetrachtungen

Der Energieverbrauch für die Straßenbeleuchtung von mittelgroßen Städten hat einen Anteil von etwa 50 bis 60 % des gesamten Elektroenergiebedarfes der öffentlichen Verwaltung. Somit ist mindestens die Hälfte der Stromkosten einer Kommune der Straßenbeleuchtung geschuldet. Des Weiteren werden in den nächsten Jahren EG-Richtlinien in Kraft treten, welche eine Umgestaltung der Straßenbeleuchtung unumgänglich machen.

Jetzt ist der richtige Zeitpunkt Straßenbeleuchtungen umzustellen auf energiesparende Leuchtmittel, wie z.B. LEDs. Mit LEDs würden Straßenlampen nur ein Viertel der Energie verbrauchen, welche die häufig verwendeten Natriumdampflampen benötigen. Außerdem liegt die Lebensdauer einer LED-Straßenlampe um ein Vielfaches höher (ca. 50000 Betriebsstunden). Bei einer jährlich durchschnittlichen Betriebsdauer von ca. 13 Stunden pro Tag, ergibt dies im mitteleuropäischen Raum eine Verwendungsdauer von min.10 Jahren. Dies ist ein weiterer Aspekt der für den Einsatz der LED-Technologie spricht. (vgl. [18])

1 Einleitung

Seit einigen hundert Jahren werden Gehwege, Straßen und Plätze nachts beleuchtet, um Unfälle zu vermeiden. Aus diesem Grund werden Außenbeleuchtungen immer weiterentwickelt und auch nie wegzudenken sein. Ziel dieser Arbeit ist daher die Entwicklung einer Ansteuerelektronik für Straßenbeleuchtungen mit Hochleistungs-LEDs. Es gibt bereits einige Anwendungen für Hochleistungs-LEDs, angefangen bei Fahrradbeleuchtungen bis hin zu Scheinwerfern für Kraftfahrzeuge, außerdem Beleuchtungsmöglichkeiten für Innen und Außen. Im Bereich der Außenbeleuchtung wurden bereits erste Tests mit Straßenlampen durchgeführt. Der Entwicklungsprozess solcher Straßenlampen befindet sich allerdings im Anfangsstadium, so dass hier noch Anpassungen zur Effizienzsteigerung möglich sind. Für solche Anwendungen wurden bisher immer viele LEDs mit geringen Leistungen von 1...2 W verwendet.

Grundlage dieser Anwendung sollen vier leistungsstarke LEDs sein, die im Vergleich zu leistungsschwächeren, aber auch leistungsstärkeren LEDs das beste Preis/Leistungsverhältnis bieten. Im Vordergrund stand zunächst die Ansteuerelektronik zum Betreiben der LEDs. Diese ist notwendig, um einen konstanten Strom zu gewährleisten und Verluste gering zu halten. Des Weiteren soll es diese Steuerung ermöglichen, auf verschiedene Signale von Außen zu reagieren. Ein weiterer Aspekt war das Dimmen der LEDs auf eine gewünschte Helligkeit. Mit dieser Ansteuerelektronik soll die Lebensdauer und die Helligkeit der LEDs optimal ausgenutzt werden. Ein LED-Treiberschaltkreis würde für diese Anwendung in Betracht kommen, da ein solcher Schaltkreis in seinen Eigenschaften und mit der richtigen Beschaltung genau den Anforderungen gerecht wird, die an eine solche LED-Steuerung gestellt werden. Basierend auf der Grundlage eines Treiberschaltkreises wird ein Schaltplan entwickelt. Im Anschluss daran wird mit einem grafikfähigen Layoutprogramm ein Leiterplattenlayout entworfen, was zur Beschaltung des Treiberschaltkreises dient. Mit dem fertigen Layout kann dann eine Platine für die Ansteuerelektronik gefertigt werden. Der letzte Schritt zur Entwicklung einer kompletten Ansteuerelektronik ist die Auswahl eines geeigneten Mikrocontrollers und dessen Programmierung.

2 Stand der Technik

2.1 Geschichtlicher Hintergrund

Das Phänomen der Lichterzeugung mittels elektrischer Anregung eines Festkörpers entdeckte erstmals der Forscher H. J. Round im Jahre 1907. Grundlage für diese Entdeckung war ein Siliziumkarbid-Kristall, der zur Dedektion von Radiowellen bestimmt war. Hierbei wurde damals schon erkannt, dass es sich um kaltes Licht handelt, da die Emission ohne erkennbare Erwärmung des Kristalls erfolgte. Diese Beobachtung wurde zunächst nicht beachtet, da sich die Forscher in dieser Zeit nur für Radiodetektoren interessierten. Erst 1921 entdeckte der Forscher O. V. Lossew diese Lichtemission erneut. O. V. Lossew untersuchte dieses Phänomen in den Jahren 1927 und 1942 genauer, bis er die neuartige Lichtquelle mit einem Drehspiegel modulieren konnte. Diese Errungenschaft wollte Lossew zunächst für die Nachrichtenübertragung einsetzen. Es dauerte lange, bis sich seine Beobachtung in das theoretische Schema der Festkörperwissenschaft einordnen ließ. Erst 1951 konnte diese Lichtemission ausreichend erklärt werden. Hierzu waren das gesamte Wissen über den Transistor und der eingeleitete wissenschaftliche Fortschritt in der Halbleiterphysik erforderlich.

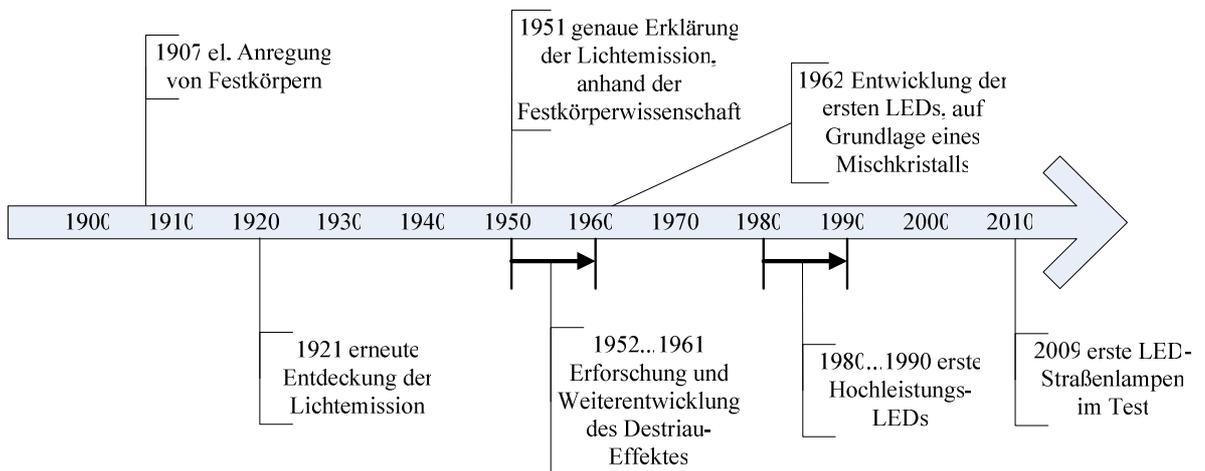


Abbildung 2.1 geschichtliche Entwicklung der LED

Von 1952 bis 1961 setzte zuerst die Erforschung und Weiterentwicklung des Destriau-Effektes ein, womit unter Verwendung von ZnS-Pulverphosphoren flache Bildschirme entwickelt werden sollten, um die Kathodenstrahlröhre zu ersetzen. Während der entsprechende Erfolg mit Zinksulfid (ZnS) ausblieb, brachten die 1952 als Halbleiter erkannten III-V-Verbindungen den erhofften Durchbruch. (vgl. [16])

Hierbei handelt es sich um eine Verbindung von Materialien der 3. und 5. Hauptgruppe, welche dadurch die Eigenschaften von Halbleitern erhalten. Etwa 1957 wurde mit intensiven grundsätzlichen Untersuchungen der Lichterzeugung mit den neuen Halbleitern begonnen. Somit begann auch die Entwicklung einer geeigneten Technologie zur Herstellung von Kristallen und Bauelementen. Von besonderer Bedeutung war die Lichtemission im sichtbaren Bereich auf der Basis eines direkten Mischkristalls aus Galliumarsenid und Galliumphosphid. Erst mit dieser Arbeit kam, 55 Jahre nach der ersten Entdeckung von H. J. Round, die LED-Entwicklung voll in Gange. (vgl. [17])

2.2 Aufbau und Arbeitsweise von Leuchtdioden

Eine LED besteht größten Teils aus Halbleitermaterial. Durch Anlegen einer äußeren Spannung in Durchlassrichtung wandern Elektronen zur Rekombinationsschicht am p-n-Übergang. Auf der n-dotierten Seite füllen diese Elektronen das Leitungsband. Nach Überschreiten der Grenzfläche wechseln diese auf das energetisch günstigere p-dotierte Valenzband und rekombinieren mit den dort vorhandenen (Elektronenmangel) Löchern. Ein Funktionsschema ist in Abb. 2.2 zu sehen.

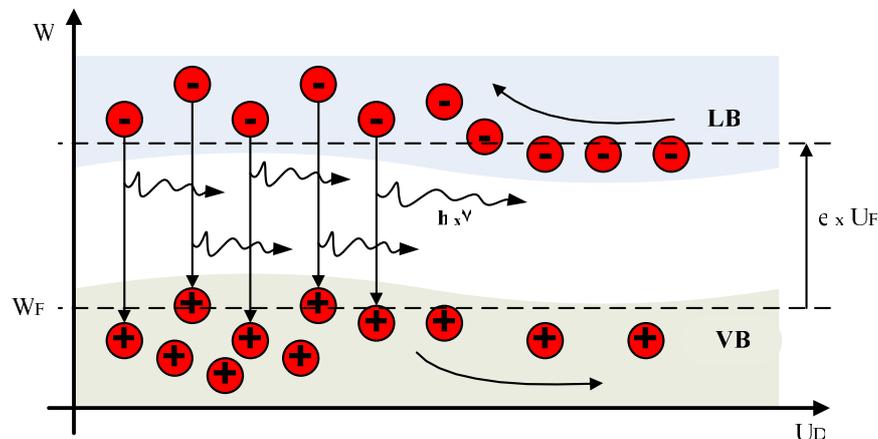


Abbildung 2.2 Funktionsschema einer LED

Bei Silizium-Dioden beispielsweise erfolgt der Übergang strahlungslos durch Phononenanregung, indem das Gitter den Impuls der Teilchen aufnimmt. Bei Gallium-Arsenid (GaAs) z.B. geht der direkte Übergang mit der Aussendung eines Photons einher. Das Verhalten der Energieübertragung wird durch die Bandstruktur des Halbleiters bestimmt. Halbleitermaterialien mit Photonenwechselwirkung und direktem Übergang, wie Gallium-Arsenid, werden in der Literatur als direkter Halbleiter bezeichnet. (vgl. [17])

Anders als Halbleitermaterialien mit Phononenwechselwirkung und indirektem Übergang, wie bei Silizium (Si), diese werden als indirekter Halbleiter bezeichnet. Hierbei handelt es sich um eine Gitterschwingung die zur Erwärmung des Halbleiters führt. Allerdings kommen als Werkstoff für LEDs nur direkte Halbleitermaterialien zur Anwendung.

Eine Wechselspannung ist zum Betreiben von LEDs nicht von Vorteil, da die Elektronen nur in Durchlassrichtung wandern und dabei Licht emittieren. Es würde zwangsläufig ein Flackern entstehen, welches von der Frequenz abhängig wäre, da die LED bei jeder positiven Halbwelle ein und bei jeder negativen Halbwelle wieder aus wäre. Dies würde außerdem die Lebensdauer einer LED verkürzen. Da bei Gleichspannung die Elektronen kontinuierlich wandern, leuchtet die LED somit permanent. Es gibt einige LED-Anwendungen, die ohne Regelelektronik mit Wechselspannung funktionieren, allerdings wird bei vielen anderen LED-Applikationen, denen kein Gleichrichter vorgeschaltet ist, die Steuerelektronik zerstört.

Aus Kostengründen wird für Beleuchtungszwecke fast immer die Variante einer Leuchtdiode in Kombination mit Leuchtstoffen verwendet. Beispielsweise bestehen weiße LEDs aus einer blauen LED und einer darüber liegenden gelblich fluoreszierenden Schicht aus Cer-dotiertem Yttrium-Aluminium-Granat- Pulver. Da bei blauen LEDs der Ultraviolett-Anteil, welcher am kurzwelligen Ausläufer ihres Strahlungsspektrums ausgesendet wird, ebenfalls durch die Fluoreszenzschicht weitgehend in gelbliches Licht umgewandelt wird, besitzen sie den höchsten Wirkungsgrad. Aus diesem Grund ist das die wirtschaftlichste Methode, weißes Licht per LED zu erzeugen. (vgl.[17])

2.3 Aufbau und Arbeitsweise der Treiberschaltung

Zum Betreiben von Hochleistungs- LEDs sind eine entsprechende Gleichspannung und ein konstanter Strom notwendig. In diesem Fall wird dies durch einen Abwärtswandler, oder auch Buck Converter, und die entsprechende Beschaltung gewährleistet.

Ein Abwärtswandler ist ein Spannungswandler, der eine hohe Gleichspannung in eine niedrigere Gleichspannung definierter Größe umwandelt. Die Spannung am Ausgang ist dabei abhängig von der Last und wird durch einen konstanten Strom festgelegt. Dieser Strom wird durch eine entsprechend dimensionierte Induktivität und einen Ausgangswiderstand definiert. Ein Abwärtswandler besteht im einfachsten Fall aus einer Spule, einem Kondensator, einer Diode und einem Transistor. In Abb. 2.3 ist das Schema eines Abwärtswandlers dargestellt.

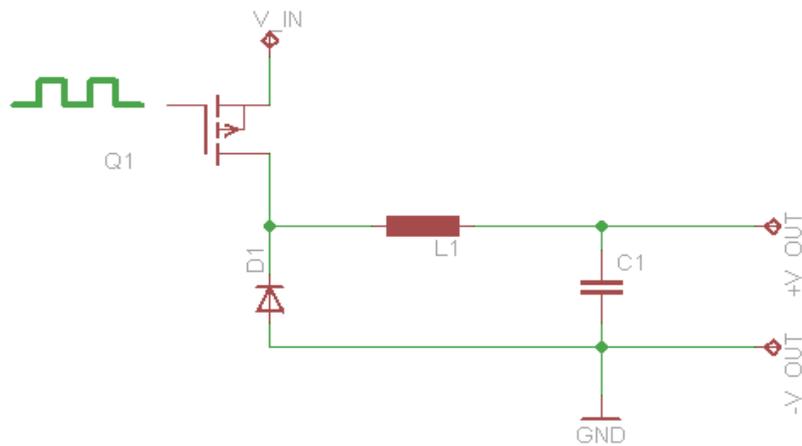


Abbildung 2.3 Schematische Darstellung eines Abwärtswandlers

Der Transistor Q1 dient hierbei als Schalter, welcher durch geregeltes Ein- und Ausschalten die Höhe der Ausgangsspannung regelt. Im eingeschalteten Zustand fließt ein Ladestrom durch die Induktivität L1, der den Kondensator C1 lädt. Dabei steigt der Strom durch die Spule und die Last stetig an. Die Diode D1 ist in diesem Zeitraum gesperrt. Im ausgeschalteten Zustand (Transistor sperrt) entlädt sich die in der Spule gespeicherte Energie über die Diode und die Last. Hierbei wird der Strom solange aufrecht gehalten, bis die Induktivität entladen ist. Der Kondensator entlädt sich ebenfalls teilweise über die Last. Sobald der Transistor wieder in den leitenden Zustand übergeht, wiederholt sich der Vorgang und es fließt wieder ein Ladestrom. Wird der Transistor periodisch ein- und ausgeschaltet, wird sich das System in einen stationären Zustand begeben. Die Spule garantiert, dass der Strom auch im ausgeschalteten Zustand, für eine bestimmte Zeit, weiter fließt und der Kondensator glättet die Ausgangsspannung. Diese Zeit ist von der Induktivität der Spule abhängig. Somit wird ein nicht lückender Strom am Ausgang bereitgestellt und die Ausgangsspannung ist über das Tastverhältnis zwischen EIN und AUS am Transistor einstellbar. Schaltkreise dieser Art erreichen einen Wirkungsgrad von 80 % bis 95 % und lassen sich mittels eines PWM-Signals ansteuern. (vgl.[12])

2.4 Pulsweitenmodulation

Bei der Pulsweitenmodulation oder auch Pulsbreitenmodulation wird ein Rechtecksignal generiert, welches in seiner Frequenz gleich bleibt, aber in seinem Tastverhältnis verändert wird. Beim Tastverhältnis handelt es sich um das Verhältnis zwischen Einschaltzeit t_{ein} und Periodendauer T , was in Abbildung 2.4 noch einmal dargestellt ist.

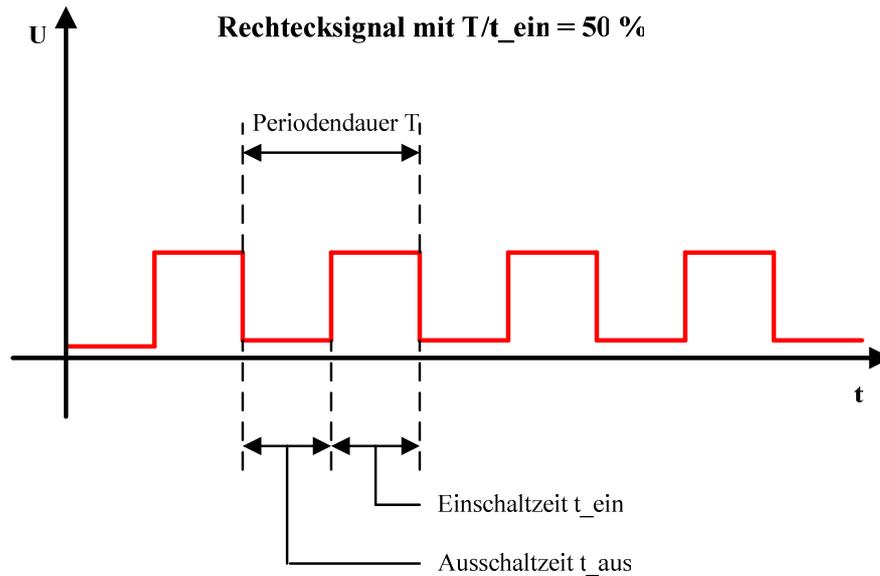


Abbildung 2.4 Aufbau eines Rechtecksignals für PWM

Grundsätzlich gibt es zwei Arten von Pulsweitenmodulation (PWM). Zum einen die analoge PWM und zum anderen die digitale PWM. Auf beide soll nachfolgend noch einmal näher eingegangen werden. (vgl.[1, 2, 3])

2.4.1 Analoge Pulsweitenmodulation

Eine analoge PWM ist theoretisch in der Lage unendlich viele Tastverhältnisse und somit eine unendlich hohe Auflösung zu generieren. Dabei verhält sich die Einschaltzeit eines Spannungswandlers proportional zur Ausgangsspannung. Das Schaltsignal wird mittels zwei Modulen erzeugt, einem Regler und einem Impulsbreitenmodulator. Dieser Modulator besteht wiederum aus einem Komparator und einem Sägezahn-generator. Ein Blockschaltbild zur Erzeugung des Schaltsignals ist in Abb. 2.5 dargestellt.

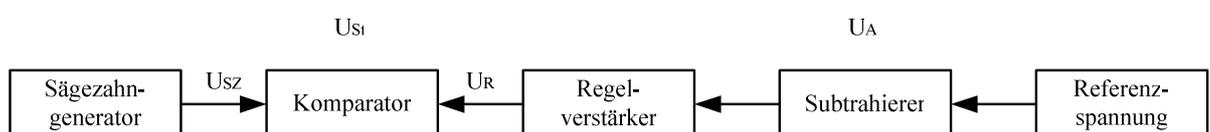


Abbildung 2.5 Schematische Darstellung der Steuereinheit

Solange die verstärkte Referenzspannung U_R größer als die erzeugte Sägezahnspannung U_{SZ} ist, schaltet der Komparator ein. Aus der daraus entstehenden Steuerspannung geht folgendes Tastverhältnis D hervor. $D = U_R/U_{SZ}$ Durch einen Subtrahierer wird die Differenz zwischen Referenzspannung und Ausgangsspannung gebildet. $U_{ref} - k \cdot U_A$

Die Referenzspannung wird vom Regelverstärker solange erhöht, bis diese Differenz Null wird. Der Wert der Ausgangsspannung ergibt sich dann aus $U_A = U_{ref}/k$. (vgl.[2]) Das Funktionsprinzip des Impulsbreitenmodulators ist in Abb. 2.6 dargestellt. (vgl.[1, 2, 3])

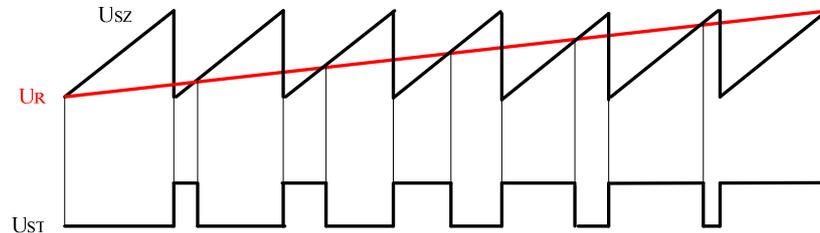


Abbildung 2.6 Funktionsprinzip des Impulsbreitenmodulators

2.4.2 Digitale Pulsweitenmodulation

Die PWM-Frequenz kann maximal die CPU-Frequenz geteilt durch Timerauflösung sein. Zudem ist entscheidend, ob der Modus Fast-PWM oder phasenkorrekte PWM gewählt wird. Allerdings kann diese PWM-Frequenz noch durch einen Vorteiler, dem so genannten Prescaler, verkleinert werden. Die resultierende Frequenz ist somit von der Auflösung des verwendeten Timers und vom eingestellten Prescaler abhängig, z.B. Timerauflösung 8 Bit $\rightarrow (2^8-1)$ und Prescaler max. 1024.

$$f_{\text{PWM}} = \frac{f_{\text{CPU}}}{2 \cdot \text{Prescaler} \cdot (2^8 - 1)} \quad [1]$$

$$f_{\text{PWM}} = \frac{20\text{MHz}}{2 \cdot 1024 \cdot 255} = 38,3\text{Hz}$$

Dies bedeutet bei einer CPU-Frequenz von 20 MHz und phasenkorrekter PWM könnte minimal eine PWM-Frequenz von 38,3 kHz eingestellt werden.

Der Timer dient hierbei als Zähler, der kontinuierlich von 0 bis zur eingestellten Auflösung (255) zählt und danach wieder zurück bis 0. Eine Periode wäre beispielsweise, einmal hoch und wieder runter zählen. Über das so genannte Compare Register wird das Tastverhältnis eingestellt, was in Abbildung 2.7 noch einmal graphisch dargestellt ist. (vgl.[12])

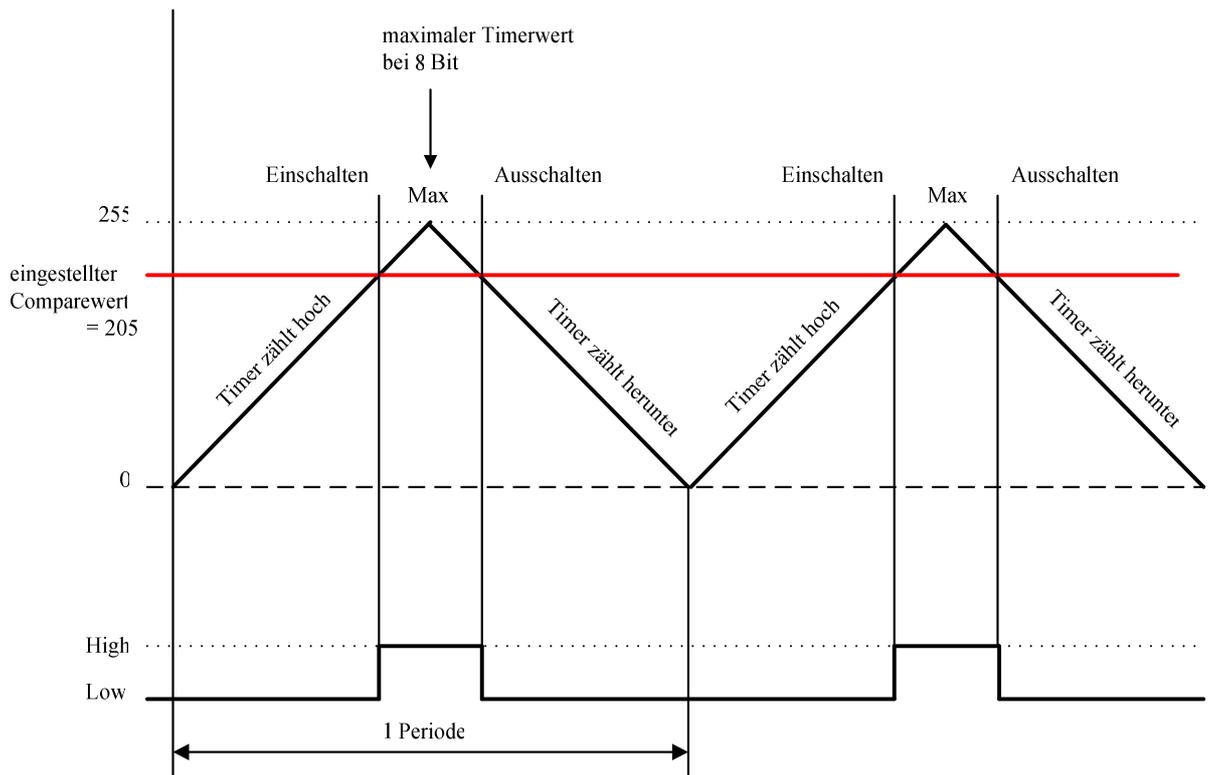


Abbildung 2.7 Einstellen des Tastverhältnisses mittels Compare-Wert

Bei Übereinstimmung von Timer-Wert und Compare-Wert wird der Ausgang geschaltet. Beim Hochzählen wird auf EIN (High) und beim Herunterzählen auf AUS (Low) geschaltet. Im dargestellten Beispiel beträgt das Tastverhältnis lediglich 20 %. Wäre der Compare-Wert viel kleiner, wäre die Dauer des High-Signals entsprechend höher.

Es gibt zwei Arten von PWM, die Fast-PWM und die phasenkorrekte PWM. Bei der oben dargestellten Variante handelt es sich um die phasenkorrekte PWM. Bei der Fast-PWM werden die Pulse durch ein Sägezahnsignal erzeugt, wodurch dieser Vorgang verkürzt wird. In diesem Fall zählt der Timer, von Null beginnend, bis zum eingestellten Maximalwert, erzeugt einen Überlauf und wird wieder auf Null gesetzt. Anschließend beginnt dieser Zählvorgang von Neuem. Anders als bei der phasenkorrekten PWM wird hier beim Hochzählen der Ausgang auf AUS (Low) geschaltet, sobald der Timer-Wert mit dem Compare-Wert übereinstimmt. Sobald der Timer seinen Maximalwert erreicht, wird der Ausgang auf EIN (High) geschaltet und der Timer wird wieder auf Null zurückgesetzt. Dies ermöglicht eine schnellere PWM-Erzeugung, da der Timer in jeder Periode nur einmal zählt. (vgl.[12])

Eine Ansteuerung mittels Pulsweitenmodulation findet in der Praxis immer häufiger Anwendung, z.B. zur Drehzahlregelung von Elektromotoren, zur Temperaturregelung mittels Heizwiderstand oder für Beleuchtungszwecke. Letzteres wird besonders bei LEDs angewendet, um diese auf eine bestimmte Helligkeit zu dimmen.

Da eine LED immer mit einem konstanten Strom betrieben werden sollte, ist diese Methode besonders von Vorteil. Denn bei einer PWM ist die Höhe des gepulsten Signals immer gleich und somit auch der Stromfluss im gediminten Betrieb konstant. Die Spannung am Ausgang lässt sich über das Tastverhältnis berechnen. Bei einem Tastverhältnis von 20 % und einer Frequenz von 39,2 kHz würde sich diese nach Formel [2] und [3] berechnen lassen.

$$\Lambda \vartheta \quad [2]$$

$$t_{\text{ges}} = \frac{1}{39215\text{Hz}} = 25,5\mu\text{s}$$

$$U_{\text{A}} = U_{\text{E}} \cdot \frac{t_{\text{ein}}}{T} \quad [3]$$

$$U_{\text{A}} = 14\text{V} \cdot \frac{5,1\mu\text{s}}{25,5\mu\text{s}} = 2,8\text{V}$$

Genauso lässt sich auch die aufgenommene Leistung einer gediminten LED berechnen, wie in Formel [4] dargestellt.

$$P_{\text{ges}} = P_{\text{LED}} \cdot \frac{t_{\text{ein}}}{t_{\text{ges}}} \quad [4]$$

$$P_{\text{ges}} = 10\text{W} \cdot \frac{5,1\mu\text{s}}{25,5\mu\text{s}} = 2\text{W}$$

Dies macht deutlich, dass eine Ansteuerung mittels PWM ebenfalls stark zu Energieeffizienz beiträgt.

3 Entwurf und Fertigung einer Platine zur Integration in eine Leuchte

3.1 Konzept zur Entwicklung einer Leuchte

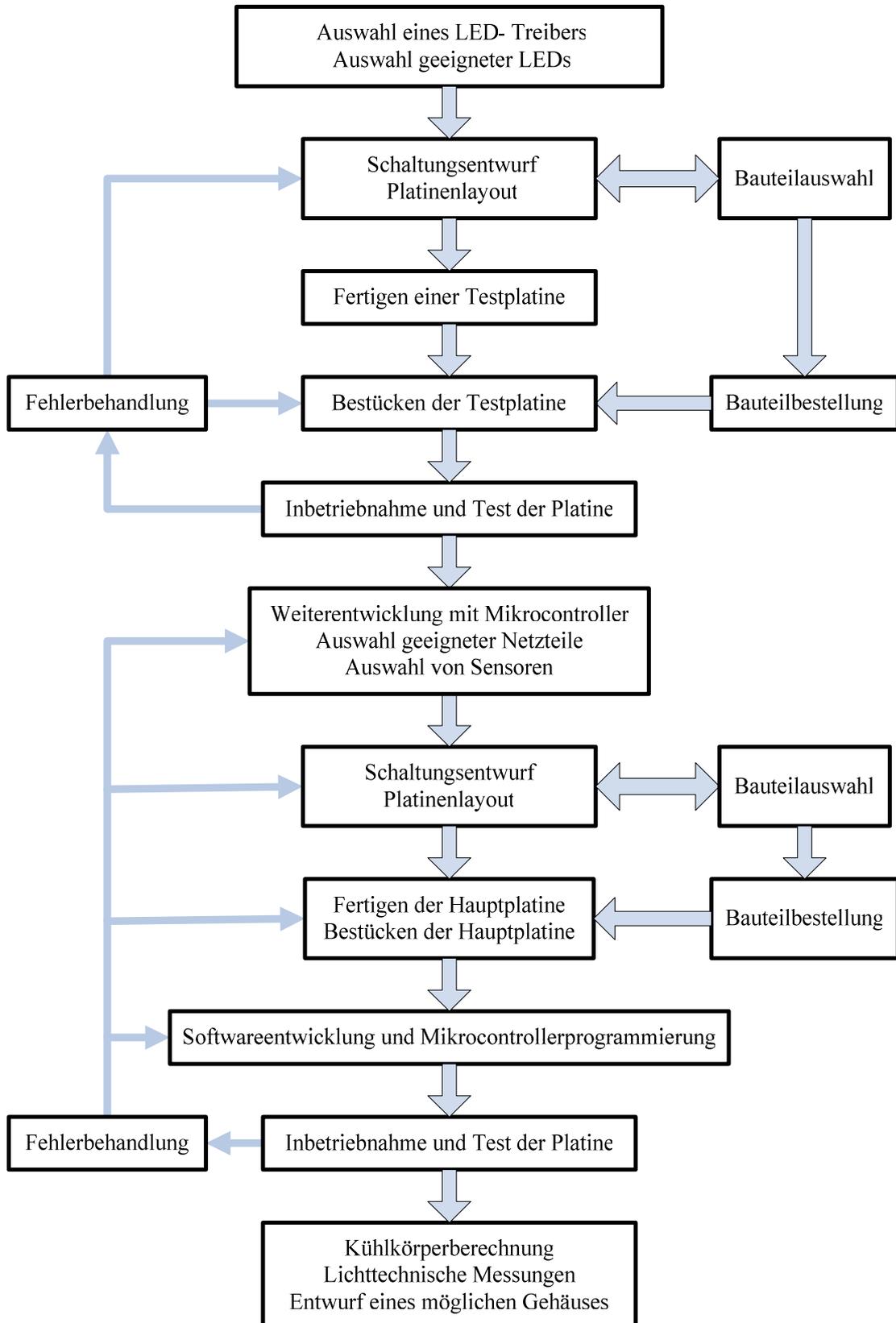


Abbildung 3.1 Konzept zur Entwicklung einer Leuchte

3.1.1 Erläuterungen zum Konzept

Um eine vollwertige Ansteuerelektronik entwickeln zu können, welche in eine Leuchte integriert werden kann, wurde ein Konzept aufgestellt. Nach einigen Recherchen geht es im ersten Schritt um die Auswahl geeigneter LEDs und eines dazu passenden LED-Treibers. Basierend auf dieser Grundlage kann mit einem beliebigen Layoutprogramm eine Schaltung entworfen werden, was ein Platinenlayout mit einbezieht. Die Auswahl der zu verwendeten Bauteile erfolgt parallel zum Schaltungsentwurf, da deren Baugröße entscheidend beim Layout ist. In dieser Arbeit wurden alle Schaltungen mit dem grafischen Layoutprogramm EAGLE-5.6.0 entworfen. Mit einem fertigen Layout kann dann eine Platine belichtet und geätzt werden. Danach wird diese Platine mit den bestellten Bauteilen bestückt und anschließend in Betrieb genommen. Treten bei einem ersten Test Fehler auf, so müssen diese gesucht und behoben werden. Fehler können einerseits beim Bestücken der Platine, durch unsauberes Löten, oder schon beim Schaltungsentwurf auftreten. Nach einem fehlerfreien Funktionstest sind Weiterentwicklungen an der Platine möglich. Diese Weiterentwicklungen bestehen darin, die LED- Treiberschaltung über einen Mikrocontroller anzusteuern. Hierfür sind zwei Netzteile notwendig, eines für die Eingangsspannung der LED-Treiber und ein Weiteres für die Betriebsspannung des Mikrocontrollers. Des Weiteren sind verschiedene Sensoren notwendig, die dem Mikrocontroller bestimmte Signale übermitteln. Nachdem diese Komponenten ausgewählt sind werden eine neue Schaltung und ein neues Platinenlayout entworfen. Im Zuge dessen müssen neue und zusätzliche alte Bauteile ausgewählt und bestellt werden. Daraufhin wird eine neue Platine gefertigt und bestückt. Vor der Inbetriebnahme der fertigen Platine muss die Softwareentwicklung für die gewünschten Anforderungen und anschließend die Programmierung des Mikrocontrollers erfolgen. Nach erfolgreichem Übertragen des Programms in den Controller kann die Platine getestet werden. Sollten beim Test Fehler auftreten, sind diese entweder auf eine fehlerhafte Software, unsauberes Löten, Layoutfehler oder die Auswahl falscher Sensoren zurückzuführen. Sollte einer oder mehrere dieser Fehler auftreten, müssen diese beseitigt werden, bevor die Elektronik zur Integration in eine Leuchte freigegeben wird. Zum Einstellen bestimmter Parameter und zum Abgleichen der Software sind verschiedene Tests notwendig. Im Anschluss daran wird eine Kühlkörperberechnung durchgeführt, um die LEDs im Dauerbetrieb ausreichend zu kühlen. Darauf folgen lichttechnische Messungen, welche Aufschluss über die Leistungen der Ansteuerelektronik und damit der LEDs geben. Zu Demonstrationszwecken wird zum Schluss der Entwurf eines möglichen Gehäuses vorgestellt.

3.2 BauteilAuswahl und Schaltungsentwurf

Für diesen Anwendungsfall sind fertige Treiberschaltkreise bei vielen Herstellern käuflich zu erwerben. Für den Aufbau dieser Platine wurde ein Schaltkreis der Firma *National Semiconductor*, mit der Bezeichnung LM3409 HV verwendet. Hierbei handelt es sich um einen Abwärtswandler. Für diesen wurde sich entschieden, da jener für den Anwendungsfall die gewünschten Parameter aufweist und zudem noch preiswert ist. Die Parameter des gewählten Abwärtswandlers sind in der Tabelle 3.1 ersichtlich. (vgl.[10])

Tabelle 3.1 Parameter des Buck Converters LM3409 HV

Parameter	Größe
Funktionsprinzip	Abwärtswandler
Dimming Controltyp	PWM
Treiber	IC- Typ
Anschlussart	SMD
Anzahl der Anschluss- Pins	10
Eingangsspannung	6 V...75 V
max. Ausgangsstrom	5 A
Betriebstemperaturbereich	-40 °C...+125 °C
Wirkungsgrad	95 %

Des Weiteren besteht die Möglichkeit diesen Abwärtswandler über eine PWM anzusteuern und somit die Last am Ausgang zu dimmen. Ein Buck Converter kann am Ausgang maximal die Spannung realisieren, die am Eingang zur Verfügung steht. Da die verwendeten LEDs eine Betriebsspannung von 14 V haben und zwei LEDs in Reihe betrieben werden sollen, wurde sich für eine Eingangsspannung von 36 V entschieden. Somit kann eine stabile Ausgangsspannung von 28 V bereitgestellt werden. Weiterhin benötigen diese LEDs einen konstanten Strom von 700 mA, welcher vom Controller und aufgrund der Beschaltung konstant bereitgestellt wird. Die Parameter der verwendeten High- Power LEDs sind in Tabelle 3.2 aufgeführt.

Tabelle 3.2 Parameter der verwendeten High- Power LEDs

Parameter	Größe
Technologie	GaInN/GaN
Farbe	cool- white
Wellenlänge	8000 K
Lichtstärke	820 lm
Abstrahlwinkel	120 °
Betriebsspannung	14 V
Betriebsstrom	700 mA
Leistung	10 W

3.3 Entwicklung einer Testplatine

Die Bauelemente zur Beschaltung des LM3409 wurden nach einem Beschaltungsbeispiel aus dem Datenblatt des LM3409 ausgewählt. Die Dimensionierung dieser Bauelemente erfolgte anhand im Datenblatt angegebener Berechnungsvorschriften, um die gewünschten Werten zu realisieren. Die gewählte Treiberschaltung und die ausgewählten Bauelemente sind in Abb. 3.2 und Tabelle 3.3 aufgeführt. Die Formeln zur Berechnung der Bauelemente befinden sich im Anhang, unter A1 Berechnungsformeln.

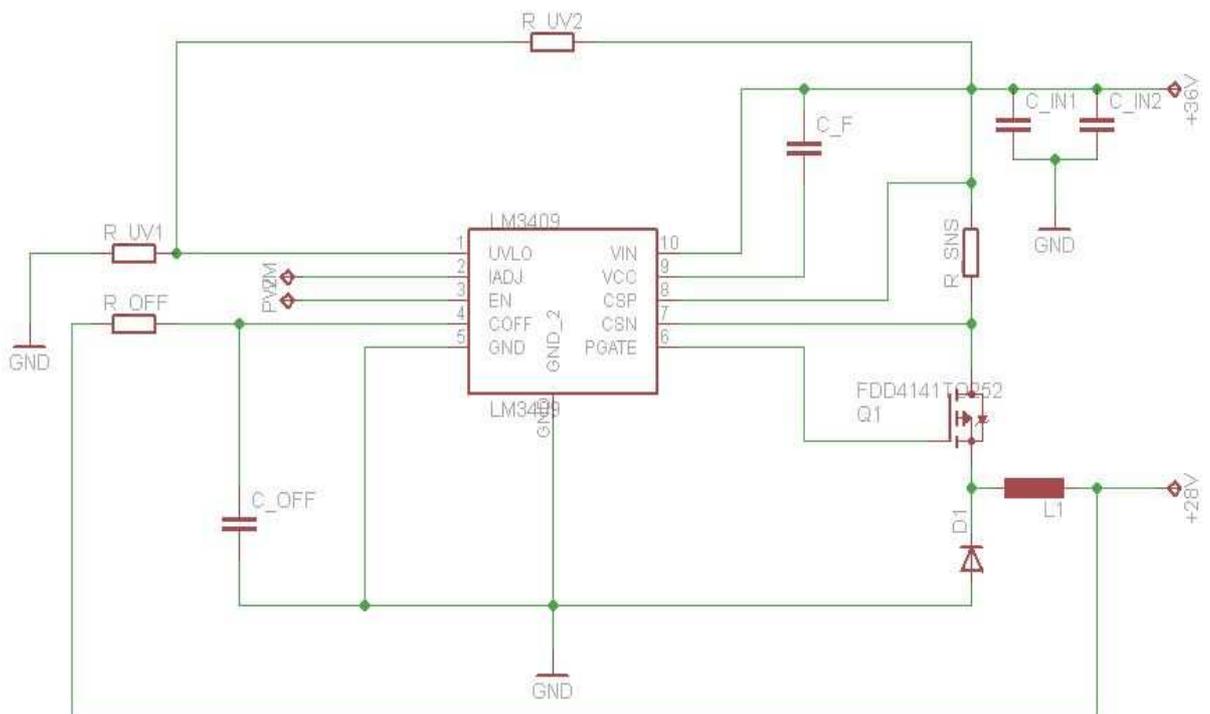


Abbildung 3.2 LED- Treiberschaltung für 10 W LEDs

Einige Bauteile wurden bewusst größer dimensioniert, da die Schaltung ursprünglich dafür gedacht war alle vier LEDs mit einem Schaltkreis zu betreiben. Geplant war eine Parallelschaltung von je zwei LEDs in Reihe. Diese Idee wurde aber später verworfen, da es immer besser ist, LEDs ausschließlich in Reiche zu schalten, so dass alle LEDs vom gleichen Strom durchflossen werden. Dies garantiert eine gleichmäßige Helligkeit. Für zukünftige Erweiterungen soll der Schaltkreis ebenso für eine Eingangsspannung von 48 V geeignet sein. Somit wäre unter Verwendung eines 48 V Akkus auch ein Betrieb bei Netzausfall möglich.

Tabelle 3.3 Bauelemente zur Beschaltung des Buck Converters

Bauteil	Bezeichnung	el. Größen	Hersteller
Kondensator	C_F	1 μ F	KEMET
Kondensator	C_{off}	470 pF	KEMET
Widerstand	R_{off}	16 k Ω	VISHAY
Induktivität	L_1	33 μ H; 3,9 A	PANASONIC
Widerstand	R_{SNS}	0,3 Ω	VISHAY
Kondensator	C_{in1}, C_{in2}	2,2 μ F; 100 V	KEMET
Widerstand	R_{UV2}	49,9 k Ω	VISHAY
Widerstand	R_{UV1}	6,98 k Ω	VISHAY
Transistor	Q_1	100 V; 3,8 A	DIODES INC.
Schottky Diode	D_1	100 V; 3 A	VISHAY

3.3.1 Platinenlayout

Nachdem die Bauteile der Schaltung festgelegt waren, konnte mit dem Layout der Testplatine begonnen werden. Dies geschah ebenfalls, wie schon beim Schaltungsentwurf, mit dem Programm EAGLE-5.6.0. Beim Layout ist darauf zu achten, dass die Bauteile, welche direkt mit dem IC verbunden sind möglichst nahe an diesem platziert werden. Dies dient der EMV und somit zur Vermeidung von Störungen. Weiterhin waren einige Durchkontaktierungen notwendig. Zum einen um zwei Masseflächen miteinander zu verbinden und somit eine Vergrößerung der Massefläche zu erreichen. Zum anderen um das Anschlusspad an der Unterseite des ICs mit Masse zu verbinden. Bei diesem Masseanschluss ist darauf zu achten, dass dieses Anschlusspad exakt mit Masse verbunden ist. Wenn dies nicht exakt auf Masse liegt, kann dies von Funktionsausfällen bis hin zur Zerstörung des ICs führen.

Des Weiteren ist darauf zu achten, dass alle Leiterzüge die richtige Breite besitzen, da innerhalb der Schaltung unterschiedlich große Ströme fließen. Der Abstand der einzelnen Leiterzüge untereinander ist ebenfalls zu beachten, um genügend Isolationsabstand zu gewährleisten. Rechtwinklige Abbiegungen und Abzweigungen in den Leiterzügen sind zu vermeiden, da dies einen schlechteren Stromfluss zur Folge hat. Bei sich stark erwärmenden Bauteilen ist es ratsam, diese auf eine möglichst große Kupferfläche zu platzieren. Dies trägt während des Betriebes zur Kühlung der Bauteile bei. Alle bedrahteten Bauteile sind gespiegelt darzustellen, da diese sich auf der anderen Seite der Platine befinden.

3.3.2 Bedrucken und Ätzen der Testplatine

Um eine Platine zu ätzen, muss diese zuerst einmal belichtet oder, wie in diesem Fall, bedruckt werden. Der Platinendruck erfolgt in zwei Schritten. Zuerst wird das fertige Layout (ohne Bauteile), unter Verwendung eines Laserdruckers, spiegelverkehrt auf Hochglanzpapier gedruckt. Ein Laserdrucker ist notwendig, weil sich der Toner unter Wärmeeinwirkung vom Hochglanzpapier löst. Im zweiten Schritt wird das ausgedruckte Layout sorgfältig auf die leere Platine aufgebügelt, wobei sich der Toner vom Papier löst und auf der Platine haften bleibt. Vor dem Aufbügeln sollte die Platine gereinigt und leicht aufgeraut werden. Nach dem Aufbügeln kann die Platine ins Ätzbad gegeben werden, wobei sich die nicht bedruckten Kupferflächen von der Platine lösen. Für das Ätzbad wurde eine Natriumpersulfat-Lösung verwendet. Der bestmögliche Ätzvorgang wird erreicht, wenn die Platine mit der bedruckten Seite nach unten, schwimmend auf der Ätzlösung, liegt. In diesem Fall können die abgelösten Kupferteilchen direkt zum Boden sinken. Würde die Platine in der Lösung liegen oder stehen, so müsste die Lösung ständig in Bewegung gehalten werden, um die abgelösten Bestandteile von der Platine zu spülen. Sind alle nicht bedruckten Flächen komplett vom Kupfer befreit, muss die Platine sorgfältig mit Azeton gereinigt werden, um den Toner wieder zu entfernen. Danach ist eine optische Kontrolle, auf eventuelle Fehler oder Kurzschlüsse notwendig. Hierbei empfiehlt es sich die Platine mit einer möglichst hohen Auflösung einzuscannen. Nachdem die Kontrolle positiv ausgefallen ist, müssen noch die Löcher für die Durchkontaktierungen gebohrt werden, so dass im Anschluss die Bauteilbestückung erfolgen kann. Das Platinenlayout und die fertige Testplatine sind in Abbildung 3.3 dargestellt.

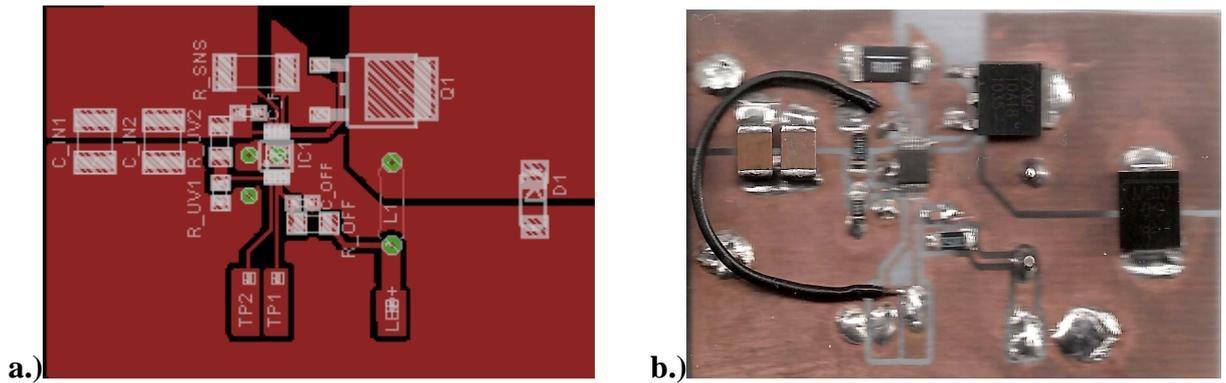


Abbildung 3.3 Platinenlayout der LED- Treiberschaltung a.) und fertige Testplatine b.)

3.4 Versuche und Messungen an der Testplatine

Nachdem alle Bauteile ordnungsgemäß aufgelötet waren, konnte mit ersten Messungen begonnen werden. Als erstes wurde der Durchgang zwischen den einzelnen Leiterbahnen gemessen, um festzustellen, ob Kurzschlüsse zwischen diesen bestehen. Danach wurde geprüft, ob sich Bauteile beim Anlegen einer Gleichspannung zwischen 0 V und 43 V erwärmen. Nachdem beide Tests positiv verliefen, konnte damit begonnen werden Spannung und Strom am Ausgang der Treiberschaltung zu messen. Zu diesem Zweck wird die Schaltung im Leerlauf betrieben, sodass der IC lediglich die Eingangsspannung durchsteuert. Dies geschieht, indem der Enable-Pin des ICs auf das positive Potential der Eingangsspannung gelegt wird. Dabei wurde festgestellt, dass der IC abschaltet sobald am Eingang weniger als 18 V anliegen. Das heißt, auf Grund der Beschaltung ist die minimale Leerlaufspannung des ICs auf 18 V festgelegt. Bei dieser Messung war zu erkennen, dass die Ausgangsspannung proportional zur Eingangsspannung ist und sich solange mit erhöht, bis die 28 V erreicht sind für welche die Schaltung ausgelegt ist. Wird die Eingangsspannung weiter erhöht, bleibt die Spannung am Ausgang konstant auf 28 V. Dies funktioniert natürlich nur bei angeschlossener Last, andernfalls wäre die Ausgangsspannung immer gleich der Eingangsspannung. Dabei sollte allerdings beachtet werden, dass der Eingangsspannungsbereich auf 48 V begrenzt ist. Die Verläufe der Spannungen am Buck Converter sind in Abbildung 3.4 graphisch dargestellt. Die blaue Kennlinie symbolisiert die angenommenen Sollwerte der Spannung am Ausgang und die grünen Punkte sind die tatsächlich gemessenen Werte. Hierbei ist zu erkennen, dass die Messwerte sehr gut mit den Sollwerten der Spannung übereinstimmen.

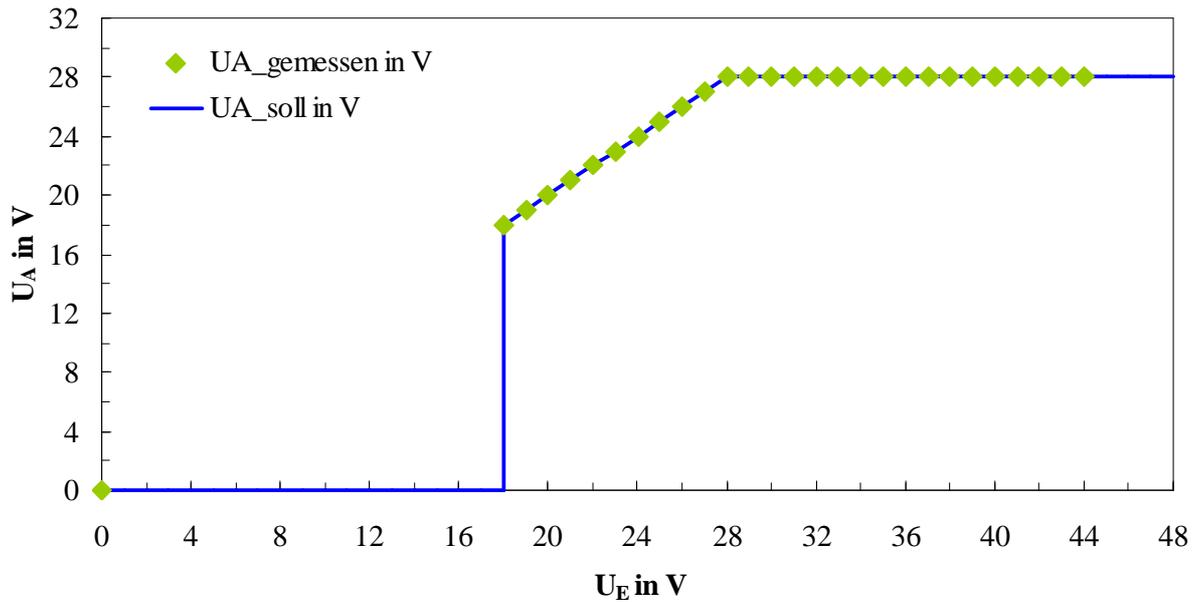


Abbildung 3.4 Ausgangsspannung in Abhängigkeit der Eingangsspannung

Beim Messen des Stromes am Ausgang ist festzustellen, dass bis zu einer Eingangsspannung von 20 V kein Strom fließt. Dies ist darauf zurückzuführen, dass der IC unter 18 V abschaltet und somit keine Spannung am Ausgang anliegt, um eine mögliche Last zu betreiben. Des Weiteren ist aus dem Datenblatt der LEDs zu entnehmen, dass diese erst ab einer Spannung von circa 11 V einen Strom fließen lassen. Bei zwei LEDs in Reihe geschaltet, würde dies eine Spannung von 22 V ergeben, was in guter Näherung den gemessenen 20 V entspricht. Ab dieser Eingangsspannung geht der Strom durch die 10 W LEDs steil nach oben und pendelt sich dann, nach leichtem Überschwingen, bei 690 mA ein. Die Verläufe von Spannung und Ausgangsstrom am Buck Converter sind in Abbildung 3.5 noch einmal graphisch dargestellt. Dabei symbolisiert die blaue Kennlinie die Diodenkennlinie aus dem Datenblatt der LED und die roten Punkte die tatsächlich gemessenen Werte. Bis auf geringe Abweichungen, welche auf die ausgewählten Widerstände R_{OFF} und R_{SNS} zurückzuführen sind, stimmen die gemessenen Werte ziemlich gut mit den Sollwerten überein.

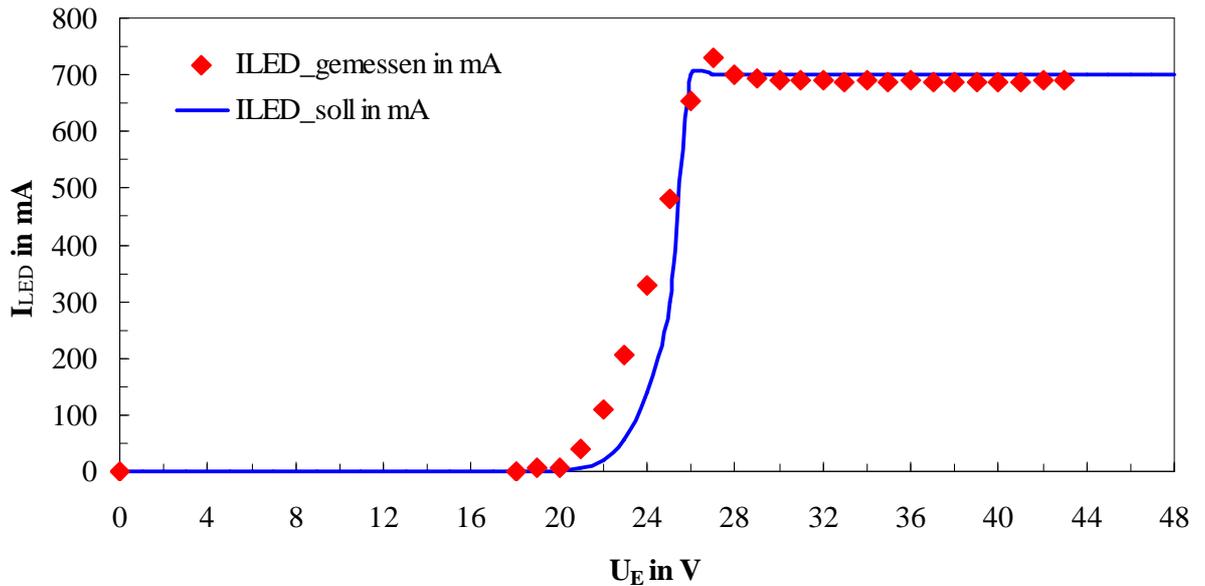


Abbildung 3.5 Ausgangsstrom in Abhängigkeit der Eingangsspannung

Es ist zu erkennen, dass ab einer Eingangsspannung von 28 V der Strom am Ausgang konstant bleibt. Somit ist bewiesen, dass die Funktion als Konstantstromquelle exakt gewährleistet wird. Beide Messungen konnten allerdings nur bis zu einer Spannung von 43 V durchgeführt werden, da das verwendete Netzteil nur eine Spannung von maximal 43 V am Ausgang bereitstellt.

3.5 Entwicklung der Hauptplatine

Bei der Auswahl eines geeigneten Mikrocontrollers wurde sich für einen Controller der Firma ATMEL entschieden, mit der Bezeichnung ATMEGA48A-PU. Dieser wurde ausgewählt, weil er genügend Ein- und Ausgänge besitzt um den Anforderungen an die Schaltung gerecht zu werden und weil er zudem noch preisgünstig war. Des Weiteren wurde sich für die bedrahtete Variante dieses Controllers entschieden, da dieser auf einen Stecksockel montiert und so jederzeit ausgetauscht werden kann. Außerdem war ein geeignetes Programmiergerät für diese Mikrocontroller vorhanden. Die grundlegende Beschaltung zum Betreiben eines Mikrocontrollers und zum Anschließen der Programmierschnittstelle stammt aus dem Datenblatt und von der Application Note AVR ISP (vgl. [12, 21]).

Für den Schaltungsentwurf wurde, wie auch schon bei der Testplatine, das Programm EAGLE-5.6.0 verwendet.

Bei der Taktfrequenz des Mikrocontrollers wurde ein Quarz gewählt, welches eine Frequenz von 20 MHz generiert und welches ohne zusätzliche Beschaltung am Mikrocontroller angeschlossen werden kann. Nachdem die Grundschialtung fertig gestellt war, musste die LED-Treiberschaltung mit eingebunden werden.

Für jede der beiden Treiberschaltungen war jeweils ein Ausgang des Mikrocontrollers notwendig, mit dem ein PWM-Signal ausgegeben werden kann. Die PWM dient dazu, die LEDs auf eine gewünschte Helligkeit zu dimmen, was zusätzlich zum Energiesparen beiträgt. Außerdem wurde ein Ausgang benötigt, um die Spannungsversorgung der LEDs (Netzteil) zuschalten zu können. Ein Konzept zur Ansteuerung mittels Mikrocontroller ist in Abb. 3.6 schematisch dargestellt.

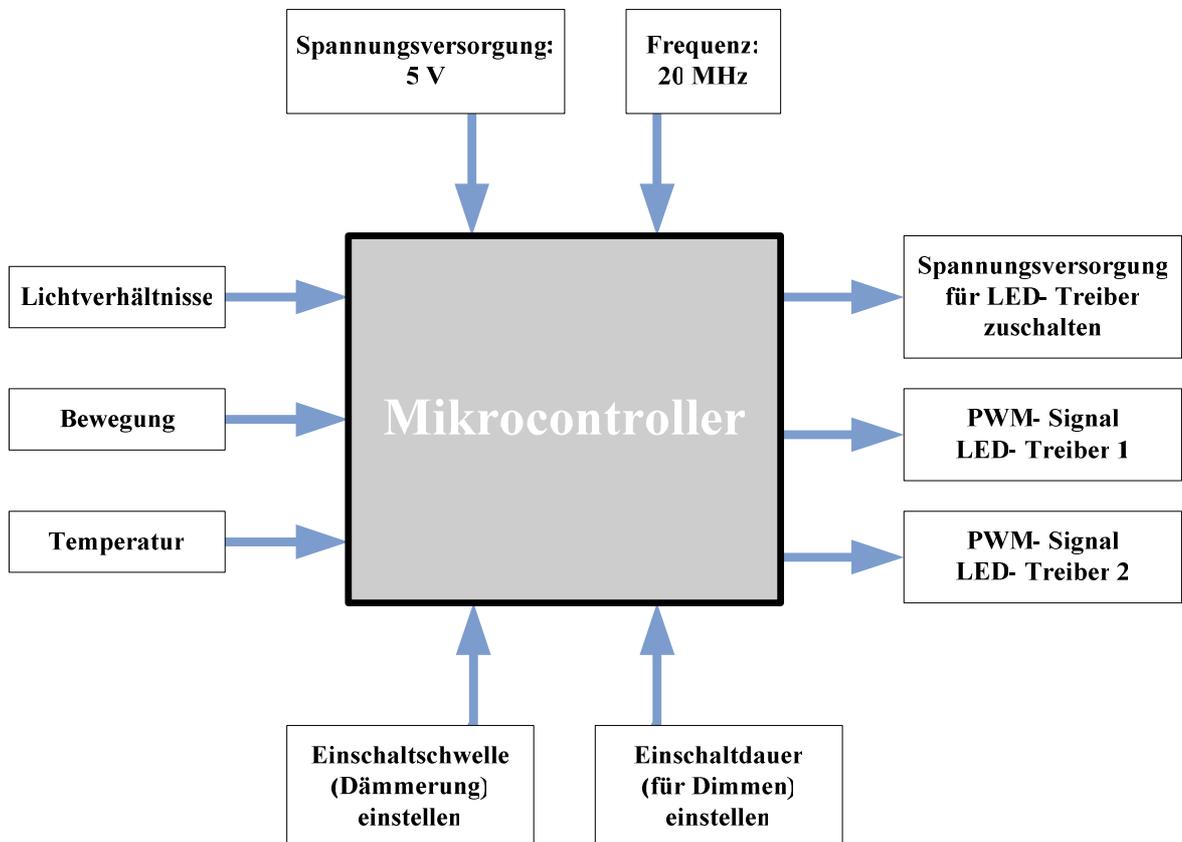


Abbildung 3.6 Schematische Darstellung der Funktionen des Mikrocontrollers

3.6 Signalerfassung mittels Mikrocontroller

Es wurden drei Eingänge des Mikrocontrollers ausgewählt, welche dazu dienen die Signale eines Helligkeitssensors, eines Bewegungsmelders und eines Temperaturfühlers zu erfassen. Mit der Auswertung dieser drei Signale soll der Mikrocontroller die LED-Treiber ansteuern.

3.6.1 Helligkeitssensor zum Ein- und Ausschalten

Als Helligkeitssensor sollte eine Photodiode dienen. Allerdings ist hier eine entsprechende Verstärkerschaltung mit Operationsverstärker notwendig, da das erzeugte Signal der Photodiode zu schwach ist. Beim Operationsverstärker wurde ein LM358D verwendet, da dieser preisgünstig ist und für solche Verstärkerschaltungen empfohlen wird. Die entsprechende Verstärkerschaltung stammt aus der Literatur *Halbleiter-Schaltungstechnik*. Nach einigen Tests wurde festgestellt, dass die Photodiode vom Lichtspektrum her bzw. bei Dämmerung sehr eingeschränkt ist. Aus diesem Grund wurde bei der endgültigen Platine ein Phototransistor verwendet, um die Helligkeit der Umgebung zu erfassen. Dieser dient mehr oder weniger als lichtabhängiger Schalter, d.h., mit steigender Helligkeit steigt auch die Leitfähigkeit des Transistors. In diesem Fall kann auf die Verstärkerschaltung mit dem OPV verzichtet werden, da hier bei voller Helligkeit die volle Eingangsspannung am Ausgang des Transistors anliegen kann. Zur Verbesserung der EMV und zur genaueren Messung wurde eine Signalanpassung mittels RC-Filter vorgenommen.

3.6.2 Bewegungsmelder zum Dimmen

Da der Bewegungsmelder wie ein Schalter anzusehen ist, konnte dieser an einen digitalen Eingang angeschlossen werden. Somit gilt bei Bewegung Signal High (5 V) und bei keiner Bewegung Signal Low (0 V). Allerdings wurde bei ersten Tests festgestellt, dass die Signalleitungen bis hin zum Mikrocontroller-Eingang sehr empfindlich gegenüber elektromagnetischen Einflüssen sind. Das heißt, am Signaleingang lag permanent ein Signal an, da die Leitungen mehr oder weniger als Antenne dienten. Um dies zu kompensieren und diese Fehlerquelle zu beseitigen wurden später ein Widerstand in Reihe zum Signaleingang und ein weiterer Widerstand und ein Kondensator parallel gegen Masse verbaut.

3.6.3 Temperatursensor zur thermischen Überwachung mit Notausschaltfunktion

Für die Temperaturmessung wurde ein NTC-Widerstand verwendet. Da die Änderung des Widerstandes bezüglich der Temperatur, im oberen Temperaturbereich sehr gering ist, ist für eine exakte Temperaturmessung immer eine entsprechende Messbrücke notwendig. Da dieser NTC nur zur Temperaturüberwachung dient und keine genauen Werte erfassen soll, kann auf eine Messbrücke verzichtet werden. In diesem Fall ist es ausreichend einen Spannungsteiler aus 2 Widerständen zu verwenden.

Einer der beiden Widerstände ist der NTC und der andere ist ein dementsprechender Messwiderstand, über dem der Spannungsabfall gemessen wird. Beim NTC wurde aus vier verschiedenen Widerständen ein Widerstand vom Typ NTC-0,2 2,2K ausgewählt. Dieser besitzt bei 25 °C einen Widerstandswert von 2,2 kΩ und hat in einem Temperaturbereich von 25 °C bis 100 °C eine relativ passable Widerstandsänderung. Passend dazu wurde ein Messwiderstand von 613 Ω ausgerechnet und daraufhin ein Widerstand von 680 Ω gewählt. Der Messwiderstand lässt sich nach Formel [5] wie folgt berechnen. Hierbei sind die Widerstände R_u bei 25 °C, R_m bei 50 °C und R_o bei 75 °C.

$$R_p = \frac{R_m \cdot (R_u + R_o) - 2 \cdot R_u \cdot R_o}{R_u + R_o - 2 \cdot R_m} \quad [5]$$

$$R_p = \frac{798\Omega \cdot (2215\Omega + 327\Omega) - 2 \cdot 2215\Omega \cdot 327\Omega}{2215\Omega + 327\Omega - 2 \cdot 798\Omega} = 613\Omega$$

Die Widerstandskurven des gewählten NTC sowie der anderen drei NTC-Widerstände sind in Abbildung 3.7 und 3.8 graphisch dargestellt. Dabei zeigt die rote Kurve die Widerstandsänderung des ausgewählten NTC-Widerstandes NTC-0,2 2,2K.

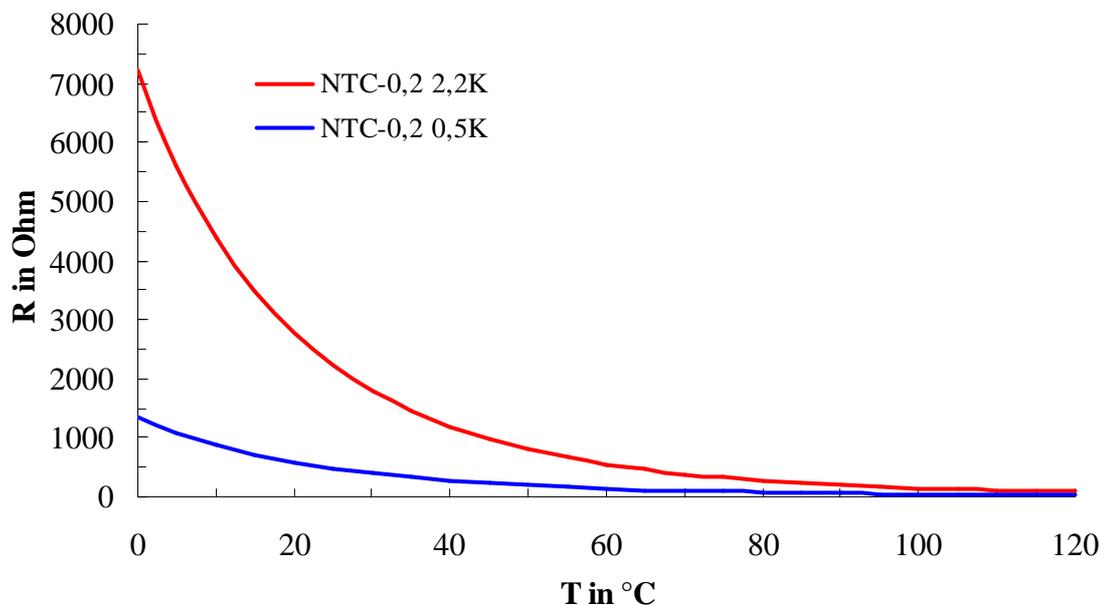


Abbildung 3.7 Widerstandsänderung am NTC in Abhängigkeit der Temperatur

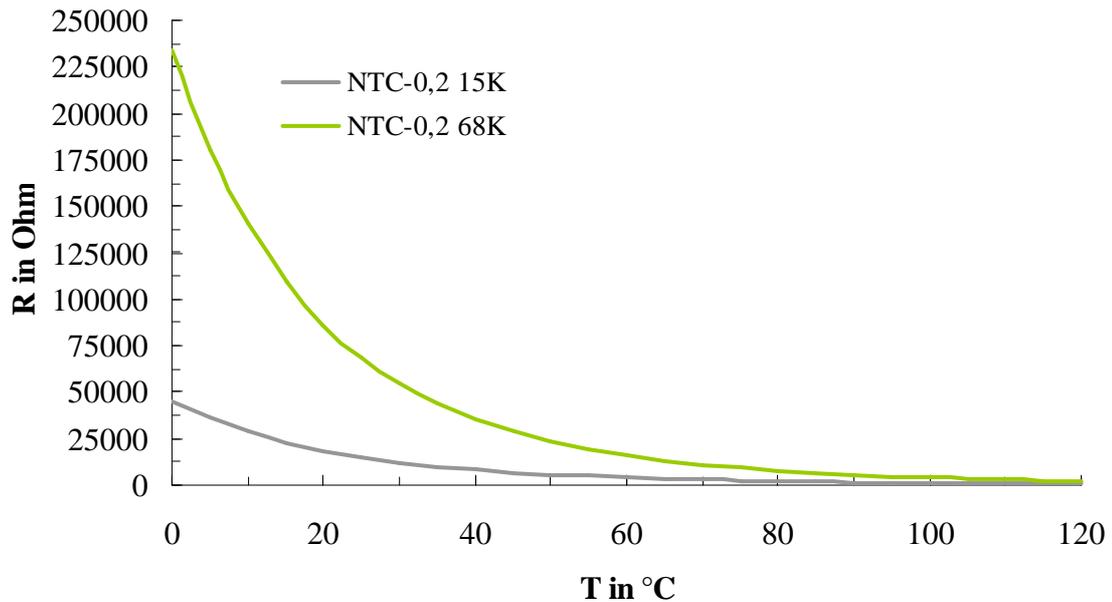


Abbildung 3.8 Widerstandsänderung am NTC in Abhängigkeit der Temperatur

3.7 Entwicklung und Fertigung der Hauptplatine

Nachdem die SMD-Bauteile ausgewählt waren, mussten noch die notwendigen bedrahteten Bauteile ausgewählt werden. Weiterhin waren zwei Netzteile notwendig, zum einen ein kleines, welches 5 V Gleichspannung bereitstellt und zur Platinenmontage geeignet ist, und zum anderen ein größeres, welches 36 V Gleichspannung bereitstellt. Das kleinere Netzteil dient zur Spannungsversorgung des Mikrocontrollers und das größere Netzteil als Spannungsquelle für die Hochleistungs-LEDs. Beide Netzteile mussten eingangsseitig für 230 V Netzspannung geeignet sein. Für das kleinere Netzteil wurde sich entschieden, weil es zur Platinenmontage geeignet ist und aufgrund seiner Baugröße problemlos auf der vorgesehenen Platine Platz findet. Bei dem anderen Netzteil war die Leistung entscheidend, die das Netzteil konstant bereitstellen kann, in diesem Fall 50 W. Weiterhin spielten aber auch die Baugröße und der Preis eine Rolle. Nachdem zwei passende Netzteile gefunden waren, konnte mit dem Schaltungsentwurf für die vollständige Platine begonnen werden. Bei dem endgültigen Schaltplan wurde nach wie vor mit dem Programm EAGLE-5.6.0 gearbeitet. Kleinere Probleme traten dahingehend auf, dass einige Bauteile in der Bibliothek von EAGLE nicht vorhanden waren. In diesem Fall mussten diese Bauteile selbst entworfen und dimensioniert werden. Es handelt sich hierbei um das kleine 5 V Netzteil und einige kleine Anschlussklemmen. Das Programm ermöglicht es allerdings, Bauteile jeglicher Art selbst zu konstruieren, womit dieses Problem schnell gelöst wurde.

3.7.1 Leistungsteil der Platine

Um das 36 V Netzteil einzuschalten, wurden zwei Relais und ein Leistungstransistor vorgesehen, welcher die Relais ansteuert. Dieser Transistor wiederum wird vom Mikrocontroller angesteuert. Zum Schutz des Transistors wurde aus Sicherheitsgründen eine Schottky-Diode parallel zu den Relais geschaltet. Diese dient als Freilaufdiode, um die Gegenspannung zu begrenzen, die beim Abschalten der Relaisspule auftritt. Diese Spannung mit entgegengesetzter Polarität entsteht beim Abschalten des Stromes durch die Relaisspule (durch Selbstinduktion) und kann den Wert der Nennspannung um ein Vielfaches überschreiten und somit den Transistor zerstören.

Um mögliche Überspannungen zum Mikrocontroller hin abzufangen, wurde in Reihe zum Enable-Eingang des Buck Converters jeweils ein Widerstand vorgesehen und in Verbindung damit je eine Zener-Diode gegen Masse. Die Diode wird hierbei in Sperrrichtung betrieben. Diese Schaltung zur Spannungsbegrenzung ist dafür gedacht, den Mikrocontroller vor zu hohen Spannungen zu schützen. Da am Buck Converter eine Spannung von bis zu 48 V anliegen kann und diese im Fehlerfall intern an den Enable- Eingang ausgegeben werden kann, wurde eine Zener-Diode mit 5,6 V Durchbruchspannung gewählt. Das bedeutet, dass im Fehlerfall die rückwärtige Spannung hin zum Mikrocontroller maximal 5,6 V betragen kann. Bei Spannungen über 5,6 V bricht die Diode durch und verursacht somit einen Kurzschluss gegen Masse, was zum Auslösen einer Feinsicherung führt. Deshalb wurde zum Überlastschutz und zum Schutz des 36 V Netzteils eine 3,15 A Feinsicherung zwischen Netzteil und Buck Converter eingebaut.

Für die Anschlussleitungen von 230 V Netzspannung und 36 V Betriebsspannung wurden auf der Platine große Anschlussklemmen vorgesehen, welche bis 250 V und 13 A belastbar sind. Außerdem sind diese Klemmen für Querschnitte bis 2,5 mm² geeignet. Für den Anschluss der Hochleistungs-LEDs und der Signalleitungen genügten kleinere Anschlussklemmen, welche aber dennoch bis 200 V und 10 A belastbar sind. Allerdings sind diese Klemmen nur bis zu einem Querschnitt von 1 mm² geeignet.

Zum Programmieren des Mikrocontrollers wird eine Programmierschnittstelle benötigt. Diese ist notwendig, um ein geschriebenes Programm über ein Programmiergerät in den Mikrocontroller zu übertragen bzw. um einen Reset durchzuführen. Die Schnittstelle besteht aus einer 6-poligen Stiftleiste, welche auf die Platine aufgelötet wird. Zur Kontrollanzeige bei Programmstart und beim erfolgreichen Übertragen eines neuen Programms wurde eine grüne LED vorgesehen. Diese LED wurde im SMD- Format gewählt und mit einem Vorwiderstand versehen.

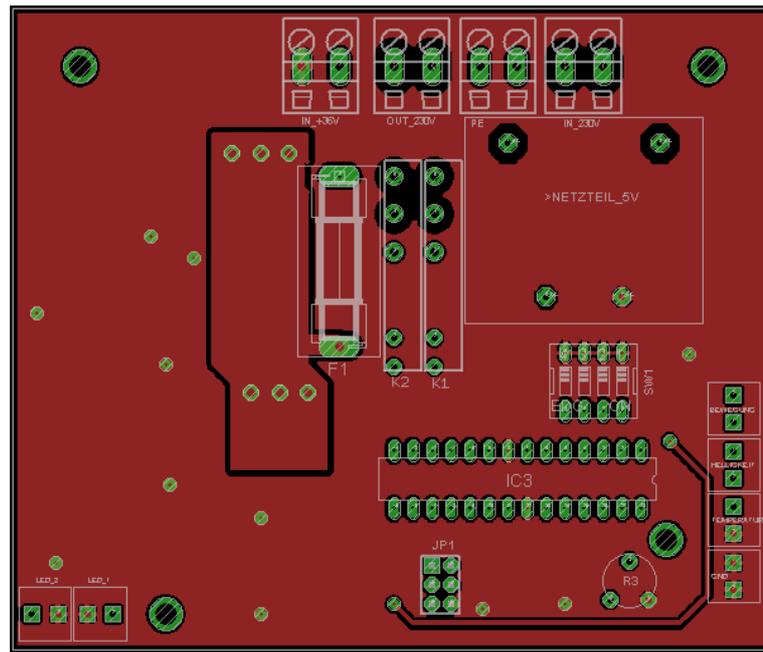


Abbildung 3.10 Platinenlayout der Oberseite mit bedrahteten Bauteilen

3.7.3 Fertigung der Hauptplatine

Nachdem das beidseitige Layout der Hauptplatine fertig gestellt war, wurde dieses mit dem gleichen Verfahren wie bei der Testplatine bedruckt und geätzt. Vor dem Bestücken wurden alle Leiterzüge auf Durchgang bzw. Kurzschluss gegenüber anderen Leiterzügen geprüft. Alle Tests verliefen positiv. Beim Bestücken der Platine wurde diesmal eine andere Herangehensweise gewählt. Alle SMD-Bauteile wurden im so genannten Lötoven-Verfahren aufgelötet. Hierbei werden zuerst alle Lötstellen mit einer speziellen Lötpaste versehen, was auch als Dispensen bezeichnet wird. Danach werden alle Bauteile, mit Pinzette, an ihre entsprechende Position gebracht. Als Letztes kommt die bestückte Platine in einen speziellen Lötoven. Dieser Ofen fährt nach einem vorgegebenen Programm bestimmte Temperaturwerte ab. Hierbei wird das Zinn in der Lötpaste weich und verbindet sich mit den Bauteilen, die anderen Bestandteile der Paste verdampfen. Dieses Lötverfahren empfiehlt sich besonders für Bauteile, welche einen Anschlusspunkt an der Unterseite besitzen. Da sich auf der Platine solche Bauteile befinden und um Lötfehler auszuschließen wurde sich bei der Hauptplatine für dieses Lötverfahren entschieden. Die bedrahteten Bauteile, welche sich auf der anderen Seite der Platine befinden, wurden von Hand aufgelötet. Anschließend wurden Funktionstests der beiden Abwärtswandler und des 5 V Netzteils durchgeführt.

Zum Schutz vor Korrosion wurde die Platine am Ende noch einmal beidseitig mit Kolofonium eingestrichen. Die Ober- und Unterseite der fertigen Platine sind in Abb. 3.11 dargestellt.

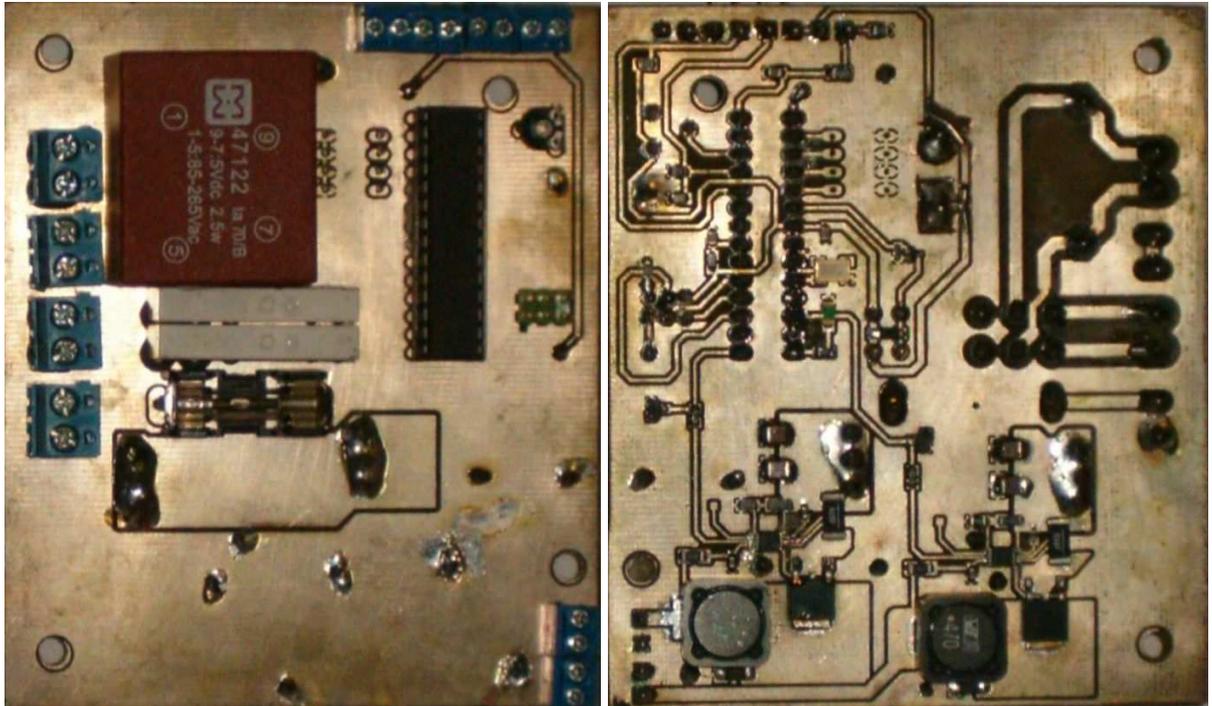


Abbildung 3.11 Oberseite und Unterseite der endgültigen Platine

3.8 Kühlkörperberechnung

Zum Betreiben der verwendeten LEDs ist ein Kühlkörper nötig, da diese im Dauerbetrieb eine Junction-Temperatur von $65\text{ }^{\circ}\text{C}$ nicht überschreiten dürfen. Die LEDs sind bereits auf einer Aluminiumfläche aufgebracht, so dass diese direkt auf einen Kühlkörper montiert werden können. Als Kühlkörpermaterial soll Aluminium dienen, da es ein sehr guter Wärmeleiter ist und sich leicht und gut zu verarbeiten lässt. Bei der Form des Kühlkörpers wurde sich für ein Blech entschieden, welches ausreichend Dicke und Fläche besitzt, um die Wärme abzuleiten. Für die Dimensionierung der Fläche wurde eine einfache Kühlkörperberechnung genutzt. Der Wärmewiderstand setzt sich hier aus drei unterschiedlichen Wärmewiderständen zusammen, dem Verbindungswiderstand zwischen LED und Gehäuse R_{thJC} , dem Übergangswiderstand vom Gehäuse zum Kühlkörper R_{thCH} (welcher durch die Wärmeleitpaste bestimmt wird) und dem Wärmeübergangswiderstand vom Kühlkörper zur Umgebung R_{thHA} . Der gesamte Wärmewiderstand R_{thJA} ergibt sich aus einer Gemischtschaltung der unterschiedlichen Teilwiderstände. Die Schaltung zur Berechnung der Widerstände ist in Abb. 3.12 zu sehen.

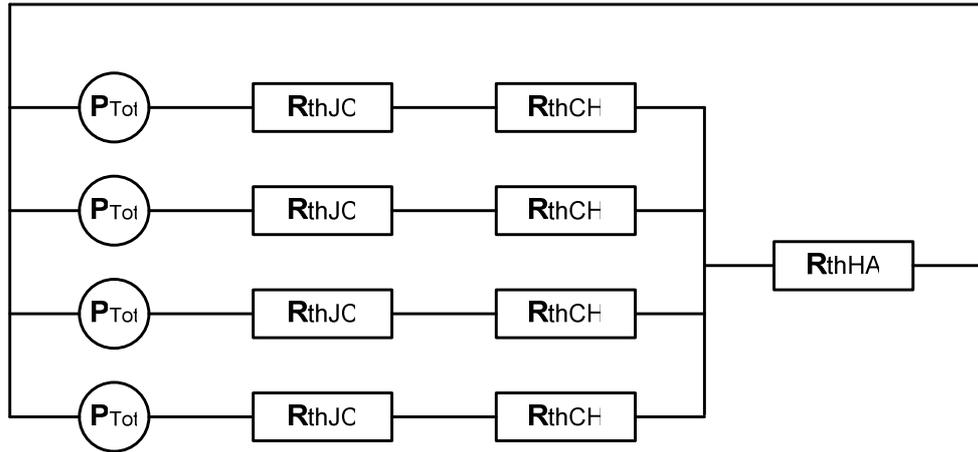


Abbildung 3.12 Gemischtschaltung der einzelnen Wärmewiderstände

Die Verlustleistung P_{Tot} der einzelnen LEDs ist hierbei als Spannungsquelle anzusehen. Da alle vier LEDs auf den gleichen Kühlkörper sollen, ergibt sich eine Gemischtschaltung aus Parallel- und Reihenschaltung, mit vier gleichen Stromquellen. Aus dieser Schaltung ergibt sich der gesamte Wärmewiderstand nach Formel [6].

$$R_{thJA_ges} = \frac{R_{thJC} + R_{thCH}}{4} + R_{thHA} \quad [6]$$

Die Temperaturdifferenz an den Betriebsmitteln ergibt sich aus Formel [7], woraus sich nach Formel [8] der Temperaturwiderstand des Kühlkörpers berechnen lässt.

$$\Delta\vartheta = \vartheta_{max} - \vartheta_A \quad [7]$$

$$\Delta\vartheta = P_{Tot} \cdot R_{thJA_ges} \rightarrow R_{thHA} = \frac{\Delta\vartheta}{P_{Tot}} - \frac{R_{thJC} + R_{thCH}}{4} \quad [8]$$

Mit diesem Wert lässt sich nun der gesamte Wärmewiderstand der Schaltung in guter Näherung berechnen, woraus sich dann nach Formel [9] die gesuchte Kühlfläche ergibt.

$$R_{thJA_ges} = \frac{\Delta\vartheta}{P_{Tot}} = \frac{1}{\alpha \cdot A} \rightarrow A = \frac{1}{\alpha \cdot R_{thJA_ges}} \quad [9]$$

Da das Kühlblech lediglich 2 mm dick ist, wurden an den Stellen der Wärmeerzeugung noch zusätzliche Aluminiumbleche angebracht, um die Querschnittsfläche und somit die Wärmeleitung an dieser Stelle zu verbessern. (vgl. [5])

4 Softwareentwicklung und Mikrocontrollerprogrammierung

Nach erfolgreichem Funktionstest der Platine und des Mikrocontrollers, konnte mit der Softwareentwicklung begonnen werden. Es stehen zwei Programmiersprachen zur Verfügung, mit denen der verwendete Mikrocontroller programmiert werden kann. Die Programmiersprache Assembler und die Programmiersprache C. Bei diesem Anwendungsfall wurde sich für die Programmiersprache C entschieden. Die Softwareentwicklung erfolgt mit dem Programm AVR Studio 4, von Atmel, welches im Internet kostenlos zum Download bereitgestellt wird.

4.1 Entwicklung des Grundprogramms

Als erstes müssen die Grundfunktionen ins Programm geladen werden, auf die dann während des Programmlaufes zugegriffen wird. Im Anschluss wird die CPU-Frequenz festgelegt, mit welcher der Mikrocontroller getaktet ist. Nun kann der ADC initialisiert und gestartet werden. Ein weiterer wichtiger Schritt zu Beginn ist die Initialisierung globaler Variablen. Ein Ausschnitt des Grundprogramms ist in Abb. 4.1 zu sehen.

```
#include <avr/io.h>           // laden der notwendigen Funktionen
#include <util/delay.h>
#include <stdint.h>
#include <avr/interrupt.h>
#define F_CPU 2000000UL      // CPU- Frequenz einstellen

void ADC_Init(void);         // ADC starten
uint8_t sw1 = 0;             // Initialisierung globaler Variablen
uint8_t sw2 = 0;
uint8_t sw3 = 0;
uint8_t sw4 = 0;
uint8_t i = 0;
...
```

Abbildung 4.1 Ausschnitt aus dem Quellcode des Grundprogramms

4.2 Entwicklung des Hauptprogramms

Zunächst muss eine Hauptfunktion angelegt werden, in der alle Bedingungen bzw. Schleifen abgefragt werden. Beim Programmaufruf werden alle Ein- bzw. Ausgangsports auf Null gesetzt, so dass bei Programmstart keine fälschlichen Werte eingelesen oder ausgegeben werden. Danach wird festgelegt, welche Pins als Eingang und welche als Ausgang dienen sollen. Zum Initialisieren der Ein- und Ausgänge wird das Datenblatt des ATMEGA48 benötigt, da hier Angaben zur Pinbelegung und den zugehörigen Funktionen entnommen werden können. Ein Programmausschnitt hierzu ist in Abb. 4.2 dargestellt.

```
int main(void)
{
// alle Bits von Port B, C und D auf 0 setzen
    DDRB = 0x00; PORTB = 0x00;
    DDRC = 0x00; PORTC = 0x00;
    DDRD = 0x00; PORTD = 0x00;

    DDRB = (1<<PB0) | (1<<PB1); // Ausgang Kontroll- LED & Enable LED-Treiber2
    ...

// Pullup- Widerstände für Schaltereingänge an
    PORTD = (1<<PD0) | (1<<PD1) | (1<<PD2) | (1<<PD3);
    ...
}
```

Abbildung 4.2 Ausschnitt aus dem Quellcode der Hauptfunktion

Nachdem alle verwendeten Pins initialisiert sind müssen die Timer eingestellt werden. Dies ist ebenfalls nur mit Hilfe des Datenblattes möglich. Hier ist genau beschrieben welche Bits gesetzt werden müssen, damit der Timer eine gewünschte Funktion erfüllt. Die ersten beiden Timer sollen für die Ausgabe einer PWM genutzt werden, mit der die LED-Treiber angesteuert werden.

Zunächst wird festgelegt, ob Fast-PWM oder phasenkorrekte PWM genutzt werden soll. Danach wird die PWM-Frequenz mittels Prescaler eingestellt. Dies muss für beide Timer separat eingestellt werden. Der dritte Timer wird so konfiguriert, dass damit später bestimmte Zählfunktionen realisiert werden können. Ein Ausschnitt aus dem entsprechenden Programm ist in Abb. 4.3 zu sehen. (vgl.[7, 8])

```

TCCR0A |= (1<<COM0A1) | (1<<WGM00); // Timer/Counter 0 für PWM einstellen
TCCR0B |= (1<<CS00) | (1 << CS01); // Prescaler festlegen
...
// Timer/Counter 2 konfigurieren
TCCR2A = (1<<WGM21); // CTC Modus
TCCR2B |= (1<<CS20) | (1<<CS21) | (1<<CS22); // Prescaler festlegen

```

Abbildung 4.3 Einstellen der Timer/Counter

Als Prescaler für die PWM-Frequenz wurde 64 gewählt und durch die Wahl der phasenkorrekten PWM kommt noch eine Teiler von Zwei hinzu. Der Teiler von (2^8-1) resultiert daher, dass hier ein 8 Bit Timer/Counter verwendet wurde. Damit ergibt sich nach Formel [10] die PWM-Frequenz wie folgt. (vgl.[12])

$$f_{\text{PWM}} = \frac{f_{\text{CPU}}}{2 \cdot \text{Prescaler} \cdot (2^8 - 1)} \quad [10]$$

$$f_{\text{PWM}} = \frac{20\text{MHz}}{2 \cdot 64 \cdot 255} = 612,7\text{Hz}$$

4.2.1 Aufnehmen analoger Werte über ADC-Eingänge

Um die Werte des Temperaturfühlers und des Helligkeitssensors, bzw. des Potentiometers aufnehmen zu können sind die analogen Eingänge des Mikrocontrollers notwendig. Hierfür muss als erstes der ADC initialisiert und eingestellt werden. Zunächst muss eine Referenzspannung festgelegt werden, welche vom ADC genutzt werden soll. Danach wird, wie bei den Timern, ein Frequenzvorteiler gewählt. Durch das Setzen des Bits ADEN auf 1 wird der ADC beim Aufruf der Funktion gestartet. Nach dem Start ist ein so genannter Dummy-Readout zu empfehlen, da bei einer ersten Messung falsche Werte aufgenommen werden. Hierbei wird ein Wert aufgenommen, welcher nach einer erfolgreichen AD-Wandlung wieder verworfen wird. Ein Ausschnitt aus der Funktion des ADC ist in Abb. 4.4 dargestellt.

```

void ADC_Init(void) {

ADMUX = (0<<REFS1) | (1<<REFS0);           // AVcc als Referenzspannung benutzen
ADCSRA = (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); // Frequenzvorteiler
ADCSRA |= (1<<ADEN);                       // ADC aktivieren
}

```

Abbildung 4.4 Initialisieren des ADC

Zudem wird beim Start des Programms und auch beim Schreiben eines neuen Programms in den Mikrocontroller eine grüne LED angesteuert. Diese signalisiert ein erfolgreiches Überschreiben, bzw. einen fehlerfreien Programmstart. (vgl.[12])

4.2.2 Programmstart und Temperaturüberwachung

Alle Abfragen hinsichtlich bestimmter erfüllter Bedingungen laufen in einer so genannten Endlosschleife (*while(1)*). Zuerst erfolgen die Aufnahme des Temperaturwertes und ein anschließender Vergleich mit dem festgelegten Höchstwert. Ist der gemessene Wert größer als der Höchstwert, erfolgt eine Notabschaltung. Dies bedeutet, der Programmablauf wird beendet und die Spannungsversorgung der LED-Treiber wird unterbrochen.

Da die Spannung am Messwiderstand theoretisch einen maximalen Wert von 5 V annehmen kann, würde in diesem Fall ein Strom von 7,35 mA fließen, wenn R_{NTC} gegen Null geht.

$$I_{\max} = \frac{U_{\max}}{R_{\text{mess}}} \quad [11]$$

$$I_{\max} = \frac{5V}{680\Omega} = 7,35mA$$

Der ADC arbeitet mit einer Auflösung von 10 Bit, d.h., die Spannung am NTC entspricht maximal einem Digitalwert von 1024. Der Widerstandswert des NTC sinkt mit steigender Temperatur, d.h., der Strom in der Reihenschaltung aus NTC und Messwiderstand steigt. Es wurde ein maximaler Temperaturwert von 70 °C festgelegt, bei dem eine Abschaltung erfolgen soll. Bei einem Temperaturwert von 70 °C hat der NTC einen Wert von 387 Ω. Nach Formel [12] ergibt sich somit bei 70 °C ein Strom von 4,7 mA. Bei extrem niedrigen Temperaturen würde der Strom theoretisch gegen Null gehen.

$$I_{70} = \frac{U_{\max}}{R_{\text{mess}} + R_{\text{NTC}}} \quad [12]$$

$$I_{70} = \frac{5\text{V}}{680\Omega + 387\Omega} = 4,7\text{mA}$$

Der Spannungsabfall am NTC sinkt mit steigender Temperatur, sodass sich nach Formel [13], bei 70°C eine Spannung von 1,82 V ergibt.

$$U_{\text{NTC}} = I_{70} \cdot R_{\text{NTC}} \quad [13]$$

$$U_{\text{NTC}} = 4,7\text{mA} \cdot 387\Omega = 1,82\text{V}$$

Da der Mikrocontroller die Spannung über dem NTC misst, muss für diesen Spannungswert der entsprechende Digitalwert errechnet werden.

Nach Formel [14] ergibt sich somit ein Digitalwert von 372, welcher als Referenzwert im Programm verwendet werden muss.

$$\frac{U_{\text{NTC}}}{U_{\max}} = \frac{x_{\text{Ref}}}{2^{10}} \rightarrow x_{\text{Ref}} = \frac{U_{\text{NTC}}}{U_{\max}} \cdot 2^{10} \quad [14]$$

$$x_{\text{ref}} = \frac{1,82\text{V}}{5\text{V}} \cdot 1024 = 372$$

Ein Ausschnitt aus dem Quellcode zur Temperaturüberwachung ist in Abb. 4.5 zu sehen.

```

while (adcval_1 < 372) // maximale Temperatur überschritten
{
    PORTD &= ~(1<<PD4); // Notausschaltung mittels Spannungsunterbrechung

```

Abbildung 4.5 Temperaturüberwachung und Notabschaltung

4.2.3 Anlegen von Funktionen

Um die LEDs Ein- und Auszuschalten bzw. zum Dimmen, wurden verschiedene Funktionen angelegt. Diese stehen außerhalb der Endlosschleife und werden bei Bedarf aufgerufen. Hierbei handelt es sich um Funktionen für den ADC, zum Aufnehmen von Analogwerten, zum Ein- und Ausschalten, zum Dimmen und zum Erzeugen verschiedener Zeiten.

Die ADC-Funktionen werden gebraucht, um die Lichtverhältnisse zu erkennen. Diese werden also nur aufgerufen, bei Änderung der Lichtverhältnisse (hell/dunkel). Die Funktionen zum Ein- und Ausschalten werden nur einmal, bei Erreichen der Dämmerung, bzw. bei Erreichen des Tageslichts aufgerufen. In Abb. 4.6 ist die Funktion zur ADC-Messung dargestellt und in Abb. 4.7 die Funktion zum Einschalten.

```
// ADC Mehrfachmessung mit Mittelwertbildung
uint16_t ADC_Read_Avg( uint8_t channel, uint8_t average )
{
    uint32_t result = 0;
    for (uint8_t k = 0; k < average; ++k )
        result += ADC_Read( channel );
    return (uint16_t)( result / average );
}
```

Abbildung 4.6 ADC- Messung

```
void einschalten (void)
{
    PORTD |= (1<<PD4);           // Netzteil an bei Dämmerung
    _delay_ms(100);             // Verzögerung für Relais
    while (i<32)
    {
        i++;                    // hochdimmen auf Dauerleuchtstärke
        OCR0A = i;
        OCR1A = i;
        _delay_ms(30);
    }
}
```

Abbildung 4.7 Einschaltfunktion der LEDs

Die beiden Funktionen zum Dimmen werden immer dann benötigt, wenn ein Signal vom Bewegungsmelder kommt. Die Funktionen zum Dimmen sind ähnlich aufgebaut wie die Einschaltfunktion. Allerdings wird hier der Compare-Wert bis 255 hoch gezählt bzw. von 255 auf 32 herunter gezählt. Die Zeitfunktion läuft in einer Interrupt Service Routine (ISR).

Die ISR wird aufgrund der CPU-Frequenz und des eingestellten Vorteilers 76 mal in der Sekunde aufgerufen. In dieser werden kontinuierlich die Sekunden hoch gezählt, bis diese den Wert 60000 erreichen. Danach wird der Zähler zurückgesetzt und der Vorgang beginnt von Neuem. Des Weiteren werden in dieser ISR die Schaltzustände von drei Schiebeschaltern abgefragt, mit denen bestimmte Zeiten vorgegeben werden können. Dieser Wert (0...7) wird mit einem Faktor beaufschlagt und dient dann als Vergleichswert für das Ablaufen einer bestimmten Zeit. Diese Bedingungen sind so angelegt, dass sie nur einmal aufgerufen werden. Beim Durchlaufen einer Einschaltbedingung wird eine Variable um eins erhöht, was später als Kriterium für die Ausschaltbedingung notwendig ist. Beim Ausschalten wird diese Variable wieder auf Null zurückgesetzt. Dies soll vermeiden, dass beim Erfüllen einer Bedingung ständig hoch- und heruntergedimmt wird. Des Weiteren soll es möglich sein, mit einem vierten Schiebeschalter die LEDs dauerhaft auf 100 % zu betreiben. Zudem ist diese Einschaltbedingung unabhängig von den anderen drei Schaltern. Das heißt, es kann zu jedem Zeitpunkt auf dauerhaft 100 % umgeschaltet werden. Der Aufbau der ISR ist in Abb. 4.8 dargestellt.

```

// ISR zum Erzeugen von verschiedenen Zeiten
ISR (TIMER2_COMPA_vect)
{
    tick++;
    if(tick == 76) // 76*13,1 = 1s
    {
        Wartefaktor = (PIND & 0x07); // Schalterkombination auslesen und übergeben
        s++; // Sekunden hoch zählen bis 60000
        tick = 0;
        if(s == 60000)
        {
            s = 0;
        }
    }
}

```

Abbildung 4.8 Aufbau der Zeitfunktion innerhalb der ISR

4.2.4 Ein- und Ausschalten anhand der Lichtverhältnisse

Das Ermitteln der Lichtverhältnisse erfolgt anhand des Auslesens der ADC-Werte, welche vom Phototransistor kommen. Weiterhin wird der ADC-Wert ausgelesen, der mit dem Potentiometer eingestellt werden kann. Direkt nach dem Auslesen werden beide Werte miteinander verglichen. Der Vergleich wird in einer *if-Bedingung* abgearbeitet. Nach einer ersten Messung wird eine Warteschleife von drei Sekunden durchlaufen und danach die Messung erneut durchgeführt. Dies dient zur Sicherheit, da sich die Umgebungshelligkeit nicht sprunghaft ändern kann. Sobald der Wert vom Sensor kleiner ist als der vom Potentiometer eingestellte Wert, wird die Spannungsversorgung der LEDs (36 V Netzteil) über zwei Relais eingeschaltet. Der Wert vom Potentiometer dient somit als Referenzwert, mit dem eine Helligkeitsschwelle vorgegeben werden kann. Mit den aufgenommenen Werten wird außerdem eine Mittelwertbildung aus zehn Werten durchgeführt, um die Messgenauigkeit zu erhöhen. Nach dem Einschalten wird noch einmal eine Warteschleife von 100 Millisekunden durchlaufen, um die Trägheit der Relais und des Netzteils zu berücksichtigen. Erst nach Ablauf dieser Zeit sollen die LED-Treiber über ein PWM-Signal angesteuert werden und die LEDs hochdimmen. Ein kurzer Programmausschnitt ist in Abb. 4.9 dargestellt.

```
// Einlesen der Werte von Phototransistor und Poti und Vergleich beider Werte
if ( (ADC_Read_Avg(2, 10) < ADC_Read_Avg(3, 5)) & (x<1) )
{
    long_delay(3000); // 3 Sekunden warten
    if (ADC_Read_Avg(2, 10) < ADC_Read_Avg(3, 5) // erneuter Vergleich → OK
        {
            einschalten();
            x = 1;
        }
}
```

Abbildung 4.9 Einschalten der LEDs bei Erreichen der Dämmerung

An zwei separaten Ausgängen des Mikrocontrollers wird ein PWM-Signal ausgegeben, mit dem die LEDs eingeschaltet und auf eine bestimmte Helligkeit gedimmt werden. Mit dieser Helligkeit sollen die LEDs dauerhaft betrieben werden, solange keine Bewegung registriert wird. Diese Helligkeit wird durch den Compare-Wert bestimmt, welcher auf 32 festgesetzt ist.

Aus diesem Wert lässt sich nach Formel [16] die Einschaltzeit der LEDs berechnen und nach Formel [17] die Leistungsaufnahme im gedimmten Betrieb. Der Timer/Counter zum Erzeugen der PWM arbeitet hierbei mit 8 Bit.

$$t_{\text{ges}} = T = \frac{1}{f_{\text{PWM}}} \quad [15]$$

$$t_{\text{ges}} = \frac{1}{612,7\text{Hz}} = 1,63\text{ms}$$

$$\frac{\text{Compare}}{2^8} = \frac{t_{\text{ein}}}{t_{\text{ges}}} \rightarrow t_{\text{ein}} = t_{\text{ges}} \cdot \frac{\text{Compare}}{2^8} \quad [16]$$

$$t_{\text{ein}} = 1,63\text{ms} \cdot \frac{32}{256} = 0,2\text{ms}$$

$$P_{\text{ges}} = P_{\text{LED}} \cdot \frac{t_{\text{ein}}}{t_{\text{ges}}} \quad [17]$$

$$P_{\text{ges}} = 40\text{W} \cdot \frac{0,2\text{ms}}{1,63\text{ms}} = 5\text{W}$$

Sollte sich bei abschließenden Messungen herausstellen, dass die Leuchtstärke der LEDs mit diesem eingestellten Compare-Wert nicht ausreichend ist, so müsste dieser Wert noch einmal nach oben korrigiert werden.

Der Compare-Wert wird in einer Zählfunktion kontinuierlich, bis zum festgelegten Wert erhöht. Das Hochzählen erfolgt in Schritten von $50 \mu\text{s}$, um die LEDs nicht zu schnell hoch zu dimmen. Dabei wird der Compare-Wert bei jedem Zählschritt an das jeweilige Compare-Register übergeben. Für das Ausschalten wird ein ähnlicher Vergleich gemacht, wie beim Einschalten. Allerdings muss zu diesem Zweck der aufgenommene Wert des Phototransistors größer sein, als der eingestellte Wert am Potentiometer. Auch hier wird nach einer ersten Messung eine Warteschleife von drei Sekunden durchlaufen und danach noch eine Vergleichsmessung durchgeführt. Allerdings muss hier auf den Messwert des Potentiometers noch ein Wert in Höhe von 32 aufaddiert werden. Dies ist notwendig um eine Hysterese zu schaffen, d.h. ein Bereich, in dem nichts getan wird. Dadurch soll verhindert werden, dass bei einer Übereinstimmung beider Werte die Schaltung zwischen zwei Zuständen hin- und herkippt. In einem solchen instabilen Zustand würden die LEDs ständig an- und ausgehen. Sobald die Ausschaltbedingung erfüllt ist, wird in einer Zähl Schleife der Compare-Wert bis auf Null heruntergezählt. Die Schrittbreite beträgt hierbei $20 \mu\text{s}$ und beim Erreichen des Compare-Wertes Null sind die LEDs aus.

Erst danach wird die Spannungsversorgung der LED-Treiber (36 V Netzteil) ausgeschaltet. Ein Ausschnitt aus dem Quellcode, zum Ausschalten ist in Abb. 4.10 dargestellt.

```
if (ADC_Read_Avg(2, 10) > (ADC_Read_Avg(3, 5) +32)) // Tageslicht erreicht
{
    long_delay(3000); // 3 Sekunden warten
    if (ADC_Read_Avg(2, 10) > (ADC_Read_Avg(3, 5) +32)) // erneuter Vergleich
    {
        ausschalten();
        x = 0;
    }
}
```

Abbildung 4.10 Ausschalten der LEDs bei ausreichender Helligkeit

4.2.5 Erhöhen der Beleuchtungsstärke beim Registrieren einer Bewegung

Über einen Bewegungsmelder sollen Bewegungen wahrgenommen und darauf reagiert werden. Sobald sich ein Fußgänger nähert sollen die LEDs auf ihre volle Leuchtstärke hochgedimmt werden. Solange die Bewegung bestehen bleibt, werden die LEDs mit 100 Prozent Leistung betrieben. Wird keine Bewegung mehr registriert, werden die LEDs wieder heruntergedimmt.

Die Ausschaltfunktionen der LED-Steuerung werden durch diese Funktion nicht beeinflusst, da diese immer vorrangig sind. Da der Bewegungsmelder an einem digitalen Eingang des Mikrocontrollers hängt, muss hier lediglich überprüft werden, ob an diesem Pin ein Eins-Signal ankommt oder nicht. Der ADC wird für diese Funktion nicht benötigt. Das Dimmen funktioniert wie bei der Einschaltfunktion mit einer Zählschleife, in welcher der Comparewert erhöht, oder verringert wird. In Abb. 4.11 ist der Ausschnitt aus dem Quellcode zu sehen, mit dem diese Funktion aufgerufen wird.

```
if ( (PINC) & (1 << PINC5) ) // Bewegungsmelder Signal 1
{
    hochdimmen();
    a = 1;
}
```

Abbildung 4.11 Hochdimmen beim Erkennen einer Bewegung

4.2.6 Einstellen verschiedener Zeiten mittels Schiebeschalter

Zum Einstellen verschiedener Zeiten, in der die LEDs auf 100 Prozent betrieben werden sollen, wurden vier Schiebeschalter auf der Platine vorgesehen. Daher war es notwendig, diese Schiebeschalter mit in die Software einzubeziehen. Zuerst müssen die internen Pullup-Widerstände des Mikrocontrollers hin zu den vier Schaltereingängen eingeschaltet werden, was in Abb. 4.2 zu sehen ist. Die Abfrage der Schalterzustände erfolgt in einer ISR, was in Kapitel 4.2.3 näher erläutert wird. Nach dem Hochdimmen wird der Sekundenzähler auf Null zurückgesetzt und beginnt von neuem hochzuzählen. Danach wird der Wert, welcher aus den Schalterzuständen resultiert, an eine *switch/case*-Anweisung übergeben. In dieser Anweisung ist jeder Schalterkombination ein Zeitwert in Sekunden zugewiesen. Nun wird dieser Wert mit dem Sekundenwert der ISR verglichen und bei Übereinstimmung wird die Funktion zum Herunterdimmen aufgerufen. Somit sind je nach Schalterkombination Zeiten von 2 s bis 30 min möglich. Für größere Zeiten kann die Dauerfunktion mit Schalter vier gewählt werden. In Abb. 4.12 ist die Bedingung zum Herunterdimmen dargestellt und in Abb. 4.13 die Funktion, welche in diesem Moment aufgerufen wird.

```
if ( (s == (zeit)) & (a == 0) & (b == 0) ) // Wartezeit abgelaufen
{
    runterdimmen();
}
```

Abbildung 4.12 Ablauf der Wartezeit und Funktionsaufruf

```
void runterdimmen (void)
{
    while ( i>32 )
    {
        i-- ; //herunterdimmen auf 12,5%
        OCR0A = i;
        OCR1A = i;
        _delay_ms(20);
    }
}
```

Abbildung 4.13 Funktion zum Herunterdimmen, auf 12,5 %

Die Schalterkombinationen folgen einem logischen Bitmuster von 0...15. Da aber Schalter 4 ausschließlich für die Dauerfunktion verwendet wird, gehen diese Zahlen nur von 0...7, woraus sich acht verschiedene Kombinationsmöglichkeiten und somit acht Zeiten ergeben. Alle Kombinationsmöglichkeiten und die dazugehörigen Zeiten sind in Tabelle 4.1 aufgeführt.

Tabelle 4.1 Einstellen verschiedener Zeiten

Schaltzustand	Schalterkombination/binär	Zeit in s
0	0000	2
1	0001	5
2	0010	15
3	0011	30
4	0100	60
5	0101	120
6	0110	300
7	0111	1800

In Abb. 4.14 ist die *switch/case*-Anweisung dargestellt, in welcher die verschiedenen Zeiten übergeben werden.

```

switch (Wartefaktor) // eingestellte Zeitdauer für hochdimmen
{
    case 1: zeit = 300; //5 min
            break;
    case 2: zeit = 120; //2 min
            break;
    case 3: zeit = 60; //1 min
            break;
    case 4: zeit = 30; //30 s
            break;
    case 5: zeit = 15; //15 s
            break;
    case 6: zeit = 5; //5 s
            break;
    case 7: zeit = 2; //2 s
            break;
    default:zeit = 1800; //30 min (alle Schalter on)
}

```

Abbildung 4.14 Übergabe der verschiedenen Zeiten

5 Technische Messungen an der Leuchte

Bei diesen Messungen wurden sowohl elektrische als auch lichttechnische Messungen durchgeführt. Diese sollen Aufschluss geben über die Übereinstimmung von elektrischen Größen mit verschiedenen Datenblättern und über die Leistungsaufnahme verschiedener Bauteile, sowie der gesamten Leuchte. Des Weiteren soll überprüft werden in wie weit die Leuchte bestimmten lichttechnischen Forderungen gerecht wird.

5.1 Elektrische Größen

Beim Messen von Strom und Spannung wurde festgestellt, dass diese Werte nicht ganz mit den vorgegebenen Sollwerten übereinstimmen. Der Spannungsabfall an den einzelnen LEDs liegt 0,9 V unter den angegebenen 14 V aus dem Datenblatt. Aus diesem Grund liegt auch die Leistungsaufnahme der LEDs 0,57 W unter den geforderten 10 W, was sich somit auch auf die Lichtausbeute und den Lichtstrom auswirkt. Eine Übersicht der geforderten und der gemessenen Werte stellt Tabelle 5.1 zu sehen.

Tabelle 5.1 Leistungsaufnahme

el. Größen	Sollwerte	Istwerte
U_{LED}	14 V	13,1 V
I_{LED}	0,7 A	0,72 A
P_{LED}	10 W	9,43 W
P_{LED_ges}	40 W	37,73 W
η_{LED}	82 lm / W	82 lm / W
Φ_{LED}	820 lm	773,3 lm
η_{Buck}	95 %	94,75 %
$I_{Netzteil}$	1,1 A	1,1 A
$P_{Netzteil}$	42,1 W	39,82 W
$\eta_{Netzteil}$	87 %	86,57 %
U_{ges}	230 V	230 V
I_{ges}	0,21 A	0,21 A
P_{ges}	48,8 W	48,3 W
$U_{Netzteil}$	36 V	36,2 V

Mit der Leistungsaufnahme der LEDs lässt sich ein Lichtstrom von 773 lm je LED berechnen, was deutlich unter den geforderten 820 lm liegt. Somit ergibt sich für die gesamte Leuchte ein Lampenlichtstrom von 3093 lm. Dies ergibt eine Differenz von 187 lm gegenüber den erwarteten 3280 lm. Mit den Wirkungsgraden des Abwärtswandlers und des Netzteils lässt sich die zu erwartende Leistungsaufnahme der Leuchte berechnen. Verglichen mit der tatsächlichen Leistungsaufnahme, welche sich aus Netzspannung und Eingangsstrom errechnet, sind nur geringe Abweichungen festzustellen. Aus diesen Werten ergibt sich für die gesamte Leuchte eine Leistungsaufnahme von 48,3 W, was einem Gesamtwirkungsgrad von 81,3 % entspricht. Daraus ist zu erkennen, dass die errechneten Wirkungsgrade des Abwärtswandlers und des Netzteils sehr gut mit den Wirkungsgraden aus den Datenblättern übereinstimmen. Diese Werte wurden ausschließlich mit einem Digitalmultimeter gemessen. Um exakte Spannungswerte zu erhalten, wurden diese zusätzlich noch einmal mit einem Digital-Speicher-Oszilloskop (DSO) gemessen. Zunächst wurde die Ausgangsspannung der beiden LED- Treiber im gedimmten Betrieb und auch bei 100 % gemessen, was in Abb. 5.1 zu sehen ist.

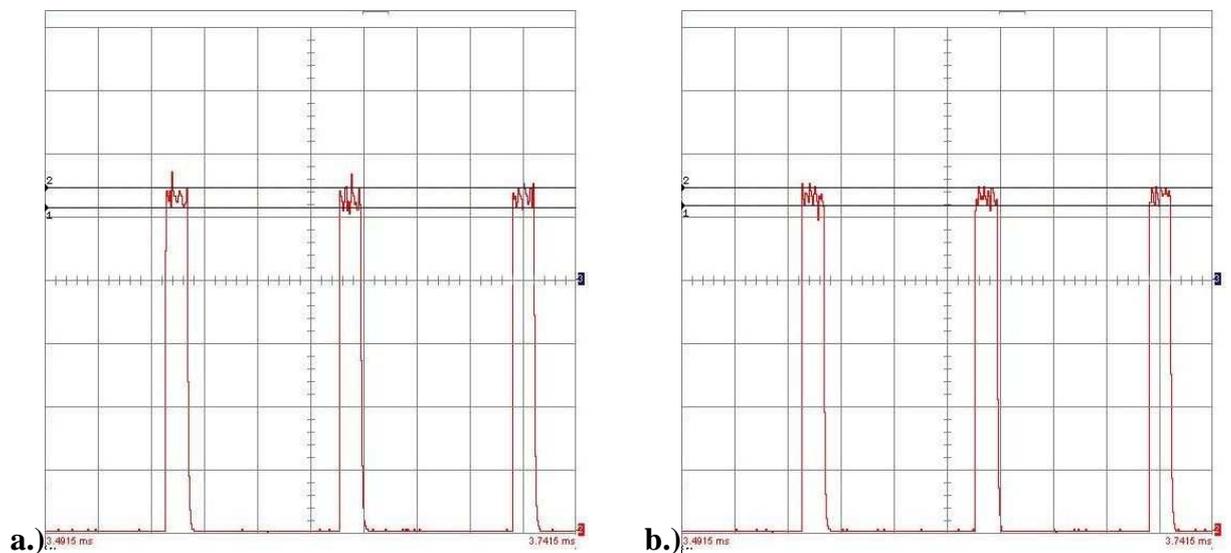


Abbildung 5.1 Ausgangsspannung LED- Treiber 1 a.) und 2 b.), gedimmt auf 12,5 %

Eingestellt sind hier eine Auflösung von 5 V/Div und eine Zeiteinteilung von 1 ms/Div. Es ist zu erkennen, dass die beiden Ausgangsspannungen sehr gut übereinstimmen. Dies garantiert, dass alle LEDs im gedimmten Betrieb annähernd gleich hell sind. Des Weiteren ist zu erkennen, dass die Spannung um einen Wert von 1,5 V schwankt. Dies ist darauf zurückzuführen, dass die LED- Treiber ohne Ausgangskondensatoren betrieben werden, welche diese pulsierende Gleichspannung noch glätten würden. Die Ausgangsspannungen der beiden LED-Treiber im nicht gedimmten Betrieb (100 %) sind in Abb. 5.2 zu sehen.

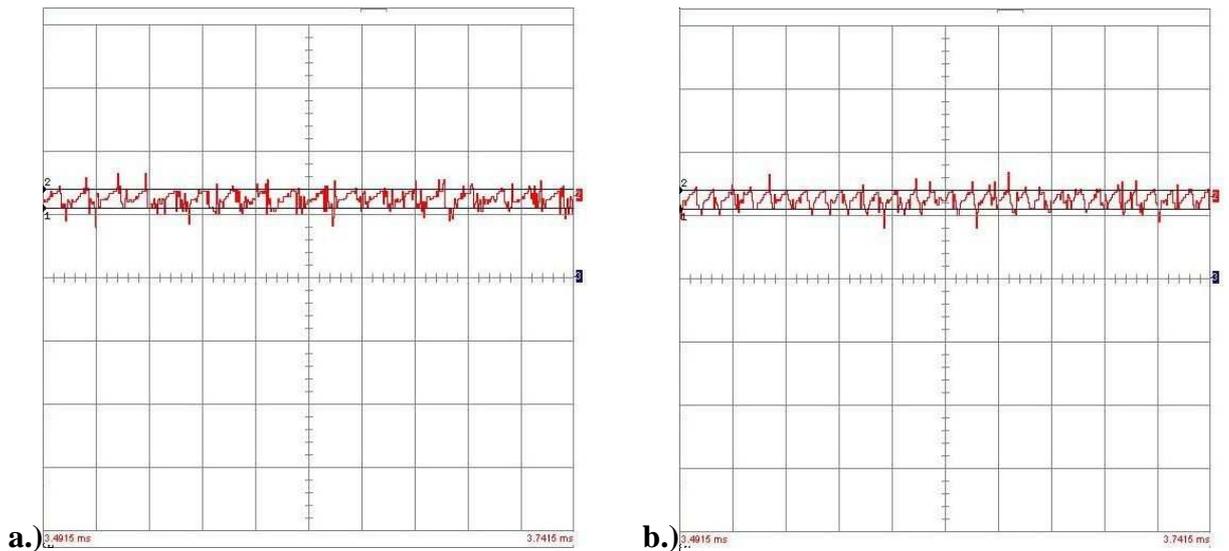


Abbildung 5.2 Ausgangsspannung LED- Treiber 1 a.) und 2 b.), bei 100 %

Die Auflösung ist hier die Gleiche, da die Höhe der Spannung immer gleich und auch die Zeiteinteilung unverändert ist. Auch hier ist zu erkennen, dass beide Spannungen sehr gut miteinander übereinstimmen und der Signalpegel um 1,5 V schwankt. Des Weiteren ist festzustellen, dass die aufgenommenen Verläufe mit denen aus dem Datenblatt übereinstimmen. Bei den Spannungen, die direkt an den LEDs gemessen wurden, ist eine Abweichung von 0,4 V gegenüber den Messwerten vom Digitalmultimeter festzustellen. Somit wurde hier ein Spannungswert von 13,5 V gemessen. Die Spannungsschwankung beträgt hier 0,8 V, was ungefähr der Hälfte der gesamten Spannungsschwankung am LED-Treiber Ausgang entspricht. Es ist zu erkennen, dass an allen vier LEDs die gleiche Spannung abfällt und somit auch die Abweichung zu den 14 V aus dem Datenblatt immer gleich ist. Die zeitlichen Spannungsverläufe von LED 1 und LED 3 sind in Abb. 5.3 dargestellt.

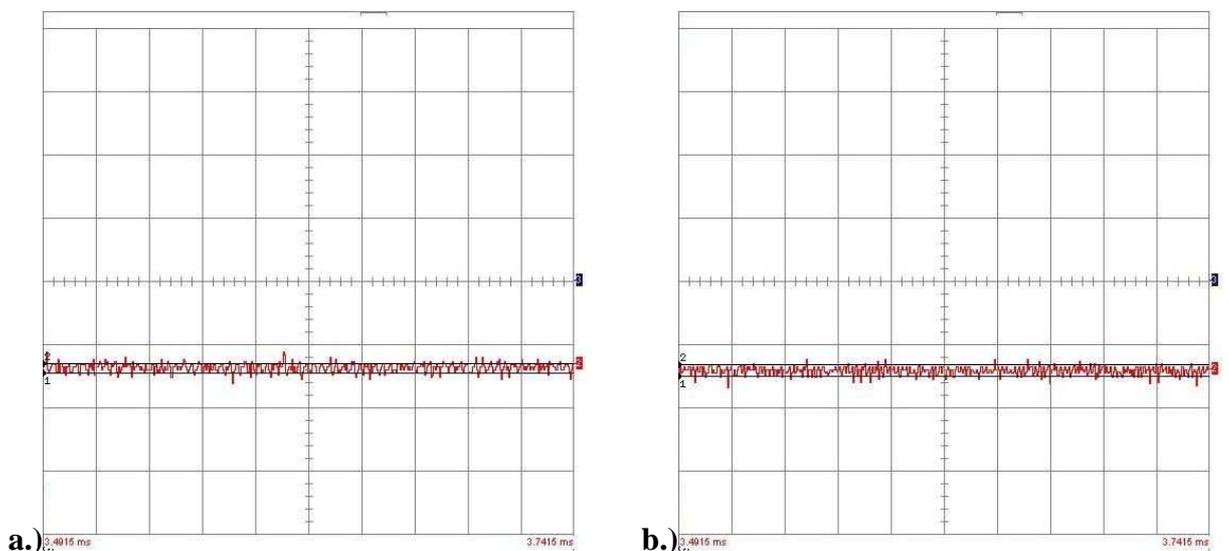


Abbildung 5.3 Spannungsverlauf an LED 1 a.) und LED 3 b.)

Zusätzlich wurde die Spannung an den PWM- Ausgängen des Mikrocontrollers gemessen. Diese beträgt 5 V, was der Betriebsspannung des Mikrocontrollers entspricht. An diesem Spannungsverlauf ist ebenfalls eine geringe Spannungsschwankung festzustellen, welche 0,3 V beträgt. Dies ist darauf zurückzuführen, dass die Betriebsspannung des Controllers schon mit dieser Welligkeit beaufschlagt ist. Die Spannungsverläufe bei 12,5 % PWM und bei 100 % sind in Abb. 5.4 zu sehen.

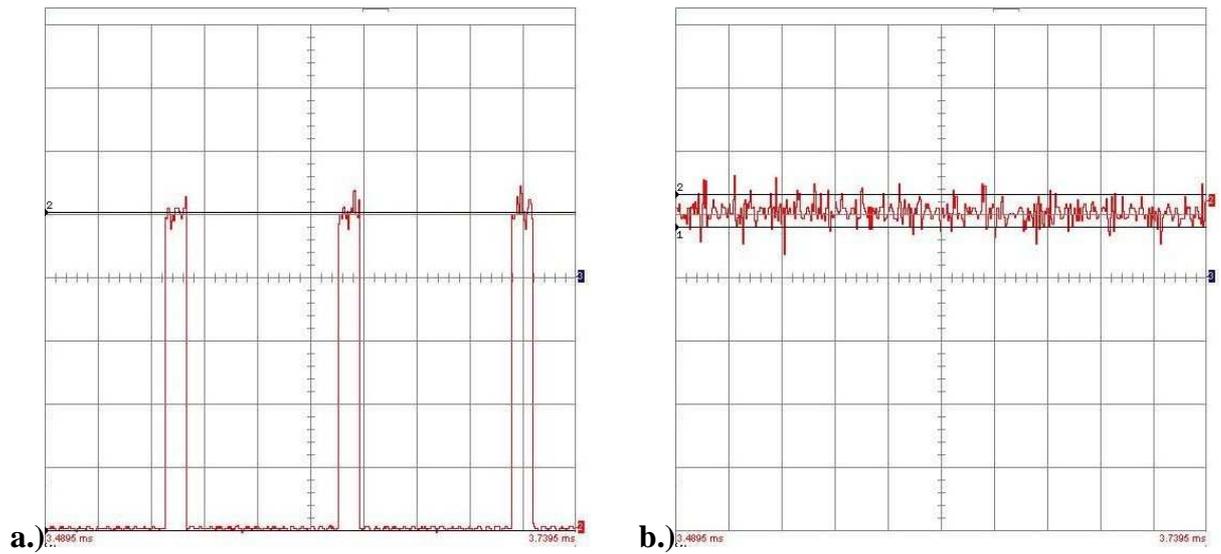


Abbildung 5.4 Spannung am PWM- Ausgang des Mikrocontrollers, a.) 12,5 %, b.) 100 %

Bei dieser Messung waren eine Auflösung von 1 V/Div und eine Zeiteinteilung von 1 ms/Div eingestellt.

5.2 Thermische Größen

Um das Temperaturverhalten bestimmter Bauteile beurteilen zu können, wurden verschiedene thermische Messungen durchgeführt. Hierbei wurden die Temperaturverläufe der beiden Leistungstransistoren und des Kühlkörpers über einen Zeitraum von einer Stunde aufgenommen. Zudem war es notwendig, die Temperaturerhöhung der LEDs innerhalb dieses Zeitraums festzustellen. Diese Messung sollte Aufschluss darüber geben, ob der verwendete Kühlkörper ausreichend dimensioniert ist und somit die LEDs ihre vorgeschriebene Maximaltemperatur aus dem Datenblatt nicht überschreiten. Die Messungen wurden mit einem Infrarot-Messgerät in einem Zeitintervall von zwei Minuten durchgeführt. Die aufgenommenen Temperaturkurven sind in Abb. 5.5 dargestellt.

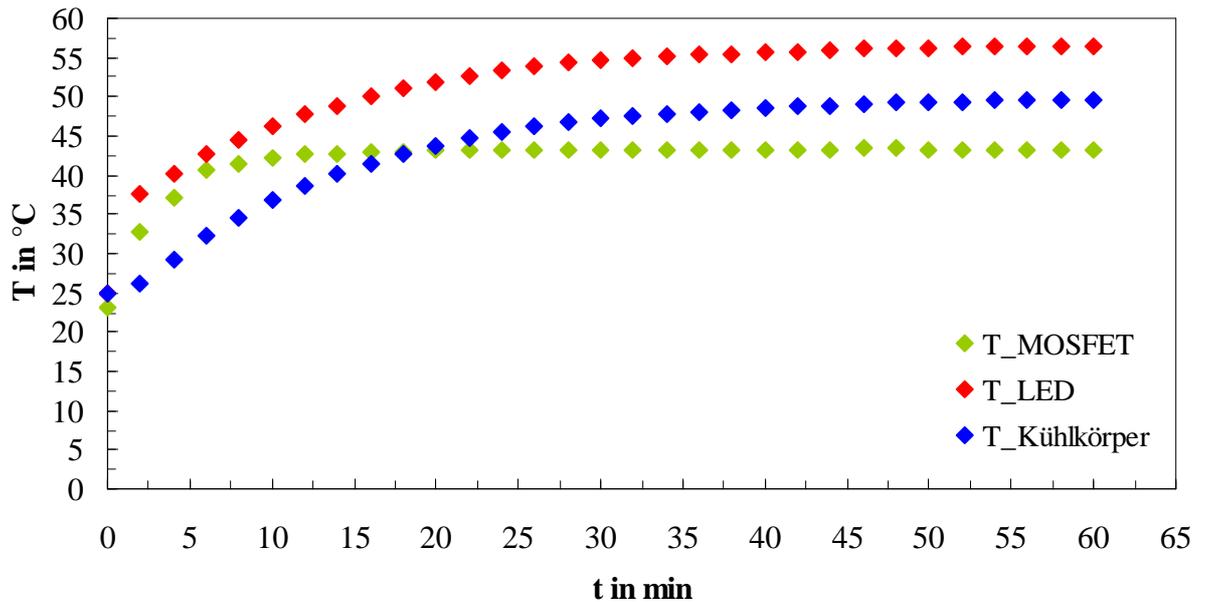


Abbildung 5.5 Temperaturverläufe an der Leuchte

Es ist zu erkennen, dass sich die LEDs und der Kühlkörper gleichmäßig erwärmen. Dies lässt auf eine gute Wärmeleitung zwischen LED und Kühlkörper schließen. Weiterhin wird deutlich, dass nach einer Stunde die maximale Temperatur erreicht ist. Bei den LEDs beträgt sie 56,5 °C und beim Kühlkörper 49,5 °C. Während sich LED und Kühlkörper relativ langsam erwärmen, hat der MOSFET schon nach kurzer Zeit seine maximale Temperatur erreicht. Das ist darauf zurückzuführen, dass dieser keinen separaten Kühlkörper besitzt und somit nur die Kupferfläche, auf der er befestigt ist, als Kühlfläche dient. Die maximale Temperatur des MOSFET beträgt 43 °C, welche sich schon nach 10...15 min einpendelt. Ein zusätzlicher Kühlkörper ist hier nicht nötig, da die Betriebstemperatur des MOSFET, laut Datenblatt, bei -55...150 °C liegt. Bei LED und Kühlkörper stellt sich die maximale Temperatur erst nach ca. 45 min ein. Für einen Betrieb im Labor sind diese Temperaturen akzeptabel, da hier keine Umgebungstemperaturen von über 25 °C auftreten können. Für einen Betrieb im Freien sollte ein größerer Kühlkörper für die LEDs vorgesehen werden, da hier auch Umgebungstemperaturen von über 30 °C auftreten können. Außerdem liegt die gemessene Maximaltemperatur von 56,5 °C schon sehr nahe an der vorgeschriebenen Maximaltemperatur. Diese beträgt laut Datenblatt 65 °C. Für den Betrieb mit einem geschlossenen Gehäuse sollte zusätzlich ein Lüfter eingebaut werden, welcher die entstehende Wärme zusätzlich abführt. Um genauere Ergebnisse zu erhalten, wurden weitere Messungen mit mehreren Temperaturnehmern durchgeführt, welche an verschiedenen Stellen der Leuchte angebracht wurden.

Die Messungen wurden mit einem Datenlogger durchgeführt, welcher bis zu 16 verschiedene Temperaturwerte gleichzeitig aufnehmen kann. Allerdings wurden für diese Messung nur acht Temperatursensoren verwendet. Sie wurden an verschiedenen Stellen der Leuchte angebracht, um einen genauen Aufschluss über die Erwärmung des Gerätes im Ganzen zu erhalten. Einer der Sensoren wurde zusätzlich auf einem Leistungstransistor der LED- Treiberschaltung angebracht. In Abb. 5.6 ist die Verteilung der Temperatursensoren zu sehen.

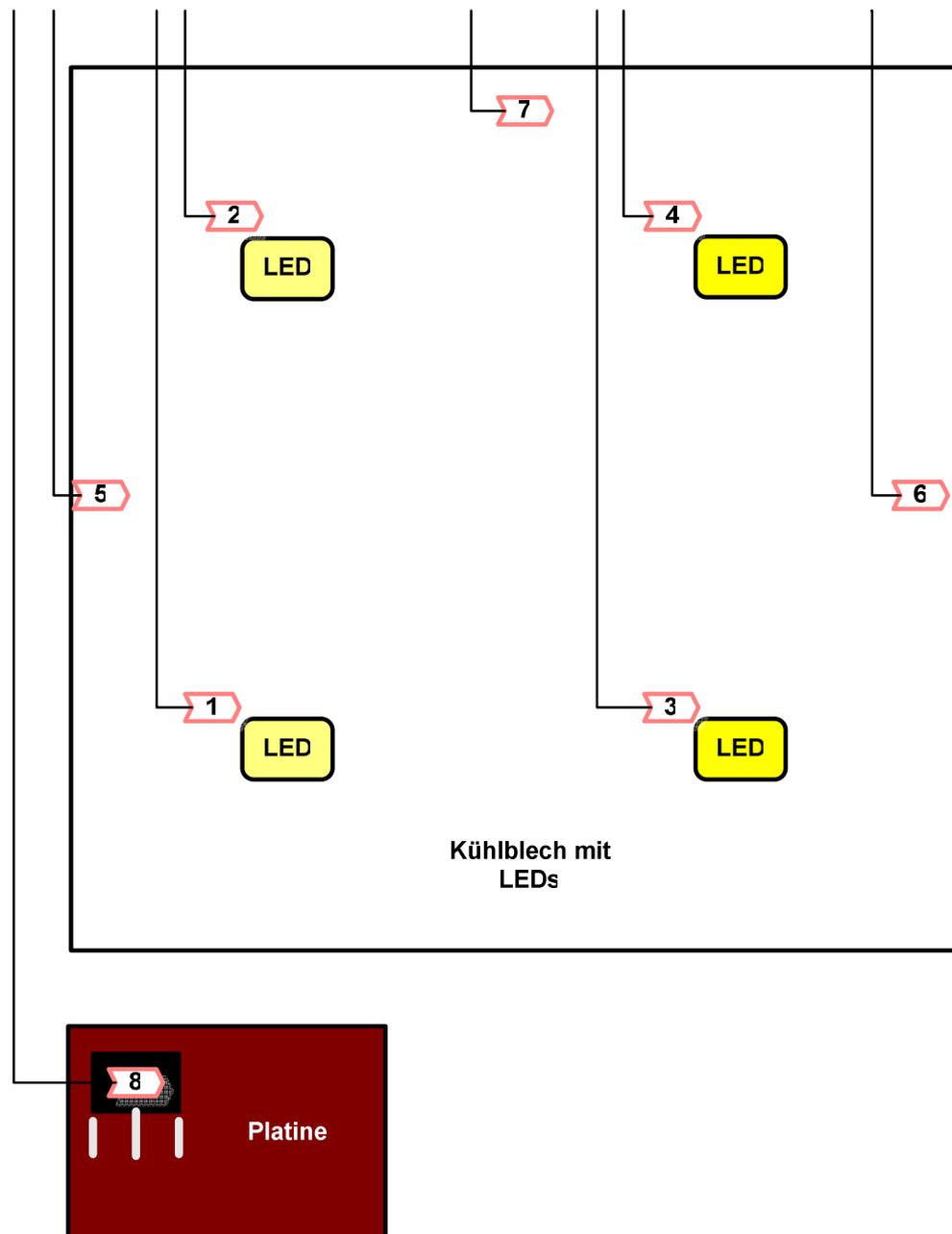


Abbildung 5.6 Verteilung der Temperatursensoren

Zuerst wurde die Erwärmung, bei 100 % Leistung, über einen Zeitraum von einer Stunde gemessen. Dabei konnte festgestellt werden, dass sich das Gerät in diesem Zeitraum auf seine Maximaltemperatur erwärmt. Die Erwärmung der Kühlfläche erfolgt an allen Stellen gleichmäßig und erreicht nach etwa 45 Minuten ihr Maximum. Die geringen Temperaturunterschiede zwischen den einzelnen Sensoren sind darauf zurückzuführen, dass sich die Sensoren nicht gleichmäßig auf dem Kühlblech aufkleben ließen. Bei der Erwärmung des MOSFET, auf der Platine, ist zu erkennen, dass dieser schon nach etwa 20 Minuten seine maximale Temperatur erreicht hat. Die Temperatur am MOSFET schwankt zwischen 43 °C und 50 °C. Dies lässt auf Messfehler schließen, welche durch die Schaltvorgänge am Transistor hervorgerufen werden. Die Maximaltemperatur des Kühlbleches beträgt 40 °C und die des MOSFET 50 °C. Um die Temperaturverläufe des Kühlbleches und des Transistors zu veranschaulichen, sind diese in Abb. 5.7 in Bezug auf die Kühlfläche dargestellt. Hierbei ist zu beachten, dass der Transistor sich nicht mit auf dem Kühlblech befindet. Dieser wurde lediglich in das Diagramm mit aufgenommen, um den Temperaturunterschied zu den Sensoren auf dem Kühlblech zu verdeutlichen.

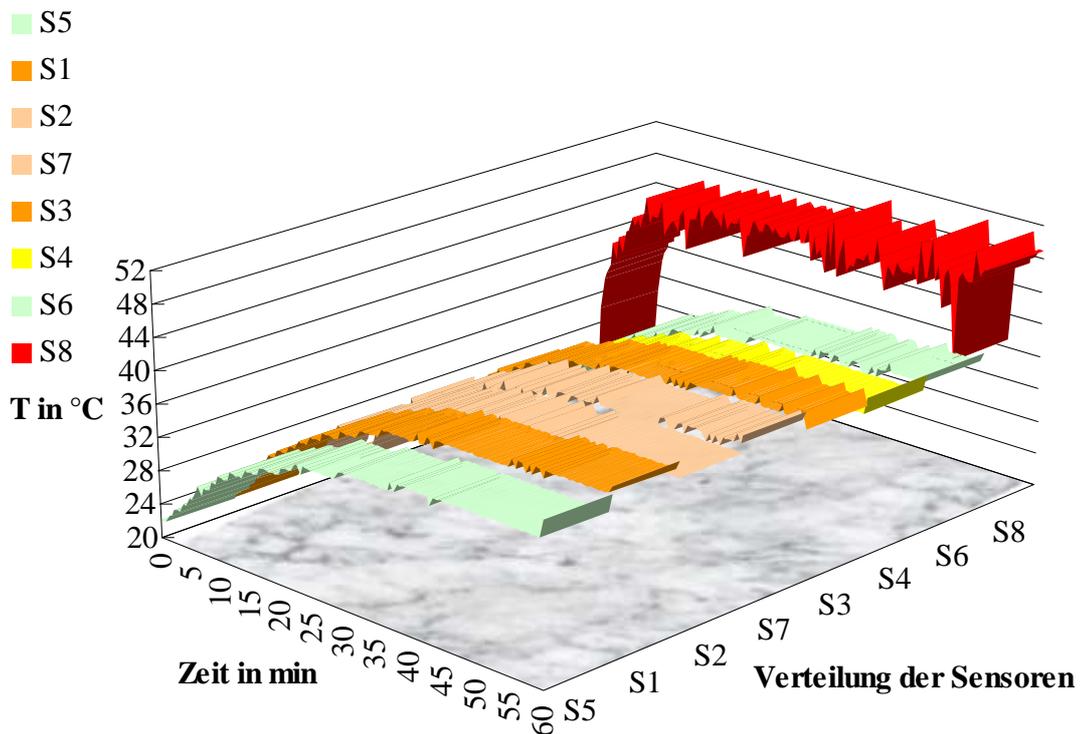


Abbildung 5.7 Temperaturverläufe von Kühlblech und MOSFET

Nachdem die Temperaturkurven im Erwärmungszeitraum aufgenommen waren, wurde noch einmal eine Temperaturmessung im erwärmten Zustand durchgeführt. Diese soll deutlich machen, dass auch nach einem Weiterbetrieb unter Volllast die Temperaturen konstant bleiben. Diese Messungen wurden ebenfalls über einen Zeitraum von einer Stunde und im Abstand von 30 Sekunden durchgeführt. Die aufgenommenen Messwerte, bezüglich des Kühlbleches sind in Abb. 5.8 zu sehen. Der Temperaturverlauf des Transistors wurde hierbei nicht mehr mit dargestellt, allerdings bleibt dieser ebenfalls im Temperaturbereich zwischen 43 °C und 50 °C konstant. Die Anordnung der Temperatursensoren entspricht in beiden Diagrammen der Anordnung in Abb. 5.6.

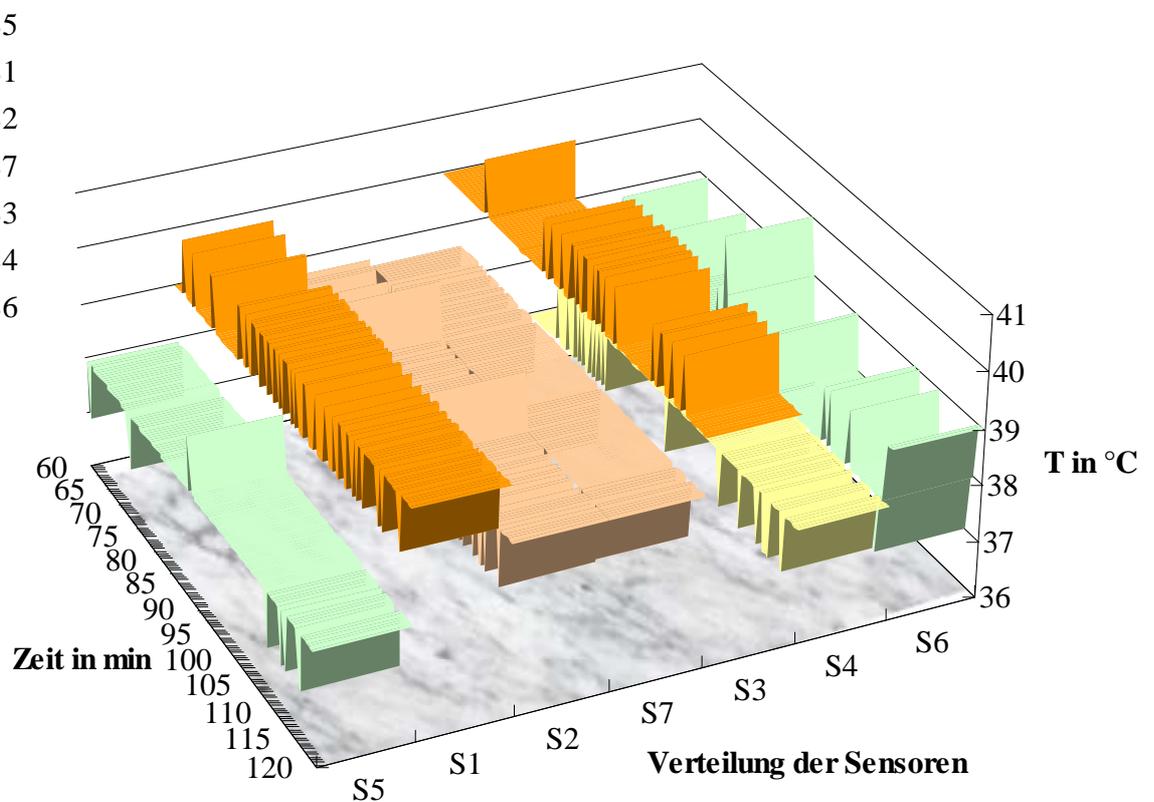


Abbildung 5.8 Temperaturverläufe nach einer Stunde Erwärmung

5.3 Lichttechnische Größen

Um festzustellen welche Lichtausbeute die LEDs im Ganzen erreichen und um zu überprüfen, ob die Lichtausbeute im gedimmten Betrieb ausreichend ist, wurden Lichttechnische Messungen durchgeführt. Zuerst wurde eine Leuchtdichtemessung durchgeführt, um festzustellen welche Leuchtdichte bei 100 % und im gedimmten Betrieb erreicht wird. Die aufgenommenen Bilder sind in Abb. 5.9 und die Messwerte in Abb. 5.10 dargestellt.

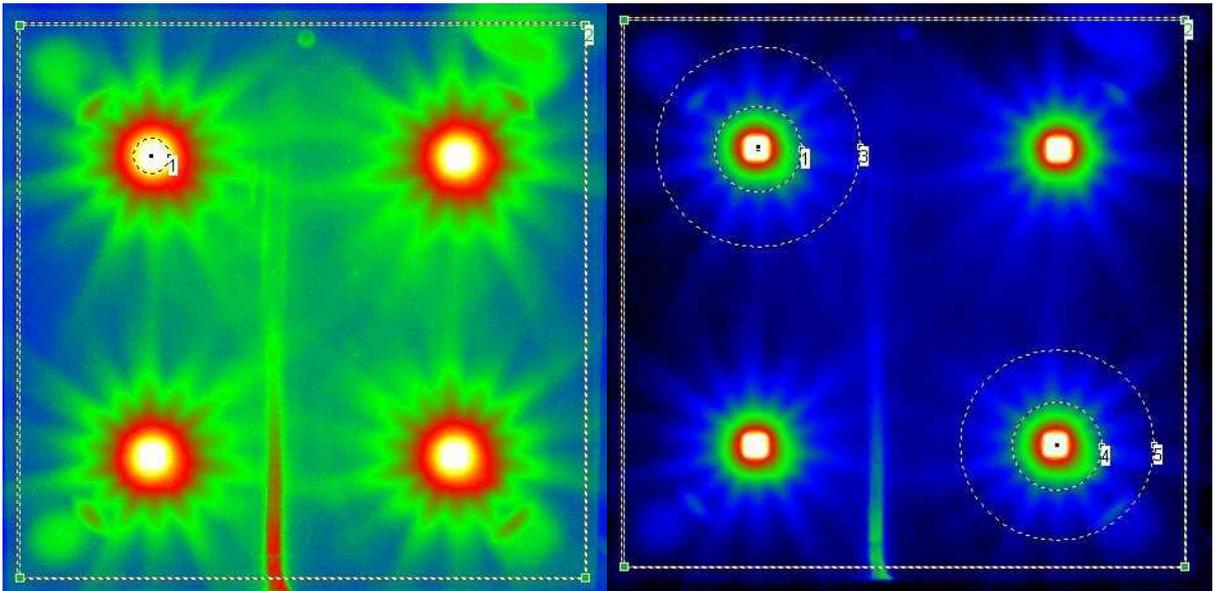


Abbildung 5.9 Leuchtdichte bei 100 % und auf 12,5 % gedimmt

Nr.	Quelle	Reg.	Klass.	Einheit	Anz	Mittel	Sigma	Min	Max	Fläche
1	Leuchtdichtebild 1		Standard	L-cd/m ²	1421	14420	6170	3142	19450	1421
2	Leuchtdichtebild 2		Standard	L-cd/m ²	431600	296,1	1782	17,64	19450	431600

Abbildung 5.10 Gemessene Leuchtdichte bei 100 % und bei 12,5 %

Anhand einer Leuchtdichteskala, welche in Abb. 5.11 dargestellt ist, lässt sich hieraus die Lichtfarbe der LEDs bestimmen. Diese ist bei den verwendeten LEDs kaltweiß. Es ist zu erkennen, dass alle vier LEDs annähernd die gleiche Leuchtdichte aufweisen.

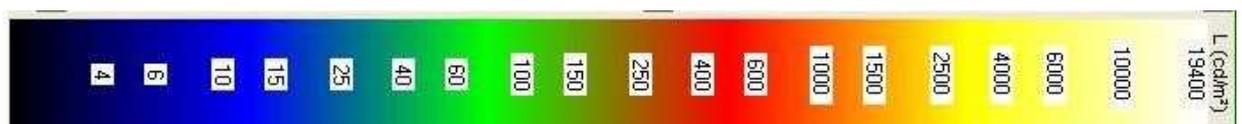


Abbildung 5.11 Leuchtdichteskala

Anschließend wurde eine Messreihe über die Leuchtstärkeverteilung der LEDs, im gedimmten und nicht gedimmten Betrieb aufgenommen. Die Leuchte wurde hierbei in einer Höhe von 2,15 m aufgehängt. Die Messung erfolgte in einer Höhe von 0,2 m über dem Boden und mit einem Rastermaß von 0,2x0,2 m. Der Messaufbau mit den Messpunkten ist in Abb. 5.12 dargestellt.

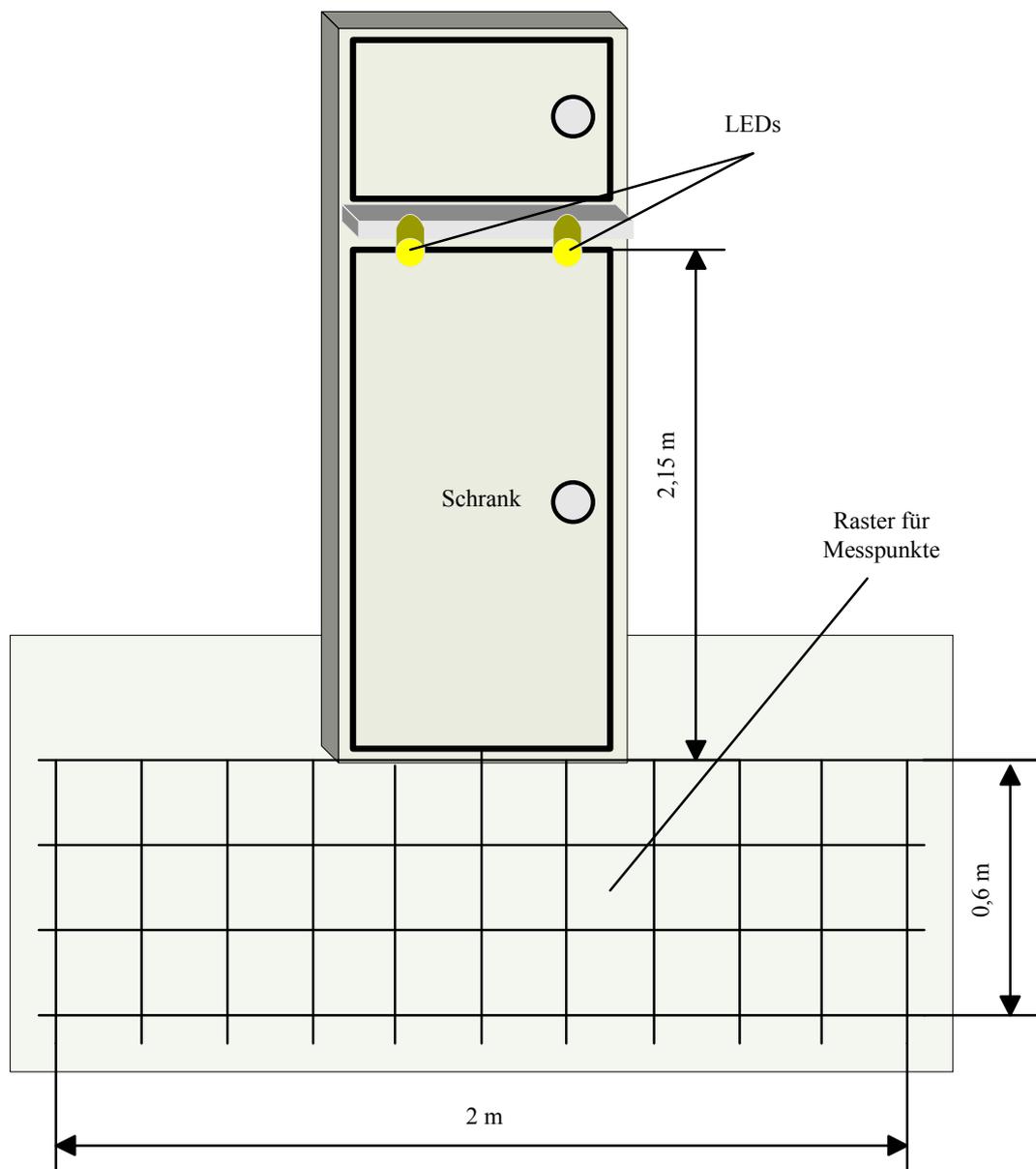


Abbildung 5.12 Messaufbau zur Leuchtstärkemessung

Aus Platzgründen wurde das Raster in vertikaler Richtung auf 0,6 m begrenzt. Die gewonnenen Messwerte geben jedoch genügend Aufschluss über die Leuchtstärke der LEDs. Diese Messung wurde erst nach einer Einschaltdauer von 30 min durchgeführt, um eventuelle Abweichungen der LEDs im erwärmten Zustand mit einzubeziehen.

Bei 100 % Leistung wird eine Beleuchtungsstärke von 338 lx erreicht und bei 12,5 % Leistung 44,4 lx. Somit ist nach EN 12464-1 die Beleuchtungsstärke im gedimmten Betrieb gerade noch ausreichend. Diese besagt, dass zur Orientierung und für den vorübergehenden Aufenthalt eine Nennbeleuchtungsstärke von 20...50 lx ausreichend ist, was in etwa der Beleuchtung eines Gehweges entspricht. Hierbei ist zu beachten, dass eine Gehwegleuchte im Freien nicht auf einer Höhe von 2,15 m hängen wird, sondern darüber (min. 3 m). Für einen Einsatz im Freien müsste somit eine erneute Leuchtstärkemessung in entsprechender Höhe durchgeführt werden. Für diesen Fall müsste wahrscheinlich die Leistung im gedimmten Betrieb erhöht werden. Die aufgenommenen Werte im gedimmten Betrieb sind in Abb. 5.13 zu sehen.

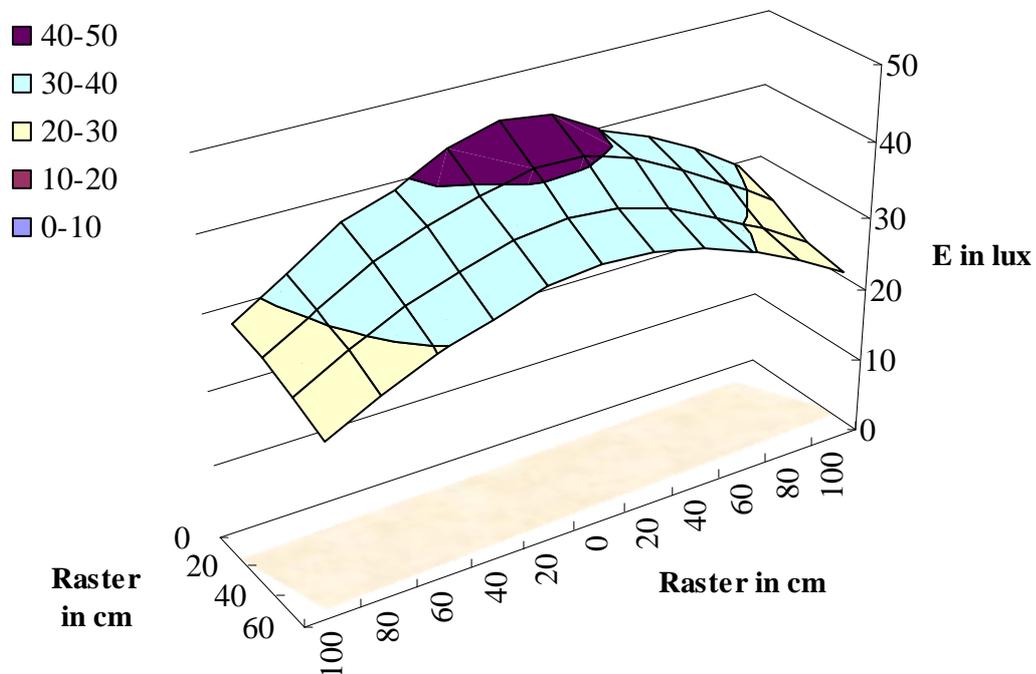


Abbildung 5.13 Beleuchtungsstärke im gedimmten Betrieb

Es ist zu erkennen, dass die Beleuchtungsstärke sehr gleichmäßig um die Lichtquelle verteilt ist und ebenso gleichmäßig mit der Entfernung zur Lichtquelle abnimmt. Dies lässt darauf schließen, dass alle vier LEDs einwandfrei funktionieren und somit die gleiche Lichtleistung erbringen. Auch bei einem Betrieb mit 100 % Leistung liefern die LEDs eine sehr gute Gleichmäßigkeit, was in Abb. 5.14 noch einmal dargestellt ist.

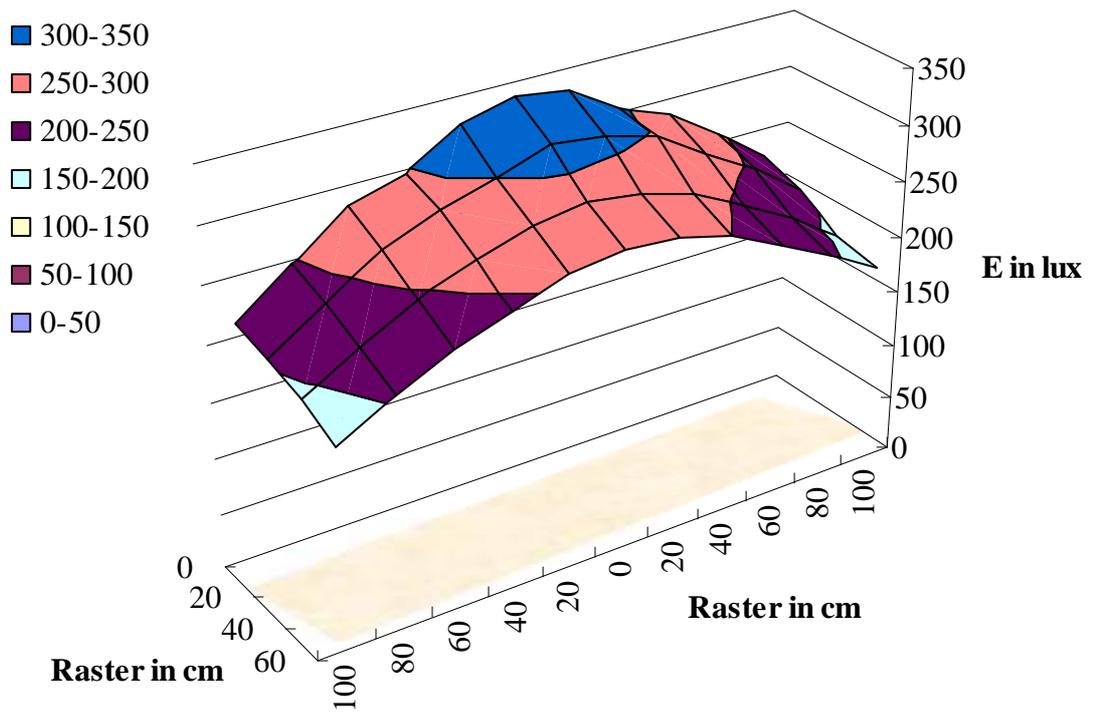


Abbildung 5.14 Beleuchtungsstärke bei 100 % Leistung

6 Zusammenfassung und Ausblick

Um Hochleistungs- LEDs stabil, stromgeregelt und unter verschiedenen Umweltbedingungen betreiben zu können, war es Ziel dieser Arbeit eine Ansteuerelektronik für Hochleistungs- LEDs zu entwickeln.

Als Grundlage diente ein Abwärtswandler der Firma *National Semiconductor*. Dieser sollte mit der richtigen Beschaltung vier Hochleistungs- LEDs betreiben. Die entwickelte und getestete Treiberschaltung ist für den Dauerbetrieb der verwendeten 10 W LEDs geeignet. In einem weiteren Schritt sollen die LED- Treiber über einen Mikrocontroller angesteuert werden, um somit einen energieeffizienten Betrieb zu realisieren. Der Mikrocontroller ermöglicht es die LEDs über einen Bewegungsmelder auf eine gewünschte Helligkeit zu dimmen. Des Weiteren ist es durch einen Mikrocontroller möglich, je nach Umgebungshelligkeit, die LEDs ein- bzw. auszuschalten. Die Ein- und Ausschaltswelle kann über ein Potentiometer eingestellt werden. Eine Notaus- Funktion mittels Temperaturüberwachung wurde ebenfalls realisiert. Die Temperaturmessung erfolgt über einen NTC- Widerstand und einen Spannungsteiler. Da diese Messmethode genau genug ist kann somit das aufwendige Abgleichen einer Messbrücke umgangen werden. Die aufgenommenen Analogwerte werden über die ADC- Eingänge des Mikrocontrollers in digitale Signale gewandelt und verarbeitet. Für den Betrieb im Labor wurden vier Schiebeschalter auf der Platine vorgesehen. Mit diesen Schiebeschaltern können verschiedene Zeiten eingestellt werden, in denen die LEDs auf 100 % hochgedimmt werden. Zum Schutz vor zu hohen Strömen wurde zwischen Netzteil und LED- Treiber eine Sicherung vorgesehen. Die komplette Schaltung einschließlich Platinenlayout wurde mit der Software EAGLE entworfen. Die Platine wurde selbst bedruckt, geätzt und bestückt, um somit Kosten zu sparen.

Die zweite Aufgabe bestand darin, eine Software zu entwickeln die den Ansprüchen an eine selbst gesteuerte Leuchte gerecht wird. Die Programmierung des Mikrocontrollers erfolgte in der Programmiersprache C und mit dem Programm AVR Studio 4, von ATMEL. Die Schwierigkeit bestand hierbei, mit vier Schiebeschaltern eine sinnvolle Zeitvorgabe zu realisieren. Somit kann nun mit drei von vier Schaltern eine Zeit von 5 s bis 30 min eingestellt werden und mit einem vierten Schalter dauerhaft auf 100 % gestellt werden. Mit dieser Funktion können zum Beispiel lichttechnische Messungen im Labor durchgeführt werden, oder das Temperaturverhalten, bestimmter Bauteile, unter Laborbedingungen festgestellt werden.

Somit wurde das Ziel erfüllt, eine Ansteuerelektronik für Hochleistung- LEDs zu entwickeln. Diese ist voll einsatzfähig und zur Integration in eine Leuchte geeignet, da sie den Anforderungen an eine Außenbeleuchtung gerecht wird. Weiterhin wird eine hohe Energieersparnis gegenüber herkömmlichen Leuchten erreicht, da die LEDs gedimmt betrieben werden. Auf Grundlage dieser Arbeit kann nun eine komplette Leuchte entwickelt werden.

Bei den bisherigen Entwicklungen handelt es sich lediglich um den elektrischen Teil einer Leuchte, ohne Gehäuse. Um dieses Gerät als vollwertige Leuchte, zum Einsatz im Freien, verwenden zu können, sind diverse Weiterentwicklungen unerlässlich. Zu diesem Zweck wurde ein Entwurf eines möglichen Gehäuses gefertigt. Dieser ist lediglich als Modell anzusehen. Außerdem wären in diesem Fall zusätzliche Kühlkörper notwendig und ein Lüfter, welcher die entstehende Abwärme aus dem Gehäuse schafft. Ferner müsste eine neue Software entwickelt werden und der Mikrocontroller neu programmiert werden. Im Zuge dessen müssten intensive Tests im Freien durchgeführt und ausgewertet werden. Somit wäre es möglich, auf Grundlage dieser Arbeit, eine vollwertige Straßenlampe zu entwickeln. Des Weiteren sind Tests hinsichtlich der EMV notwendig, um eventuelle Einflüsse bezüglich anderer Geräte zu vermeiden. Aufgrund der verwendeten Schaltnetzteile sind Untersuchungen hinsichtlich Netzrückwirkungen unerlässlich. Mit einer Weiterentwicklung der Software wären auch Fehlermeldungen und deren Aufschlüsselung denkbar.

Literaturverzeichnis

- [1] Ulrich Tietze / Christoph Schenk, *Halbleiter-Schaltungstechnik, 10. Auflage*, Springer-Verlag Berlin Heidelberg 1993 | S 563...569
- [2] Ulrich Tietze / Christoph Schenk, *Halbleiter-Schaltungstechnik, 11., völlig neu bearbeitete und erweiterte Auflage*, Springer-Verlag Berlin Heidelberg 1999, S 980...985, 1303...1306
- [3] Johann Siegel, *Schaltungstechnik – Analog und gemischt analog/digital*, Springer-Verlag Berlin Heidelberg 2009 | S 521, 615...618
- [4] PC & Elektronik, *Professionelle Schaltungstechnik Band 5*, Franzis' Verlag GmbH, 85586 Poing 2003
- [5] Joachim Specovius, *Grundkurs Leistungselektronik, 2. Auflage*, Vieweg & Sohn Verlag | GWV Fachverlage GmbH, Wiesbaden 2008 | S 87...94
- [6] Peter Monadjemi, *Jetzt lerne ich C!*, Markt und Technik, Buch- und Softwareverlag GmbH, München 1994
- [7] Günter Schmitt, *Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie*, Oldenbourgverlag München Wien 2005 | S 22, 33, 62...68, 170...172
- [8] Florian Schäfer, *Hardware und C- Programmierung in der Praxis, 1. Auflage*, Elektor Verlag GmbH, Aachen 2008 | S 119, 155...159, 180...182, 193...198
- [9] Ulrich Schlien, *Schaltnetzeile und ihre Peripherie, Dimensionierung, Einsatz, EMV, 4., korrigierte Auflage*, Vieweg + Teubner | GWV Fachverlag GmbH, Wiesbaden 2009 | S. 13...20, 282...286
- [10] National Semiconductor, *Datenblatt LM3409-HV*, 2010 | S. 1...7, 18...20,
- [11] HUEY JANN ELECTRONICS INDUSTRY CO., *Datenblatt High Power 10 W LED*, 2010 | S. 5...11, 16...20
- [12] Atmel Corporation, *Datenblatt ATmega 48V*, 2005 | S. 1..6, 81, 88, 89, 99...103, 106, 128...131, 149...154
- [13] Vishay Semiconductors, *Datenblatt Ambient Light Sensor*
- [14] PANASONIC, *Datenblatt Motion Sensor*, Matsushita Electric Works Ltd., 2006
- [15] Vishay BCcomponents, *Datenblatt NTC Thermistor*, 2005

Quellenverzeichnis

- [16] Led-info/das Rechercheportal, *Entwicklung der Halbleiterlichtemitter/LED Geschichte*, Hauke Haller 2000-2011 | gefunden Februar 2011 | <http://www.led-info.de/grundlagen/leuchtdioden/geschichte.html>
- [17] LIGHTFACTORY.EU, *Gechichte der LED/Funktionsweise der LED*, Copyright by LightFactory.eu 2004-2011 | gefunden Februar 2011 | <http://www.lightfactory.eu/Service/LED.htm>
- [18] AEK, *LED Straßenlampen Homepage*, gefunden März 2011 | <http://www.led-strassenbeleuchtung.net/>
- [19] H. Hochreutener, *LED- Treiber*, Züricher Hochschule für angewandte Wissenschaften/ SoE@ZHAW, Mai 2011 | gefunden Mai 2011 | <https://home.zhaw.ch/~hhrt/EK1/LED-Treiber.pdf>
- [20] Michael Wäste, *Elektronik: EAGLE-Tutorial: Bauteile selbst definieren*, gefunden Mai 2011 | http://www.amateurfunkbasteln.de/eagle_lib/eagle_lib.html
- [21] ATMEL, *AVR910: In-System Programming*, Atmel Corporatin 2008 | gefunden Mai 2011 | http://www.atmel.com/dyn/resources/prod_documents/doc0943.pdf
- [22] TRILUX, *Planungshilfe/Licht für Europas Straßen*, TRILUX-LENZ GmbH + Co KG April 2005 | gefunden Juni 2011 | http://www.trilux.de/tx/export/download/de.plan_net.trilux/Downloads/Lichtplanung/05_15-A-CH-D.pdf
- [23] schweizer-fn.de, *Wärmeübergangskoeffizienten*, Juli 2011 | gefunden August 2011 | http://www.schweizer-fn.de/stoff/wuebergang_gase/v2_wuebergang_gase.htm
- [24] -, *Pulsweitenmodulation*, 2007 | gefunden Mai 2011 | <http://www.strippenstrolch.de/1-3-13-pwm-pulsweitenmodulation.html>

Anhang

A1	Berechnungsformeln	XI
A1.1	LED- Treiber.....	XI
A1.2	Bauteile für OPV	XII
A1.3	Berechnung der Leistungsaufnahme im gediminten Betrieb.....	XIII
A1.4	Kühlkörperberechnung	XIII
A2	Bauteillisten	XIV
A2.1	Bauteile mit elektrischen Größen	XIV
A2.2	Bestellliste mit Bauteilgrößen.....	XV
A3	Schaltpläne	XVII
A3.1	LED- Treiberschaltung für Testplatine.....	XVII
A3.2	Mikrocontroller mit Programmierschnittstelle	XVII
A3.3	Komplette Schaltung für endgültige Platine.....	XVIII
A4	Quellcode zur Ansteuerung der LED- Treiber	XIX
A5	Gefertigte Platinen	XXVII
A5.1	Testplatine.....	XXVII
A5.2	Hauptplatine mit OPV	XXVII
A5.3	Hauptplatine entgültig.....	XXVIII
A6	Bilder	XXIX
A6.1	Leuchte mit Ansteuerelektronik und Netzteil von innen	XXIX
A6.2	Unterseite der Leuchte mit LEDs	XXX
A6.2	Komplette Leuchte mit möglichem Gehäuse.....	XXXI

A1 Berechnungsformeln

A1.1 LED- Treiber

Gegebene Größen:

$$f_{\text{SW}} = 525\text{kHz}$$

$$V_{\text{in}} = 36\text{V}, V_{\text{in_max}} = 75\text{V}$$

$$V_{\text{out}} = 28\text{V}$$

$$I_{\text{LED}} = 0,7\text{A}$$

$$\Delta i_{\text{LED_PP}} = \Delta i_{\text{L_PP}} = 1\text{A}$$

$$\Delta v_{\text{IN_PP}} = 1,44\text{V}$$

$$V_{\text{TURN_ON}} = 10\text{V}; V_{\text{HYS}} = 1,1\text{V}$$

$$C_{\text{off}} = 470\text{pF}$$

$$\eta = 0,95$$

$$D = \frac{V_{\text{out}}}{V_{\text{in}} \cdot \eta} = \frac{28\text{V}}{36\text{V} \cdot 0,95} = 0,82$$

Lösungsansatz:

$$R_{\text{off}} = \frac{-\left(1 - \frac{V_{\text{out}}}{\eta \cdot V_{\text{in}}}\right)}{(C_{\text{off}} + 20\text{pF}) \cdot f_{\text{SW}} \cdot \ln\left(1 - \frac{1,24\text{V}}{V_{\text{out}}}\right)} = \frac{-\left(1 - \frac{28\text{V}}{0,95 \cdot 36\text{V}}\right)}{470\text{pF} \cdot 525\text{kHz} \cdot \ln\left(1 - \frac{1,24\text{V}}{28\text{V}}\right)}$$
$$= 16,2\text{k}\Omega \Rightarrow \text{gewählt: } 16\text{k}\Omega$$

$$t_{\text{off}} = -(C_{\text{off}} + 20\text{pF}) \cdot R_{\text{off}} \cdot \ln\left(1 - \frac{1,24\text{V}}{V_{\text{out}}}\right) = -490\text{pF} \cdot 16,2\text{k}\Omega \cdot \ln\left(1 - \frac{1,24\text{V}}{28\text{V}}\right) = 359\text{ns}$$

$$L1 = \frac{V_{\text{out}} \cdot t_{\text{off}}}{\Delta i_{\text{L_PP}}} = \frac{28\text{V} \cdot 359\text{ns}}{1\text{A}} = 10,1\mu\text{H} \Rightarrow \text{gewählt: } 47\mu\text{H}$$

$$I_{\text{L_max}} = I_{\text{LED}} + \frac{\Delta i_{\text{L_PP}}}{2} = 0,7\text{A} + \frac{1\text{A}}{2} = 1,2\text{A}$$

$$R_{\text{SNS}} = \frac{V_{\text{ADJ}}}{5 \cdot I_{\text{L_max}}} = \frac{1,24\text{V}}{5 \cdot 1,2\text{A}} = 0,21\Omega \Rightarrow \text{gewählt: } 0,3\Omega$$

$$t_{\text{on}} = \frac{1}{f_{\text{SW}}} - t_{\text{off}} = \frac{1}{525\text{kHz}} - 359\text{ns} = 1,55\mu\text{s}$$

$$C_{\text{in_min}} = \frac{I_{\text{LED}} \cdot t_{\text{on}}}{\Delta V_{\text{IN_PP}}} = \frac{0,7\text{A} \cdot 1,55\mu\text{s}}{1,44\text{V}} = 1,1\mu\text{F} \Rightarrow \text{gewählt : } 2,2\mu\text{F}$$

$$R_{\text{UV2}} = \frac{V_{\text{HYS}}}{22\mu\text{A}} = \frac{1,1\text{V}}{22\mu\text{A}} = 50\text{k}\Omega \Rightarrow \text{gewählt : } 49,9\text{k}\Omega$$

$$R_{\text{UV1}} = \frac{V_{\text{ADJ}} \cdot R_{\text{UV2}}}{V_{\text{TURN_ON}} - V_{\text{ADJ}}} = \frac{1,24\text{V} \cdot 49,9\text{k}\Omega}{10\text{V} - 1,24\text{V}} = 7,06\text{k}\Omega \Rightarrow \text{gewählt : } 6,98\text{k}\Omega$$

$$I_{\text{T}} = \frac{V_{\text{out}} \cdot I_{\text{LED}}}{V_{\text{in}} \cdot \eta} = \frac{28\text{V} \cdot 0,7\text{A}}{36\text{V} \cdot 0,95} = 0,57\text{A}$$

$$I_{\text{T_RMS}} = I_{\text{LED}} \cdot \sqrt{D \cdot \left[1 + \frac{1}{12} \cdot \left(\frac{\Delta i_{\text{L_PP}}}{I_{\text{LED}}} \right)^2 \right]} = 0,7\text{A} \cdot \sqrt{0,82 \cdot \left[1 + \frac{1}{12} \cdot \left(\frac{1\text{A}}{0,7\text{A}} \right)^2 \right]} = 0,69\text{A}$$

$$P_{\text{T}} = I_{\text{T_RMS}}^2 \cdot R_{\text{DS(ON)}} = (0,67\text{A})^2 \cdot 190\text{m}\Omega = 85,3\text{mW}$$

\Rightarrow gewählt : Q1 = P_MOSFET mit $R_{\text{DS(ON)}} = 190\text{m}\Omega$, $U = 100\text{V}$, $I = 3,8\text{A}$

$$V_{\text{D_max}} = V_{\text{in_max}} = 75\text{V}$$

$$I_{\text{D}} = (1 - D) \cdot I_{\text{LED}} = (1 - 0,82) \cdot 0,7\text{A} = 126\text{mA}$$

$$P_{\text{D}} = I_{\text{D}} \cdot V_{\text{D}} = 126\text{mA} \cdot 750\text{mV} = 94,5\text{mW}$$

\Rightarrow gewählt : D1 = Schottky Diode mit $U = 100\text{V}$, $I = 3\text{A}$

A1.2 Bauteile für OPV

$$f_{3\text{db}} = \sqrt{\frac{\text{GBP}}{2 \cdot \pi \cdot R_1 \cdot C_1}} = \sqrt{\frac{1\text{MHz}}{2 \cdot \pi \cdot 1\text{M}\Omega \cdot 34\text{pF}}} = 68,4\text{kHz}$$

$$C_1 = \frac{R_{\text{D}}}{R_1} \cdot C_1 = \frac{50\text{M}\Omega}{1\text{M}\Omega} \cdot 34\text{pF} = 1,7\text{nF} \Rightarrow \text{gewählt : } 2,2\text{nF}$$

A1.3 Berechnung der Leistungsaufnahme im gedimmten Betrieb

$$f_{\text{PWM}} = \frac{f_{\text{CPU}}}{\text{Vorteiler}} = \frac{20\text{MHz}}{64} = 312,5\text{kHz}$$

$$t_{\text{ges}} = T = \frac{1}{f_{\text{PWM}}} = \frac{1}{312,5\text{kHz}} = 3,2\mu\text{s}$$

$$\frac{32}{2^8} = \frac{t_{\text{ein}}}{3,2\mu\text{s}} \rightarrow t_{\text{ein}} = \frac{32}{256} \cdot 3,2\mu\text{s} = 0,4\mu\text{s}$$

$$P_{\text{ges}} = P_{\text{LED}} \cdot \frac{t_{\text{ein}}}{t_{\text{ges}}} = 40\text{W} \cdot \frac{0,4\mu\text{s}}{3,2\mu\text{s}} = 5\text{W}$$

A1.4 Kühlkörperberechnung

$$P_{\text{Tot}} = P_{\text{konv.}} \approx 32\text{W}$$

$$R_{\text{thJC}} = 4 \frac{\text{K}}{\text{W}}, R_{\text{thCH}} = 0,1 \frac{\text{K}}{\text{W}}$$

$$\vartheta_{\text{max}} = 65^\circ\text{C}, \vartheta_{\text{A}} \approx 25^\circ\text{C}$$

$$c_{\text{AL}} = 920 \frac{\text{Ws}}{\text{kg} \cdot \text{K}}, \alpha \geq 23 \frac{\text{W}}{\text{m}^2\text{K}}$$

$$R_{\text{thJA}} = R_{\text{thJC}} + R_{\text{thCH}} + R_{\text{thHA}}$$

$$R_{\text{thJA_ges}} = \frac{R_{\text{thJC}} + R_{\text{thCH}}}{4} + R_{\text{thHA}}$$

$$\Delta\vartheta = \vartheta_{\text{max}} - \vartheta_{\text{A}} = 65^\circ\text{C} - 25^\circ\text{C} = 40\text{K}$$

$$\Delta\vartheta = P_{\text{Tot}} \cdot R_{\text{thJA_ges}} = P_{\text{Tot}} \cdot \left(\frac{R_{\text{thJC}} + R_{\text{thCH}}}{4} + R_{\text{thHA}} \right)$$

$$\rightarrow R_{\text{thHA}} = \frac{\Delta\vartheta}{P_{\text{Tot}}} - \frac{R_{\text{thJC}} + R_{\text{thCH}}}{4} = \frac{40\text{K}}{32\text{W}} - \frac{(4+0,1)\text{K}}{4\text{W}} = 0,225 \frac{\text{K}}{\text{W}}$$

$$R_{\text{thJA_ges}} = \frac{(4+0,1)\text{K}}{4\text{W}} + 0,225 \frac{\text{K}}{\text{W}} = 1,25 \frac{\text{K}}{\text{W}}$$

$$R_{\text{thJA_ges}} = \frac{\Delta\vartheta}{P_{\text{Tot}}} = \frac{1}{\alpha \cdot A} \rightarrow A = \frac{1}{\alpha \cdot R_{\text{thJA_ges}}} = \frac{1 \cdot \text{m}^2\text{KW}}{23\text{W} \cdot 1,25\text{K}} = 0,03478\text{m}^2$$

d.h. der Kühlkörper für alle 4 LEDs müsste eine Seitenlänge von rund 19 cm haben (vgl. [5])

A2 Bauteillisten

A2.1 Bauteile mit elektrischen Größen

Bauteil	Bezeichnung	el. Größen
Kondensator	C_F, C_F1	1 μ F, 16 V
Kondensator	C_OFF, C_OFF1	470 pF
Widerstand	R_OFF, R_OFF1	16 k Ω
Induktivität	L1, L2	47 μ H, 2,7 A
Widerstand	R_SNS, R_SNS1	0,3 Ω
Kondensator	C_IN1 ... C_IN4	2,2 μ F, 100 V
Widerstand	R_UV2, R_UV4	49,9 k Ω
Widerstand	R_UV1, R_UV3	6,98 k Ω
Transistor	Q1, Q2	100 V, 3,8 A
Schottky Diode	D1, D2, D3	40 V, 3 A
Buck - Converter	IC1, IC2	6 V...75 V, 1 A
Mikrocontroller	IC3	1,8 V...5,5 V, 8 Bit
Quarz	QUARZ	20 MHz, 18 pF
Kondensator	C1, C2	100 nF, 50 V
Transistor	Q3 / FDY301	4,5 V, 1 A
Induktivität	L3	1 μ H, 180 mA
Widerstand	R1, R7	100 k Ω
Widerstand	R2, R4, R5, R6	1 k Ω
Widerstand	R8	680 Ω
Widerstand	R9	220 k Ω
Kondensator	C3, C4	10 nF, 50 V
Z - Diode	D4, D5	5,6 V, 0,3 W
LED	LD1	2,1 V
Phototransistor		570 nm, 100 mW
Bewegungsmelder		3 V...6 V
NTC - Widerstand		2,2 k Ω
Relais	K1, K2	5 V, 6 A
IC Sockel		28 - ploig
Sicherungshalter		10 A, 250 V

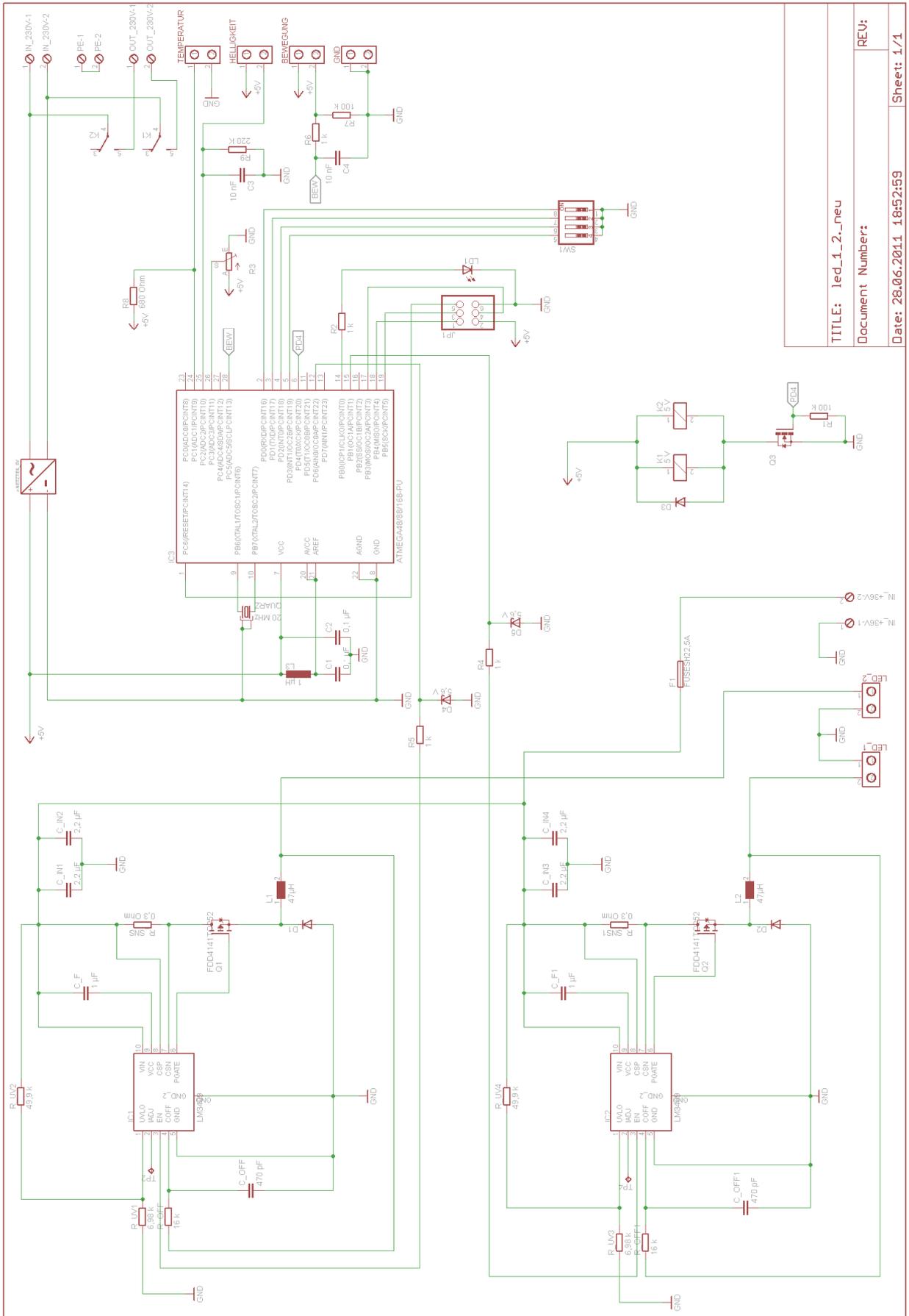
Bauteil	Bezeichnung	el. Größen
Feinsicherung	F1	3,15 A
Potentiometer	R3	10 k Ω
Anschlussklemmen		200 V, 10 A
Anschlussklemmen		250 V, 15 A
Stiftleiste		3A
Dip - Schalter	SW1	24 V, 50 mA
Netzteil		36 V, 1,4 A, 50 W
Netzteil	NETZTEL_5V	5 V, 0,55 A, 2,75 W
Platine		35 μ m Cu

A2.2 Bestellliste mit Bauteilgrößen

Bauteil	Stückzah	Bauteilgröße	Bestellnummer	Shop
Widerstand	1	0805	1099818	Farnell
Widerstand	2	0805	1576128RL	Farnell
Widerstand	2	1206	1653136	Farnell
Widerstand	2	1206	1100224RL	Farnell
Widerstand	2	0805	1653032	Farnell
Widerstand	4	0805	1099800RL	Farnell
Widerstand	1	0805	1099799RL	Farnell
Widerstand	2	2512	1107412	Farnell
Kondensator	4	1812	1838748	Farnell
Kondensator	2	0603	1650837	Farnell
Kondensator	2	0603	1692286RL	Farnell
Kondensator	2	0805	1692287RL	Farnell
Kondensator	2	0603	1650849	Farnell
Induktivität	2	SMD 12 x 12 mm	1635928	Farnell
Induktivität	2	0805	1704104	Farnell
Schottky Diode	3	SOD123	1829202	Farnell
Z - Diode	2	SOT23	1466617RL	Farnell
LED	1	1206	1581232	Farnell
Transistor	2	D-PAK_TO252AA	1843773	Farnell

Bauteil	Stückzah	Bauteilgröße	Bestellnummer	Shop
Transistor	1	SC-89	1495198	Farnell
Quarz	1	SMD 5 x 3,2 mm	1611815	Farnell
Buck - Converter	2	MSOP 3 x 4,9 mm	1778581	Farnell
Mikrocontroller	1	DIL28-3	1841613	Farnell
IC Sockel	1	Raster 7,62 x 2,54 mm	1103850	Farnell
Phototransistor	1	T1 d=3,4 mm	1497675	Farnell
Bewegungsmelder	1	PIR d=11 mm	1373711	Farnell
NTC - Widerstand	1	3,5 x 4 mm	1187025	Farnell
Anschlussklemmen	4	Raster 5 mm	1177875	Farnell
Anschlussklemmen	6	Raster 3,5 mm	1098611	Farnell
Stiftleiste	1	Raster 2,54 x 2,54 mm	3418492	Farnell
Netzteil	1	99 x 97 x 36 mm	1657440	Farnell
Netzteil	1	27 x 32 x 22 mm	1825773	Farnell
Potentiometer	1	d=6,4 mm	PT 6-L 10K	Reichelt
Relais	2	28 x 5 x 15 mm	FTR-LYCA, 5 V	Reichelt
Sicherungshalter	1	Raster 22,5 mm	PL OGN-22,5	Reichelt
Feinsicherung	1	5 x 20 mm	FLINK 3,15 A	Reichelt
Dip - Schalter	1	Raster 2,54 x 7,62 mm	NT 04	Reichelt
Platine	1	160 x 100 mm	EP2CU 160x100	Reichelt

A3.3 Komplette Schaltung für endgültige Platine



TITLE: led_1_2_neu	REV:
Document Number:	
Date: 28.06.2011 18:52:59	Sheet: 1/1

A4 Quellcode zur Ansteuerung der LED- Treiber

```
#include <avr/io.h>           // laden der notwendigen Funktionen
#include <util/delay.h>
#include <stdint.h>
#include <avr/interrupt.h>
#define F_CPU 20000000UL     // CPU- Frequenz einstellen

void ADC_Init(void);        // ADC starten
void einschalten(void);
void ausschalten(void);
void hochdimmen(void);
void runterdimmen(void);
uint16_t ADC_Read( uint8_t channel );
uint16_t ADC_Read_Avg( uint8_t channel, uint8_t average );

uint8_t sw1 = 0;           // initialisierung globaler Variablen
uint8_t sw2 = 0;
uint8_t sw3 = 0;
uint8_t sw4 = 0;
uint8_t i = 0;
uint8_t a = 0;
uint8_t b = 0;
uint8_t x = 0;
uint8_t Wartefaktor = 0;
uint16_t zeit = 0;

volatile unsigned int tick;
volatile uint16_t s;
void long_delay(uint16_t ms) // Warteschleifenfunktion
{
    for(; ms>0; ms--) _delay_ms(1);
}
int main(void)
```

```

{
// alle Bits von Port B, C und D auf 0 setzen
    DDRB = 0x00; PORTB = 0x00;
    DDRC = 0x00; PORTC = 0x00;
    DDRD = 0x00; PORTD = 0x00;

    DDRB = (1<<PB0) | (1<<PB1); // Ausgang Kontroll-LED & Enable LED-Treiber2
    DDRD = (1<<PD4) | (1<<PD6); // Ausgang Relais & Enable LED-Treiber1
    // Pullup für Schaltereingänge an
    PORTD = (1<<PD0) | (1<<PD1) | (1<<PD2) | (1<<PD3);

// Timer 0 einstellen

TCCR0A |= (1<<COM0A1) | (1<<WGM00); // Timer/Counter 1 für PWM einstellen
TCCR0B |= (1<<CS00) | (1 << CS01); // Prescaler festlegen

TCCR1A |= (1<<COM1A1) | (1<<WGM10); // Timer/Counter 2 für PWM einstellen
TCCR1B |= (1<<CS10) | (1 << CS11); // Prescaler festlegen

// Timer/Counter 2 konfigurieren

TCCR2A = (1<<WGM21); // CTC Modus
TCCR2B |= (1<<CS20) | (1<<CS21) | (1<<CS22); // Prescaler 1024
//OCR2A = 20; //50ns * presc 1024 entspricht mit Multiplikator 19,53 (~20) einer ms
OCR2A = 255; // entspricht 13,1 ms
// Compare Interrupt erlauben
TIMSK2 |= (1<<OCIE2A);
// globale Interrupts aktivieren
//PORTB |= (1<<PB0);
sei();

/* der Compare Wert:
Wenn der Zähler diesen Wert erreicht, wird mit obiger Konfiguration der OC1A Ausgang
abgeschaltet.Sobald der Zähler wieder bei 0 startet, wird der Ausgang wieder auf 1 gesetzt.

```

Durch Verändern dieses Wertes, werden die unterschiedlichen PWM Werte eingestellt. */

```
//Test bei Programmstart mittels grüner LED
```

```
DDRB = 0xff;  
PORTB |= (1<<PB0);  
long_delay(500);  
PORTB &= ~(1<<PB0);
```

```
uint16_t adcval_1 = 0;  
//uint16_t adcval_2 = 0;  
//uint16_t adcval_3 = 0;  
ADC_Init();
```

```
while (1)
```

```
{
```

```
    switch (Wartefaktor) // eingestellte Zeitdauer für hochdimmen
```

```
    {
```

```
        case 1: zeit = 300;           //5 min
```

```
            break;
```

```
        case 2: zeit = 120;           //2 min
```

```
            break;
```

```
        case 3: zeit = 60;            //1 min
```

```
            break;
```

```
        case 4: zeit = 30;            //30 s
```

```
            break;
```

```
        case 5: zeit = 15;            //15 s
```

```
            break;
```

```
        case 6: zeit = 5;             //5 s
```

```
            break;
```

```
        case 7: zeit = 2;             //2 s
```

```
            break;
```

```
        default: zeit = 1800;         //30 min (alle Schalter on)
```

```
    }
```

```

// mach was mit adcval
adcval_1 = ADC_Read_Avg(1, 10); // Kanal 1 (Temperatur), Mittelwert aus 10 Messungen
//adcval_2 = ADC_Read_Avg(2, 10); // Kanal 2 (Helligkeit), Mittelwert aus 10 Messungen
//adcval_3 = ADC_Read_Avg(3, 5); // Kanal 3 (Poti), Mittelwert aus 5 Messungen

while (adcval_1 < 372) //maximale Temperatur überschritten
{
    PORTD &= ~(1<<PD4); //Notausschaltung bei Überhitzung
    OCR0A = 0x00;
    OCR1A = 0x00;
}

//Dämmerung erreicht nach eingestelltem Wert
if ( (ADC_Read_Avg(2, 10) < ADC_Read_Avg(3, 5)) & (x<1) )
{
    long_delay(3000); // 3 Sekunden warten
    if ( ADC_Read_Avg(2, 10) < ADC_Read_Avg(3, 5) ) // erneuter Vergleich
    {
        einschalten();
        x = 1;
    }
}

// Schalterzustände der Schiebeschalter abfragen
sw4 = (PIND & (1 << PD3));

//Bewegungsmelder Signal 1, oder Schalter sw4 ein(Dauerlicht)
if ( ((PINC) & (1 << PINC5)) )
{
    hochdimmen();
    a = 1;
    tick = 0;
    s = 0;
}

```

```

if ( (!sw4) & (i>0) )
{
PORTB |= (1<<PB0);
hochdimmen();
b = 1;
}

```

//keine Bewegung mehr und nicht auf Dauerlicht geschaltet

```

if ( ((OCR0A > 32) &! (PINC & (1 << PINC5))) & (a == 1) )
{
a = 0;
}
if ( (s == (zeit)) & (a == 0) & (b == 0) )
{
runterdimmen();
}
if ( (b == 1) &! (!sw4) )
{
PORTB &= ~(1<<PB0);
runterdimmen();
b = 0;
}

```

if (ADC_Read_Avg(2, 10) > (ADC_Read_Avg(3, 5) +32)) //Tageslicht erreicht

```

{
long_delay(3000);
if ( ADC_Read_Avg(2, 10) > (ADC_Read_Avg(3, 5) +32) )
{
ausschalten();
x = 0;
}
}
}

```

```

}
// ADC initialisieren
void ADC_Init(void) {

uint16_t result;

ADMUX = (0<<REFS1) | (1<<REFS0);           //AVcc als Referenzspannung benutzen
//ADMUX = (1<<REFS1) | (1<<REFS0);         // interne Referenzspannung nutzen
ADCSRA = (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); // Frequenzvorteiler
ADCSRA |= (1<<ADEN);                       // ADC aktivieren

/* nach Aktivieren des ADC wird ein "Dummy-Readout" empfohlen, d.h.man nimmt
   einen Wert auf und verwirft diesen wieder, um den ADC "warmlaufen zu lassen" */

ADCSRA |= (1<<ADSC);                       // eine ADC-Wandlung
while (ADCSRA & (1<<ADSC) ) {}             // auf Abschluss der Konvertierung warten
/* ADCW muss einmal gelesen werden, sonst wird Ergebnis der nächsten
   Wandlung nicht übernommen. */
result = ADCW;
}

// ADC Einzelmessung
uint16_t ADC_Read( uint8_t channel )
{
// Kanal wählen, ohne andere Bits zu beeinflussen
ADMUX = (ADMUX & ~(0x1F)) | (channel & 0x1F);
ADCSRA |= (1<<ADSC);                       // eine Wandlung "single conversion"
while (ADCSRA & (1<<ADSC) ) {}             // auf Abschluss der Konvertierung warten
return ADCW;                                // ADC auslesen und zurückgeben
}

// ADC Mehrfachmessung mit Mittelwertbildung
uint16_t ADC_Read_Avg( uint8_t channel, uint8_t average )
{

```

```

uint32_t result = 0;
for (uint8_t k = 0; k < average; ++k )
    result += ADC_Read( channel );

return (uint16_t)( result / average );
}
// Funktion zum Hochdimmen
void hochdimmen (void)
{
    while (i<255)
    {
        i++; //hochdimmen auf 100%
        OCR0A = i;
        OCR1A = i;
        _delay_ms(10);
    }
}
// Funktion zum Herunterdimmen
void runterdimmen (void)
{
    while ( i>32 )
    {
        i--; //herunterdimmen auf 12,5%
        OCR0A = i;
        OCR1A = i;
        _delay_ms(20);
    }
}
// Funktion zum Einschalten
void einschalten (void)
{
    PORTD |= (1<<PD4); //Netzteil an bei Dämmerung
    _delay_ms(100); //Verzoegerung für Relais
    while (i<32)
    {

```

```

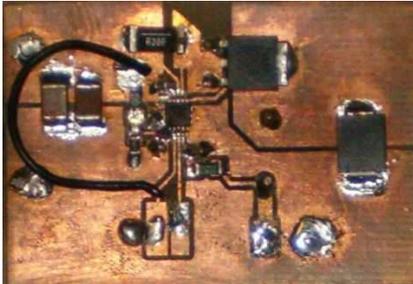
        i++;           //hochdimmen auf Dauerleuchtstärke
        OCR0A = i;
        OCR1A = i;
        _delay_ms(30);
    }
}
// Funktion zum Ausschalten
void ausschalten (void)
{
    while ( i>0 )
    {
        i--;           //herunterdimmen auf 0
        OCR0A = i;
        OCR1A = i;
        _delay_ms(20);
    }
    PORTD &= ~(1<<PD4); //Licht aus
}

// ISR zum Erzeugen von verschiedenen Zeiten
ISR (TIMER2_COMPA_vect)
{
    tick++;
    if(tick == 76) // 76*13,1 = 1s
    {
        Wartefaktor = (PIND & 0x07); //Schalterkombination auslesen und übergeben
        s++;
        tick = 0;
        if(s == 60000) // nach 60000 Sekunden Zähler zurücksetzen
        {
            s = 0;
        }
    }
}
}

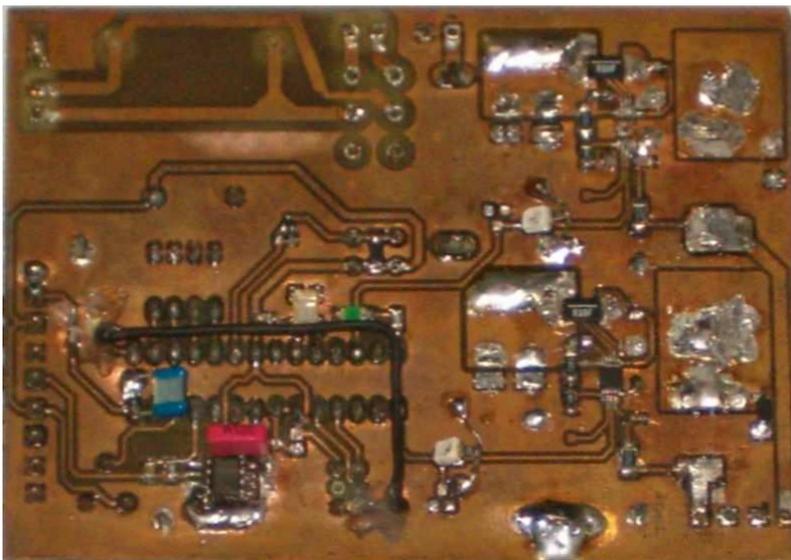
```

A5 Gefertigte Platinen

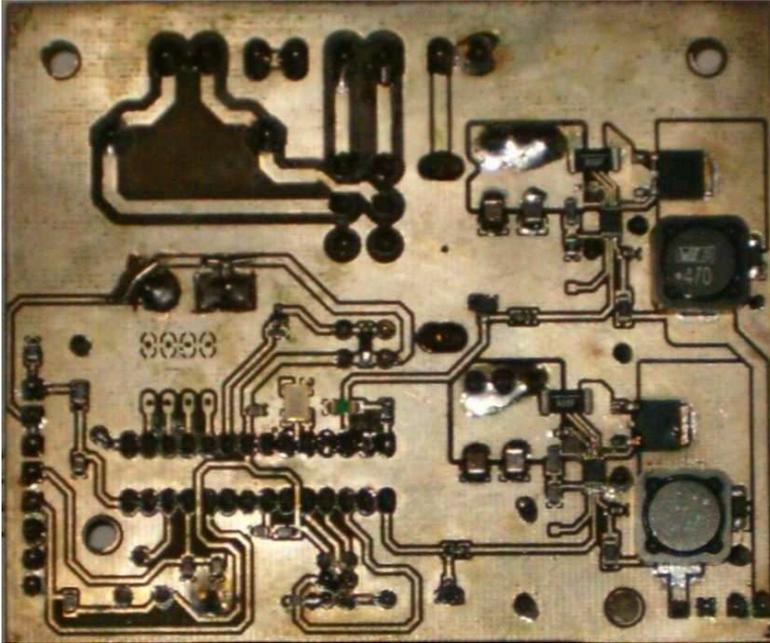
A5.1 Testplatine



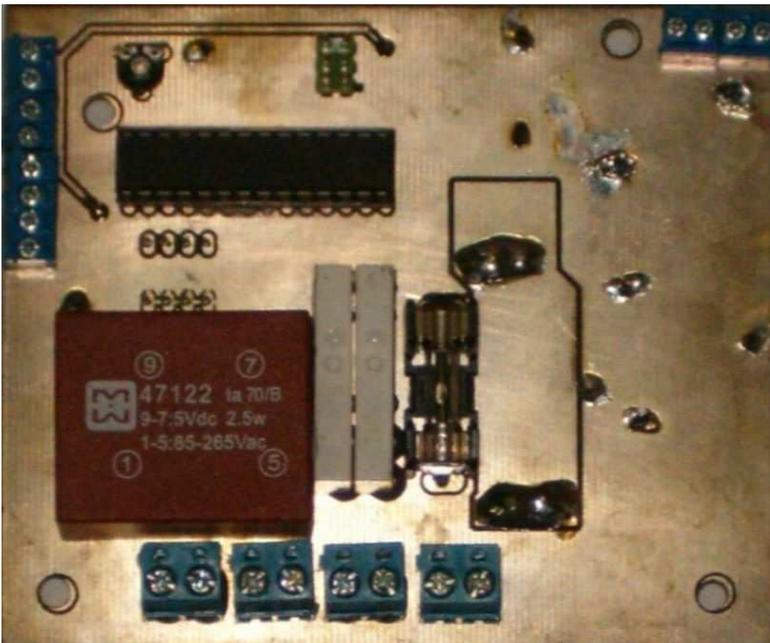
A5.2 Hauptplatine mit OPV



A5.3 Hauptplatine entgültig



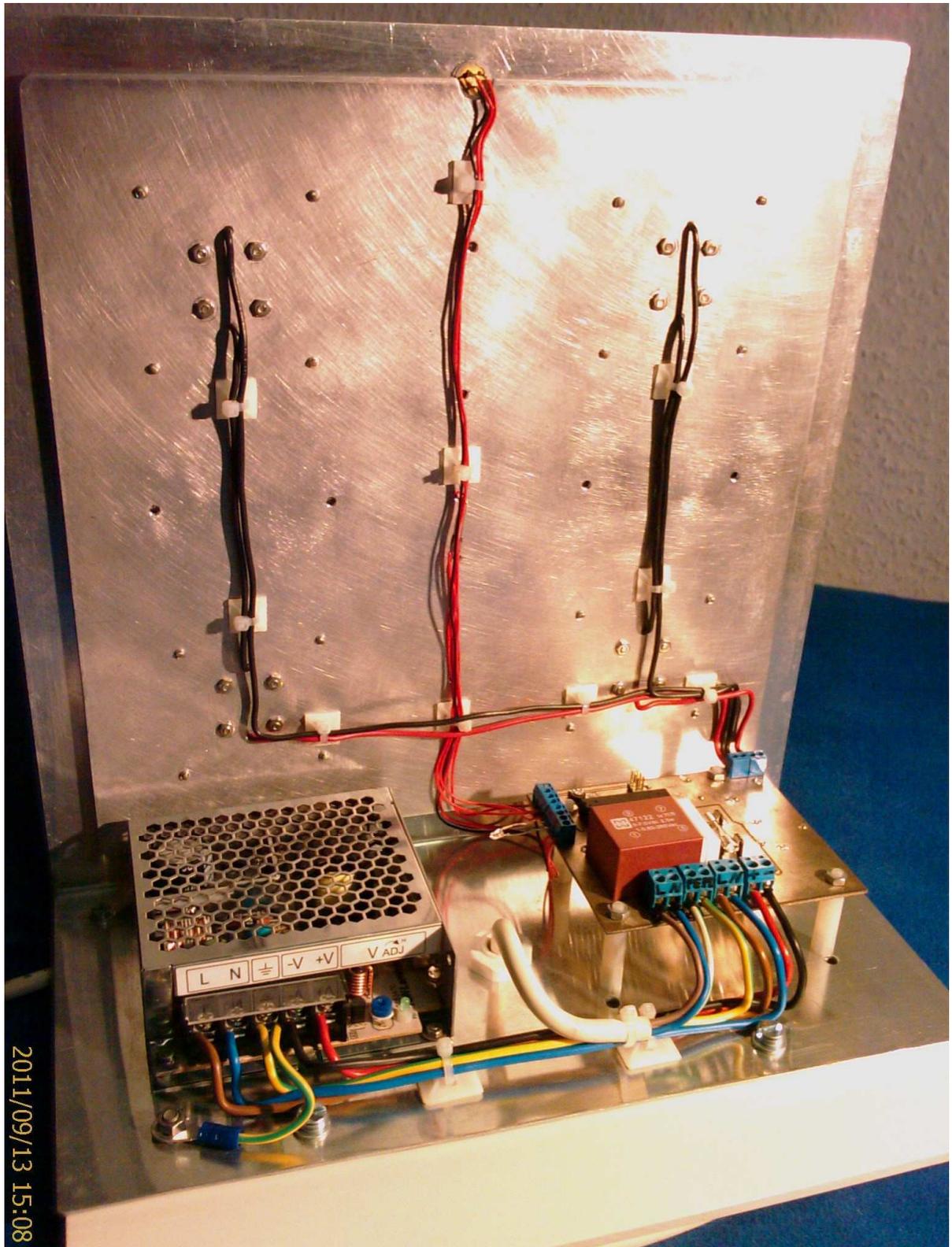
Rückseite



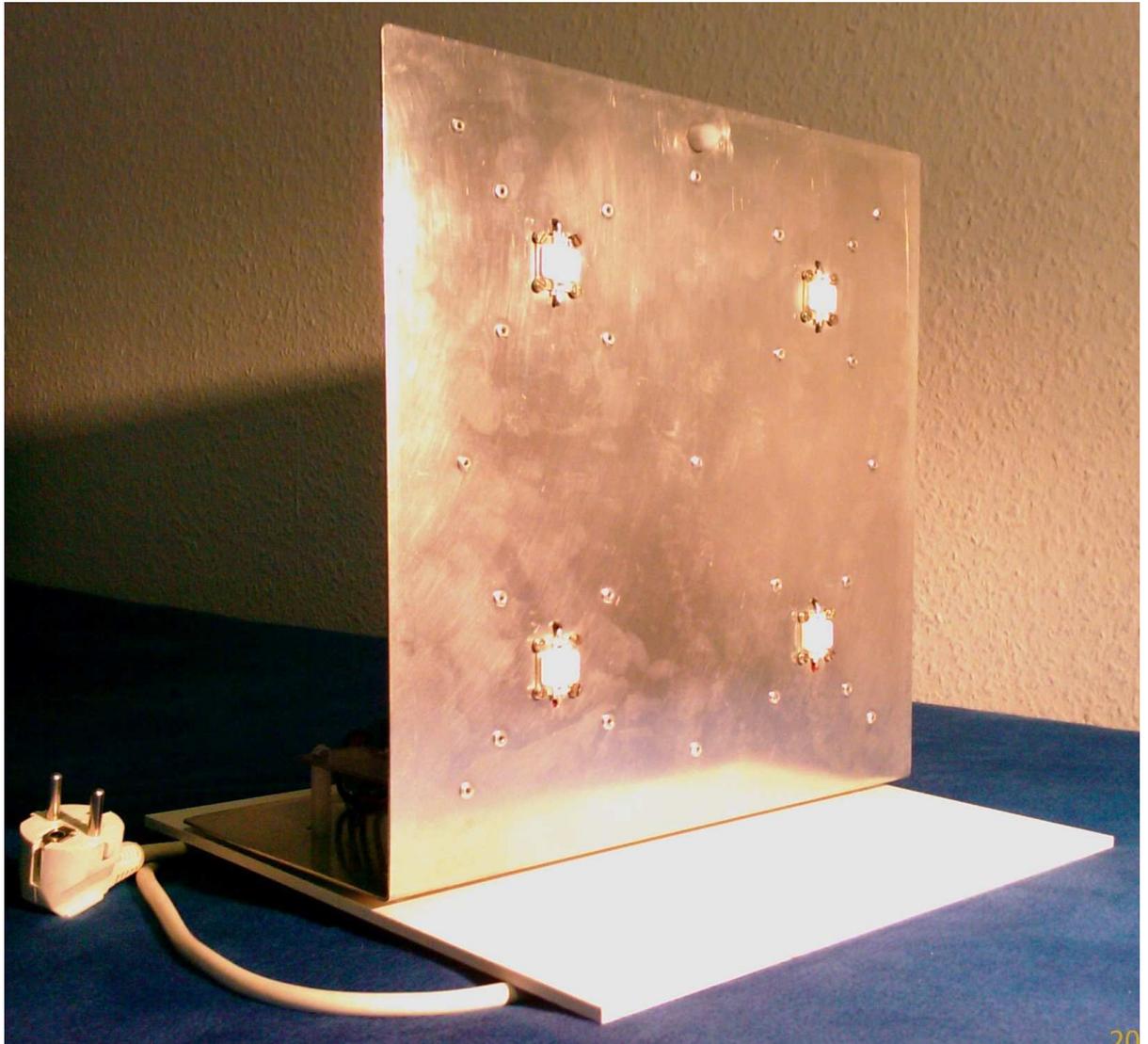
Vorderseite

A6 Bilder

A6.1 Leuchte mit Ansterelektronik und Netzteil von innen



A6.2 Unterseite der Leuchte mit LEDs



A6.2 Komplette Leuchte mit möglichem Gehäuse

