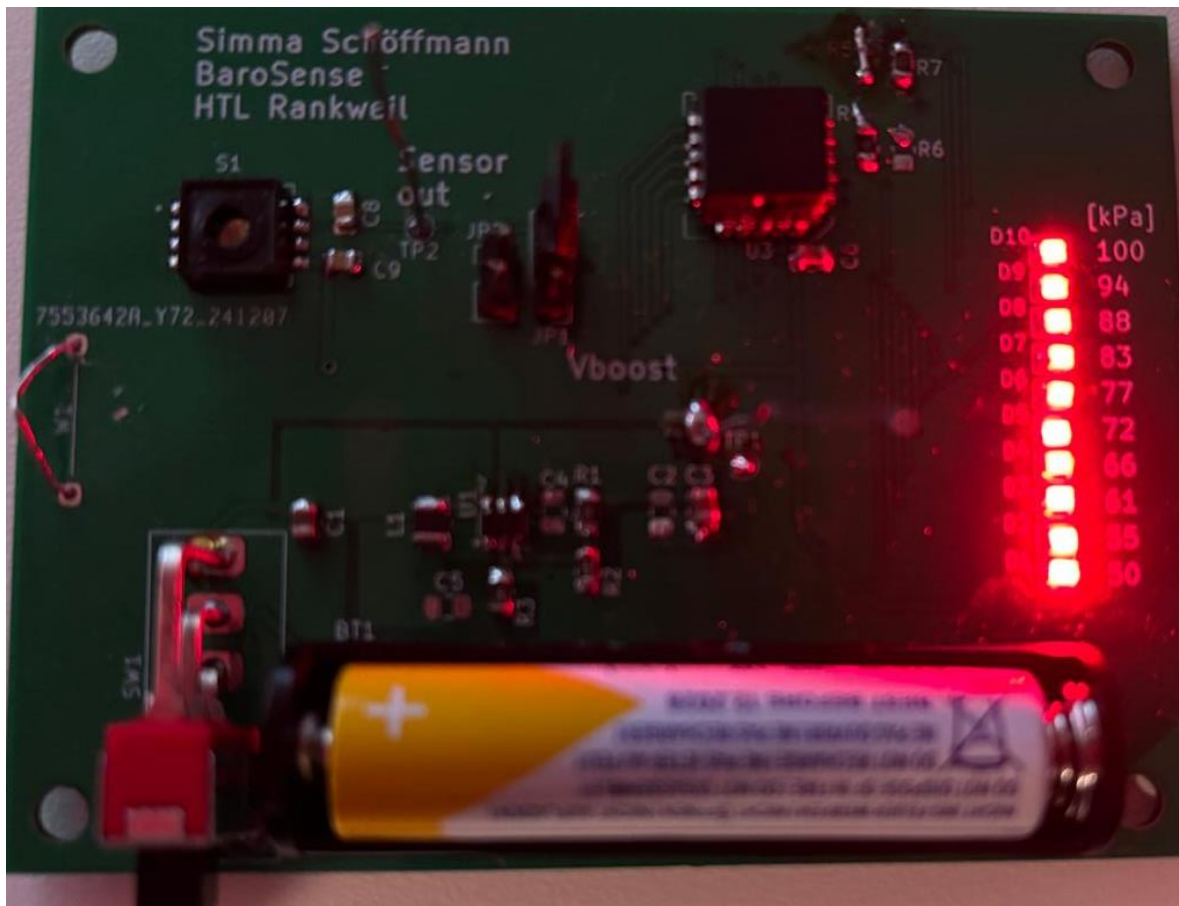


Projekt A: BaroSense



Team: Simma Kilian, Schöffmann Raphael

Betreuer: Prof. Zudrell-Koch

Jahrgang: 4BHEL

Schuljahr: 2024/25

Projekt Nummer: 24324

GitHub: <https://github.com/DarkPhoenix889/BaroSense>

Inhaltsverzeichnis

| | |
|-------------------------------------|----|
| Aufgabenstellung..... | 3 |
| Projekt-Ziele | 3 |
| Schaltung..... | 3 |
| Blockschaltbild | 3 |
| Boost-Converter MP3210..... | 4 |
| Infineon-KP234 Luftdrucksensor..... | 5 |
| LM3914 LED Treiber..... | 6 |
| Berechnungen..... | 7 |
| Simulation | 8 |
| KiCad | 9 |
| Schematic | 9 |
| Layout..... | 12 |
| 3D-View | 12 |
| Gehäuse..... | 13 |
| Inbetriebnahme | 14 |
| Softwareteil Megacard | 17 |
| Korrekturen | 18 |
| Ergebnisse..... | 19 |
| Ideen für zukünftige Projekte | 20 |
| Leistungsbeitrag Raphael..... | 20 |
| Leistungsbeitrag Kilian | 20 |

Aufgabenstellung

Es war von Beginn an klar, dass ein Sensorsignal auf LEDs ausgegeben werden muss. Mithilfe eines ICs, soll dann auf 10 LEDs das Sensor Signal ausgegeben werden.

Die Versorgung wird über eine 1.5V AAA-Batterie hergestellt, bei welchem wir die Spannung auf 5V konvertieren müssen.

Den Schaltplan sowie die Dimensionierung aller Bauteile und designen des Layouts müssen durchgeführt und in einem technischen Bericht dokumentiert werden.

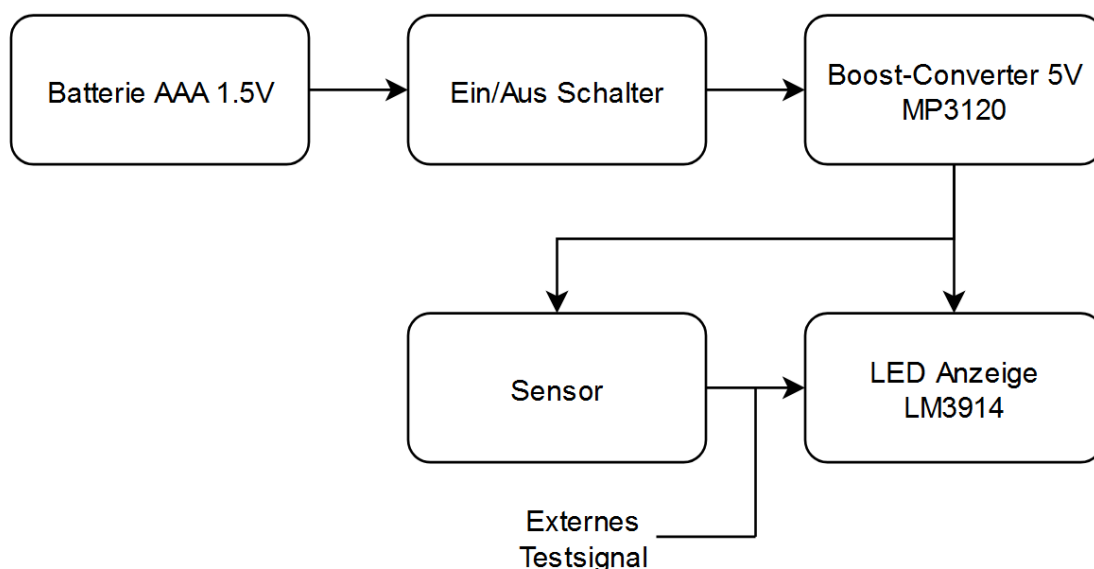
Die fertige Leiterplatte muss dann aufgebaut, getestet, gemessen und in Betrieb genommen werden.

Projekt-Ziele

Wir haben uns einen Luftdrucksensor in Angriff genommen. Dieser wandelt den gemessenen Luftdruck dann in eine Spannung um. Mit dem IC LN3914 soll die Spannung dann an LEDs ausgegeben werden. Bei dem IC lässt sich ein Messbereich mit einer Referenzspannung und einem Spannungsteiler einstellen. Wir haben zuerst einen Bereich von 50kP bis 100kP gewählt, welcher relativ stupide war, da wir in keiner Welt 50kP erreichen können. Kurz vor Ende haben wir die Widerstände nochmal umgeändert, sodass unser Messbereich zwischen 90kP und 100kP liegt, was nun ca. der Höhenunterschied von Rankweil und Sulzberg (Wohnort von Raphael) darstellt. Einen größeren Messbereich über 100kP ist quasi unmöglich gewesen, da wir in der Simulation nicht über 4,3V Ref-High erreichen konnten obwohl das benötigt worden wäre. Mithilfe des MP3210 Boost-Converters gelingt es die Spannung von 1,5V auf 5V hoch zu boosten. Dazu muss nur denen im Datenblatt beschriebenen Details befolgt werden.

Schaltung

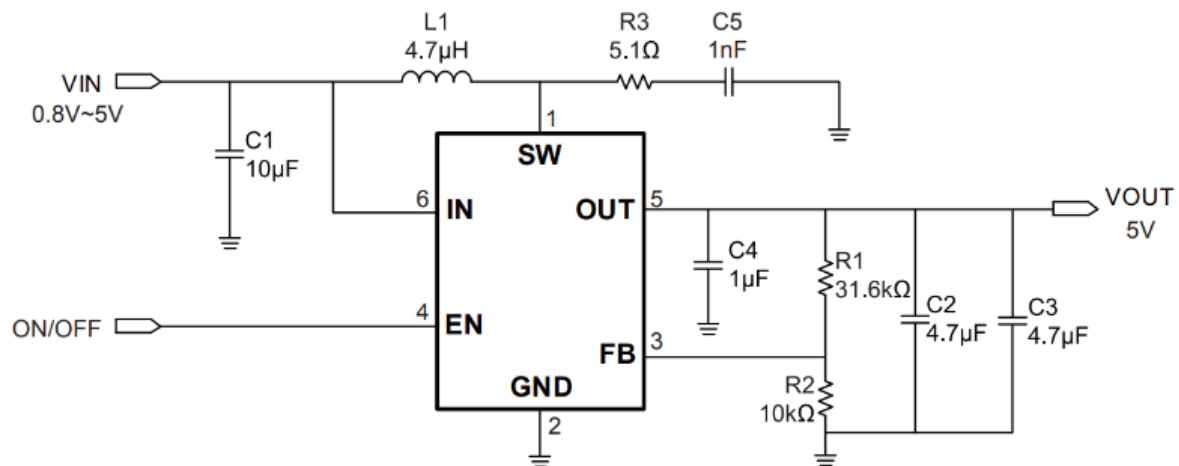
Blockschaltbild



Die Schaltung wird mit einer 1.5V AAA-Batterie versorgt. Über einen Schalter kann die Schaltung An und Aus geschaltet werden.

Der Boost-Converter zieht die Spannung von 1.5V auf eine mehr praktische Spannung von ca. 5V. Diese Spannung betreibt einen Sensor, in unserem Fall einen Luftdrucksensor. Das Signal des Sensors wird in den LED-Treiber eingespeist, welcher die LEDs dementsprechend leuchten lässt.

Boost-Converter MP3210



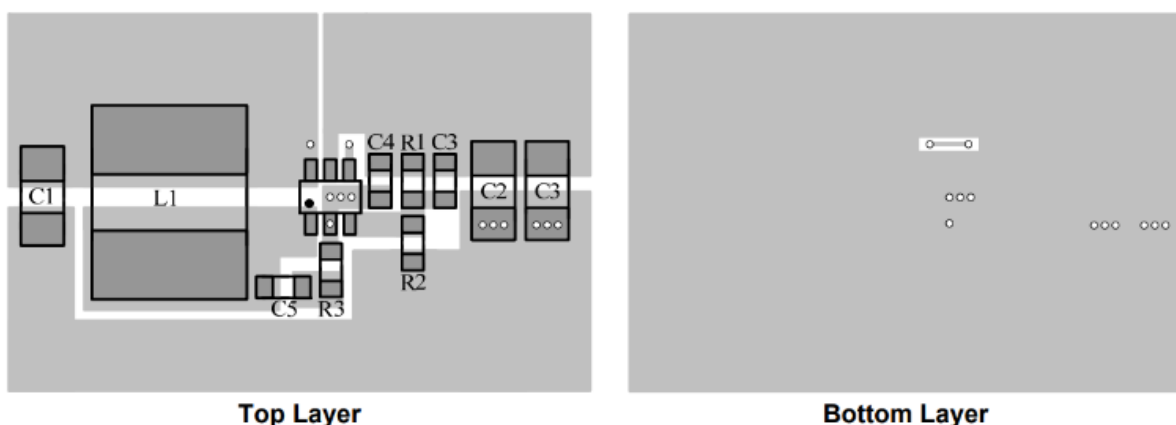
Da die 1.5V der Batterie nicht genug sind, um unsere Schaltung zu betreiben, verwenden wir dieses Bauteil, um daraus eine Spannung von 5V zu erzeugen.

Pin 4 ist direkt mit der Versorgungsspannung (VCC) verbunden, wodurch der Regler aktiviert wird. Die Spule L1 speichert während des Schaltvorgangs Strom. Wenn der im IC integrierte MOSFET eingeschaltet ist, steigt der Strom durch die Induktivität. Beim Abschalten des MOSFETs wird die gespeicherte Energie über die Diode an die Ausgangsspannung abgegeben.

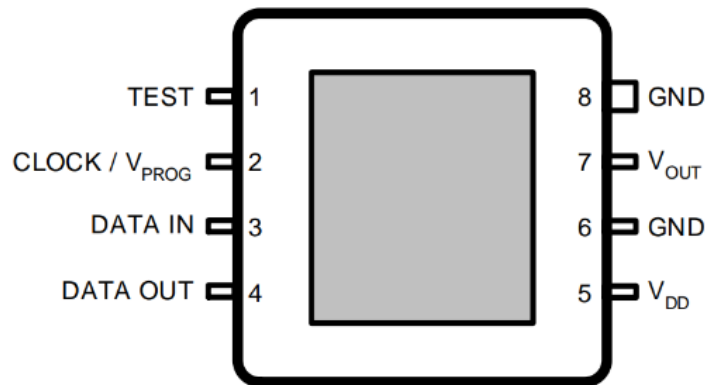
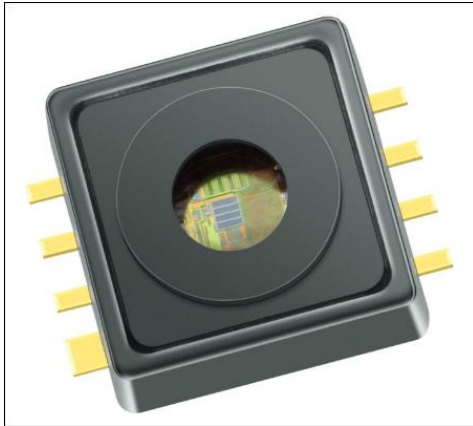
Spannungsspitzen, die durch den schnellen Schaltbetrieb von ca. einem 1 MHz entstehen, werden durch einen Widerstand R3 und dem Kondensator C5 gedämpft. Zur Glättung der Ausgangsspannung dienen C2, C3 und C4. Dabei ist C4 möglichst nahe am Ausgang des Boosters platziert. Die Kondensatoren C2 und C3 wurden in bei uns mit einem 10µF Kondensator ersetzt.

Die Regelung der Ausgangsspannung auf 5 V erfolgt durch einen Spannungsteiler, bestehend aus den Widerständen R1 und R2. R1 wurde aufgrund der E24 Reihe auf einen 30k Ohm Widerstand abgeändert.

Hier haben wir noch das ideale Layout laut Datenblatt. Die Vias im PCB dienen zur Temperaturkontrolle



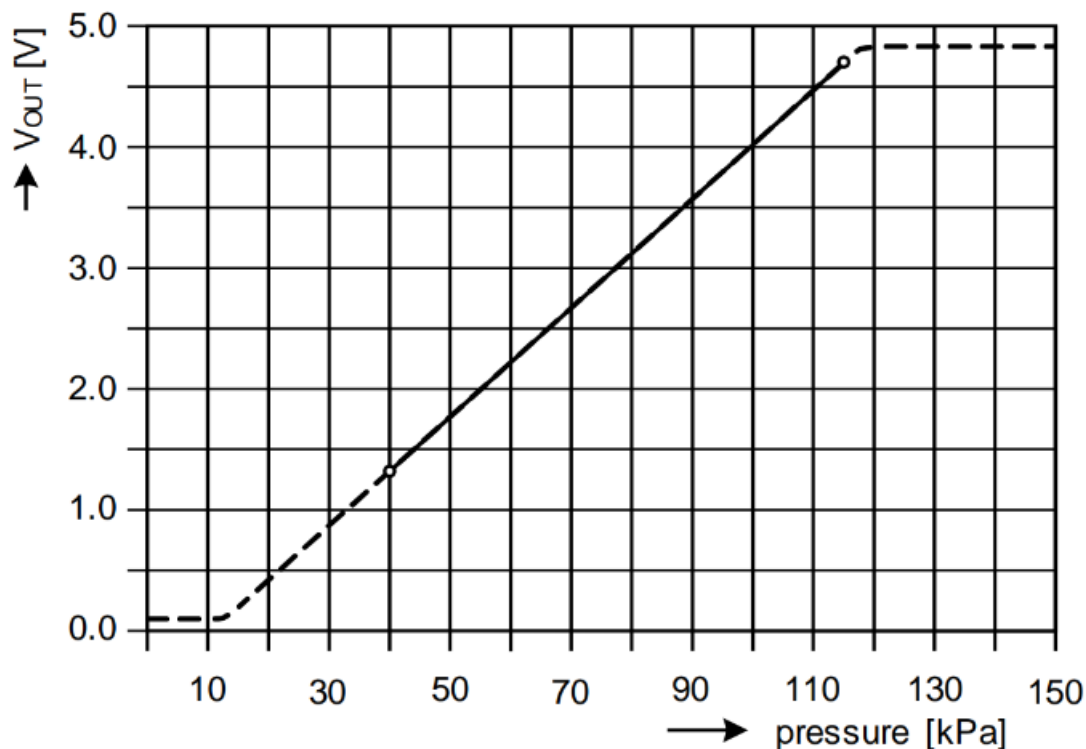
Infinion-KP234 Luftdrucksensor



Für unser Projekt wurde dieser Sensor verwendet.

Die Pins auf der linken Seite, also Pin 1 bis 4, werden nur zur Kalibrierung benötigt, weshalb sie, wie es im Datenblatt steht, floaten. Pin 6 und 8 werden mit Ground verbunden. An V_{DD} wird zusätzlich ein 100nF auf GND dazuschalten. Dasselbe gilt für das V_{out}.

V_{out} ändert sich je nach Luftdruck entsprechend diesem Graph:



Das bedeutet das wir auf Seehöhe ziemlich genau 4V erwarten können. In Rankweil liegt der erwartete Wert bei ca. 3.75V. Auf Sulzberg hätten wir als Referenz eine Spannung von etwa 3.5V,

V_{out} ist aber stark abhängig von der Versorgungsspannung, weshalb diese Wert nur stimmen wenn diese genau 5V beträgt.

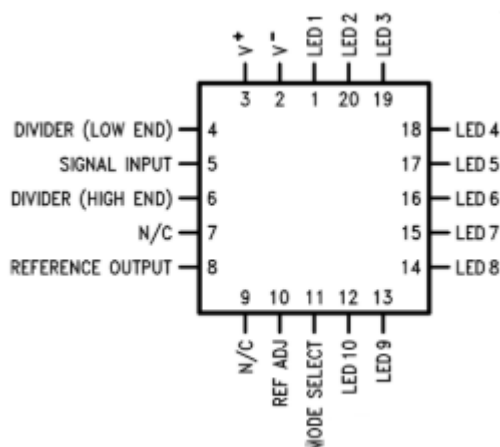
LM3914 LED Treiber

Der LM3914 ist ein IC, welcher unser Analoges Sensorsignal auf 10 LEDs ausgibt. Die Ausgabe kann entweder im Punkt- oder Balken-Modus geschehen, welcher durch das ansetzen von V_{CC} an den Pin 11 eingestellt werden kann. Der Bereich, in welchem der Treiber arbeitet, wird über 2 Referenzspannungen, welche an die Pins 4 (Low End) und Pin 6 (High End) gelegt wird. Die Referenzspannungen erzeugen wir über eine Serie von Spannungsteilern und mithilfe des Pin 11, welcher einen Spannungsabfall von 1.25V regelt. Die Leds werden mit den dementsprechenden Pins verbunden.

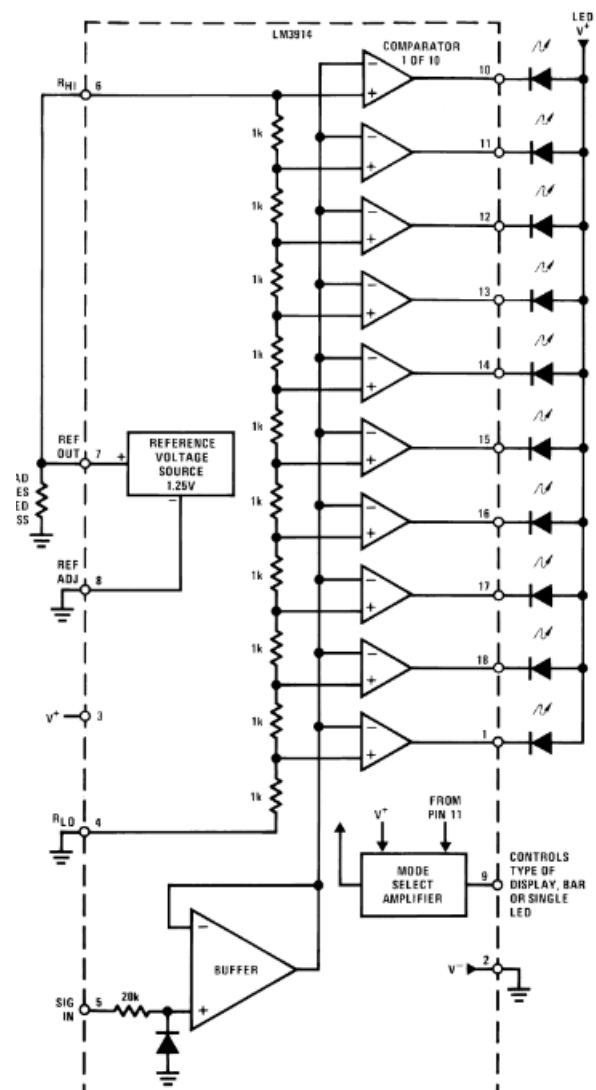
Die Versorgungsspannung wird an den Pin 3 (V^+) und GND wird an den Pin 2 (V^-) angeschlossen.

Des tolle am LED-Treiber ist das wir uns nicht über die Dioden Spannung kümmern müssen, da der IC alles regelt.

Das Sensorsignal wird mit dem Pin 5 (Signal Input verbunden)



Die Umschaltung zum Balken-Modus erfolgt durch einen Komparator, welcher auf einem Eingang des Komparators $V_{CC} - 10\%$ anliegt.



Berechnungen

Die Ursprünglichen Berechnungen für die Referenzspannung sind leider verloren gegangen. Die für die neue Version wurde im Kopf gerechnet und durch Simulationen geprüft.

Diese Berechnung wurde gemacht, um im Softwareteil aus der gemessenen Spannung ein Luftdruck zu berechnen

Die Grundformel stammt aus dem Sensordatenblatt. Bei a und b handelt es sich um im Datenblatt definierte Variablen, welche unter gleichem Namen auch im Softwareteil verwendet werden.

$$V_{out} = V_{DD} \times (a \times P + b)$$

$$V_{DD} = 5V$$

$$a = 0,009$$

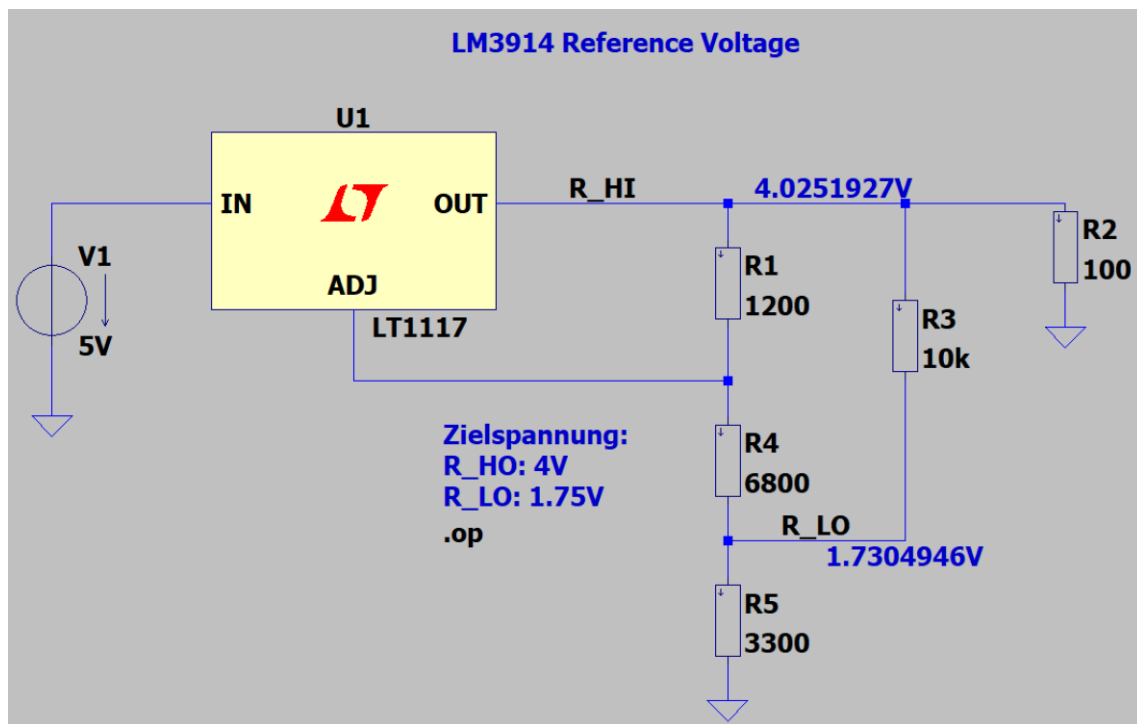
$$b = -0,095$$

$$\frac{V_{out}}{V_{DD}} = a \times P + b \quad | -b$$

$$\frac{V_{out}}{V_{DD}} - b = a \times P \quad | :a$$

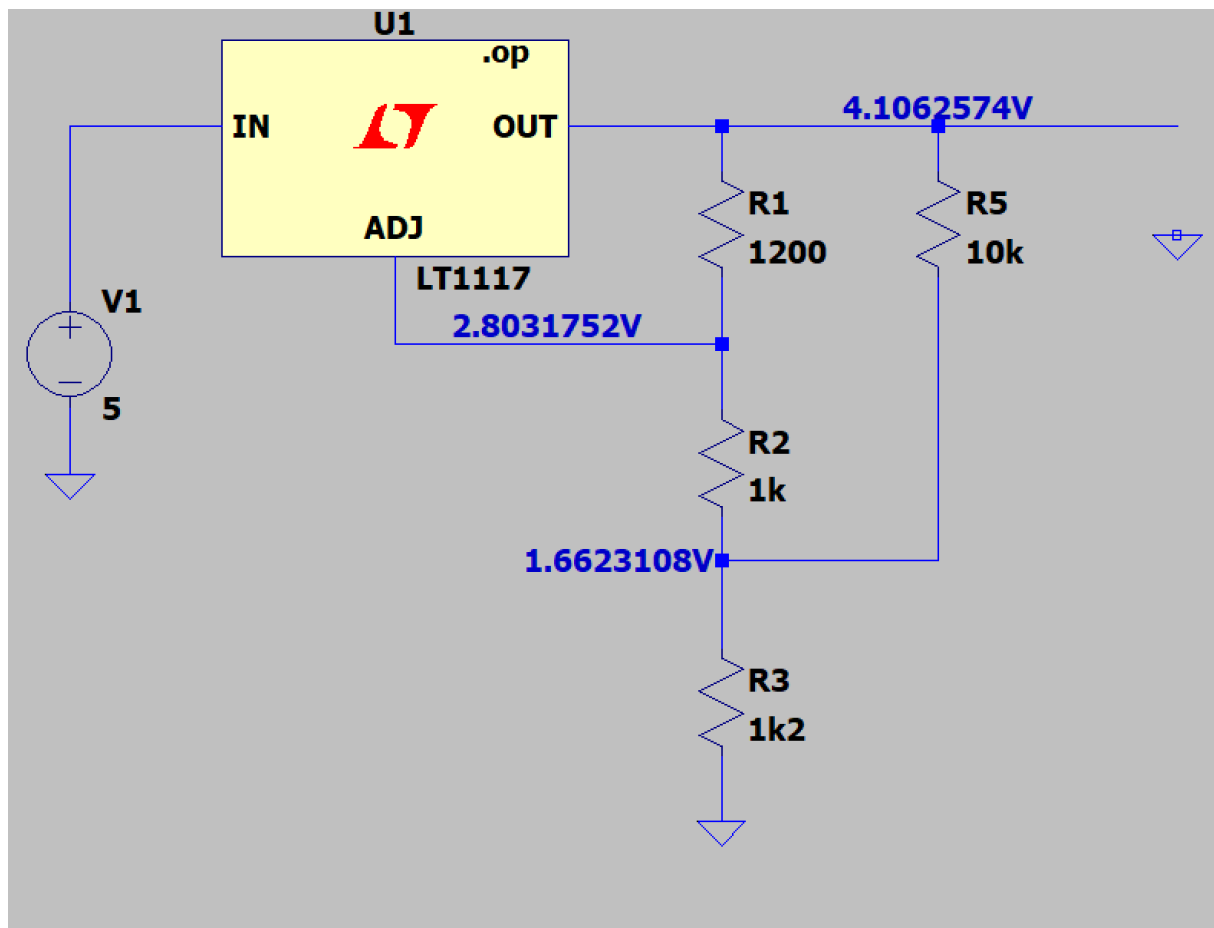
$$P = \frac{V_{out}}{V_{DD} \cdot a} - \frac{b}{a}$$

Simulation

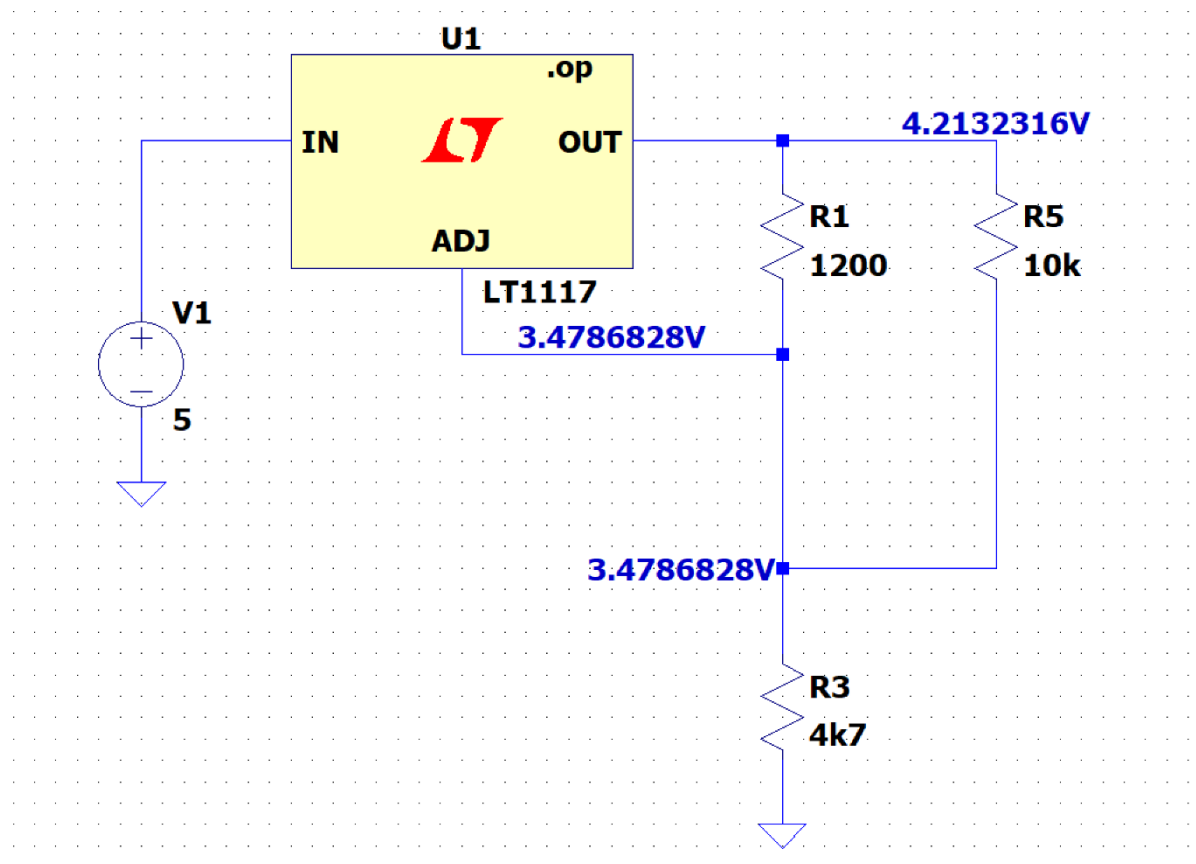


Hier haben wir Simulation für die Referenzspannung beim LM3914. Das Ziel ist es eine Skala von 1.75V bis 4V zu kriegen

Da diese Simulation auf fehlerhaften Brechnungen beruht haben wir hier einen neue richtige:



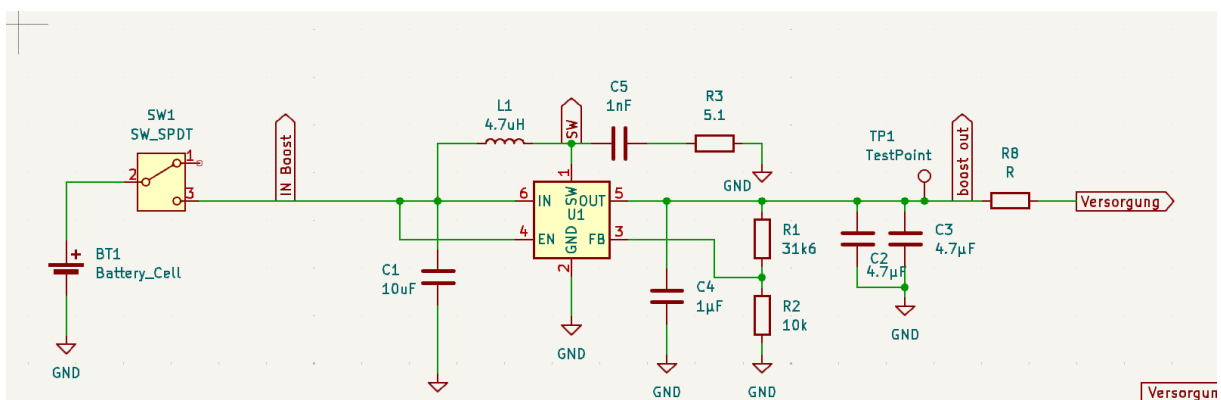
Letzte Anpassungen der Referenzspannung um einen kleineren und brauchbaren Messbereich zu erhalten.



KiCad

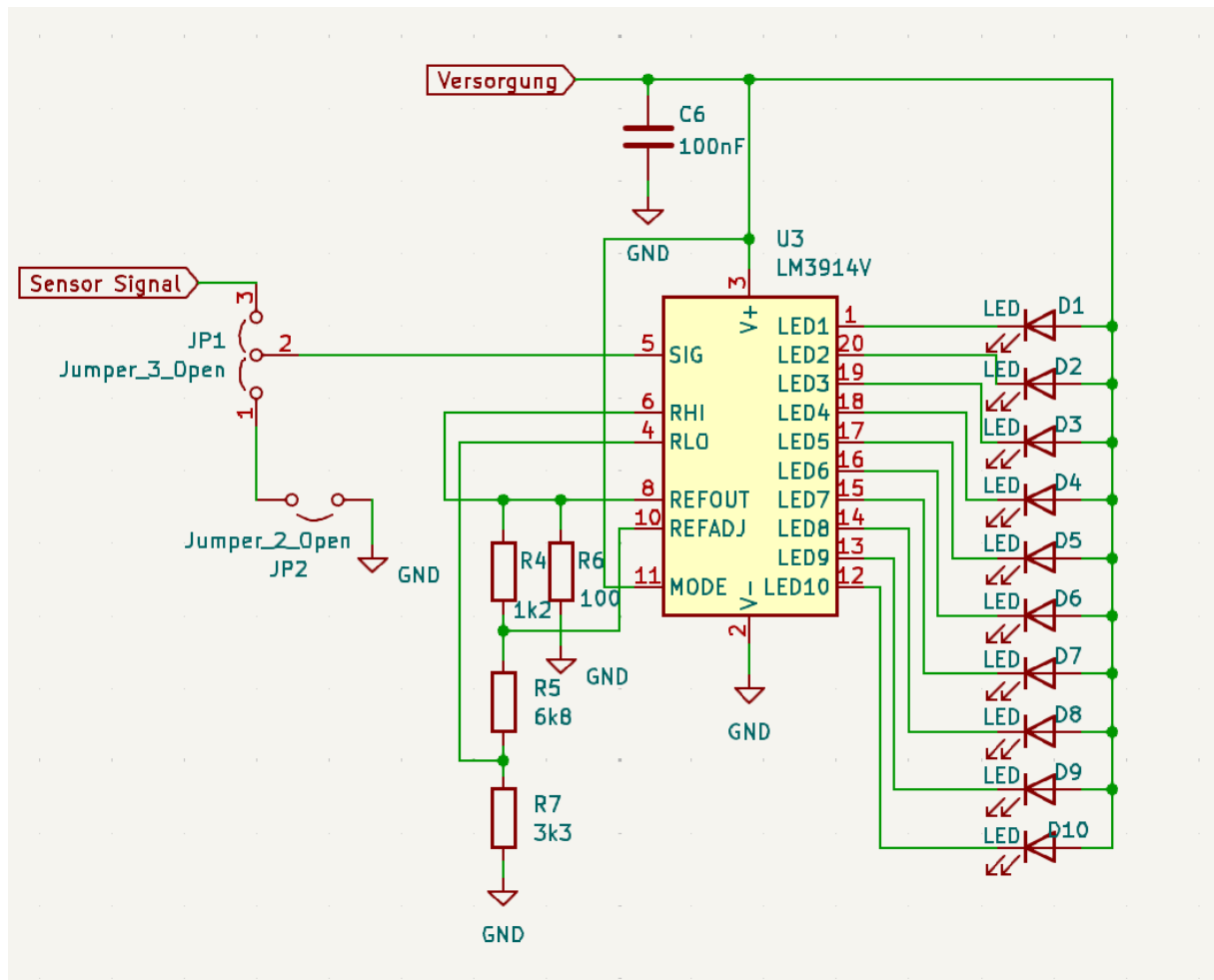
Schematic

Boost Converter



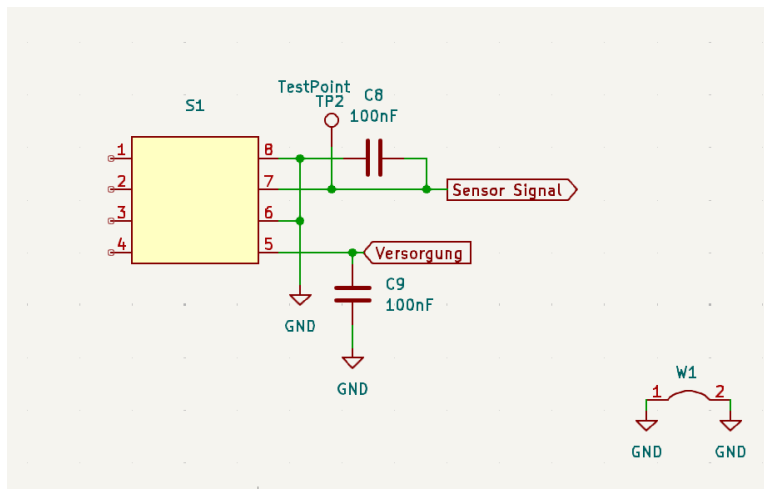
Bei dem Boost Converter gilt es zu beachten, dass die Bauteile nach Datenblatt angereicht und dimensioniert sind, da dies essenziell für den Betrieb von ihm ist. Auf der linken Seite ist noch der Batteriehalter mit dem Schalter zu sehen und rechts haben wir einen Test Point für die Messung.

LED-Treiber



Hier war es wichtig die richtigen Widerstände bzw die richtige Anordnung festzulegen. Hier sind noch die alten Widerstandswerte zu sehen, mit welchen der falsche Messbereich dargestellt wird, im Nachhinein haben wir die Widerstände geändert. Der 100nF Kondensator sollte im Layout möglichst nah am IC platziert werden. Am Signal Eingang sehen wir zwei Jumper, diese sind dazu da um zwischen dem Sensor Signal und einem extern gespeisten Signal umzuschalten um die Funktionalität des ICs nachzuweisen und zu messen. Bei der Bestückung der Leiterplatte wird noch die Orientierung der LEDs wichtig sein, da diese an Vcc angeschlossen sind und die Kathode an dem IC, was uns als eher unüblich vorkam.

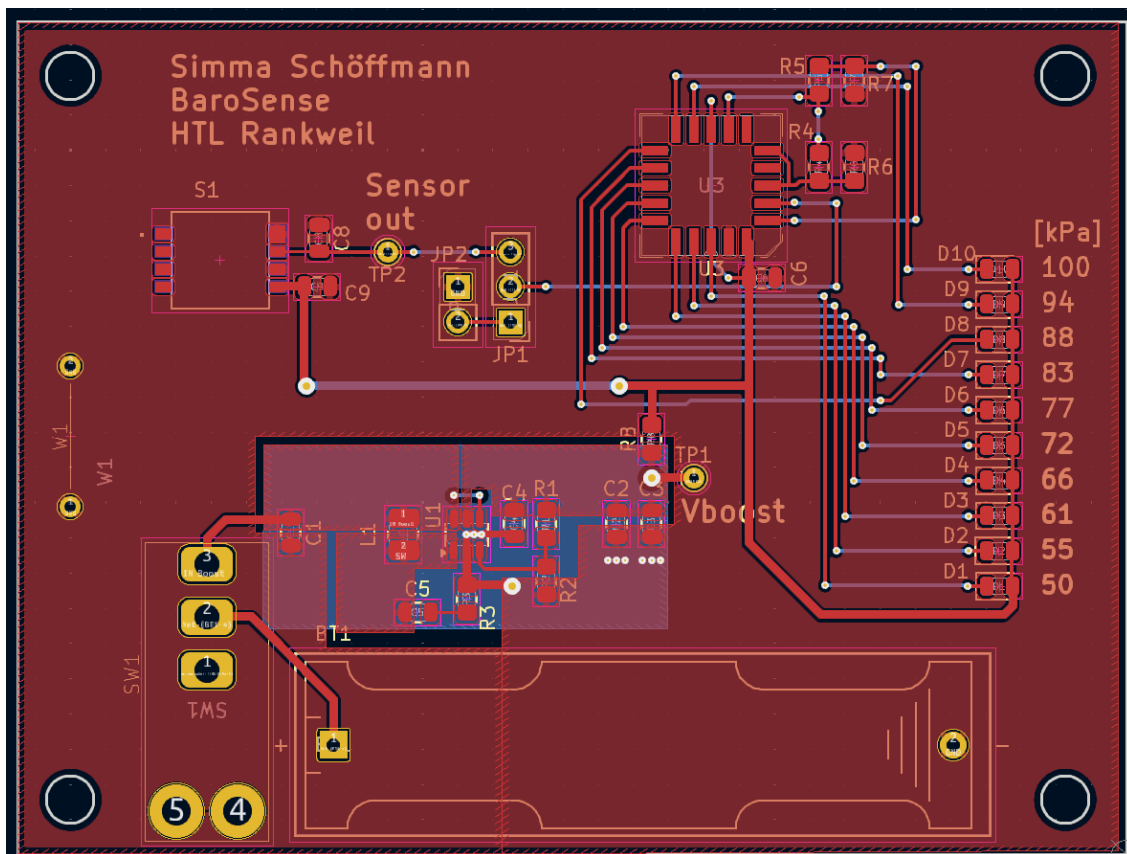
Sensor und GND-Wire



Der Sensor war am simpelsten anzuschließen, da wir nur 4 von den 8 Pins benötigen. Auch hier haben wir wieder einen Testpoint eingebaut und einen 100nF Kondensator.

Das GND-Wire ist für das Messen wichtig, da dieses das Messen um einiges erleichtert.

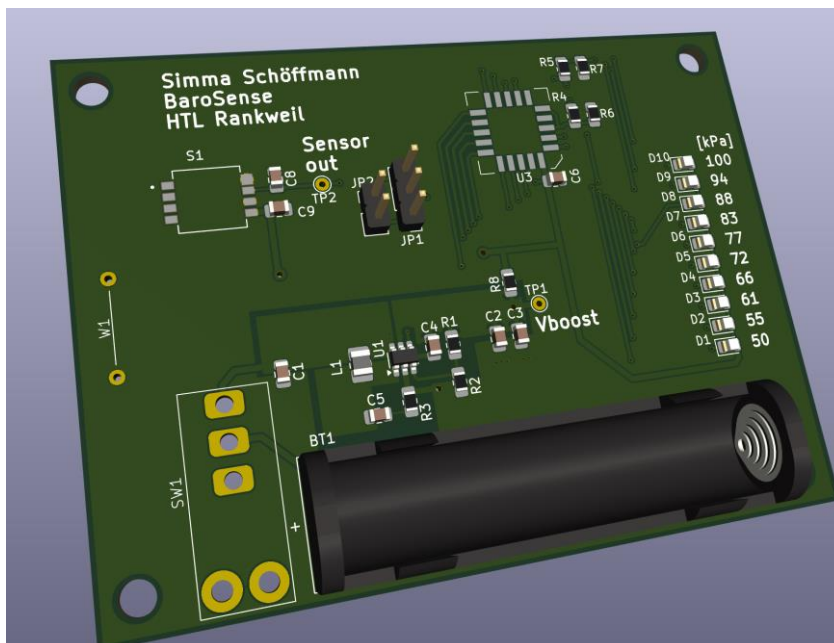
Layout



Es war uns hier wichtig, dass der Schalter für leichten Zugang am Rand platziert ist, und der Batteriehalter hat sich als praktisch erwiesen, dass er direkt daneben ist.

Dass du LEDs, der LED-Treiber sowie der Sensor relativ viel Abstand zum Boost Converter hat folgenden Grund, dass dieser keine Störungen von den anderen Bauteilen hat, da er aufgrund der hohen Frequenzen mit welchen er arbeitet, leicht durch andere ICs beeinflusst werden kann.

3D-View



Fertiges PCB in 3D-Ansicht, wie es produziert und bestückt worden ist.

Wichtig war eine leichte Lesbarkeit der Messwerte, weswegen diese groß neben den LEDs stehen. Allerdings stimmen die diese nicht mehr, da wir den Messbereich gegen Ende schnell geändert haben.

Gehäuse

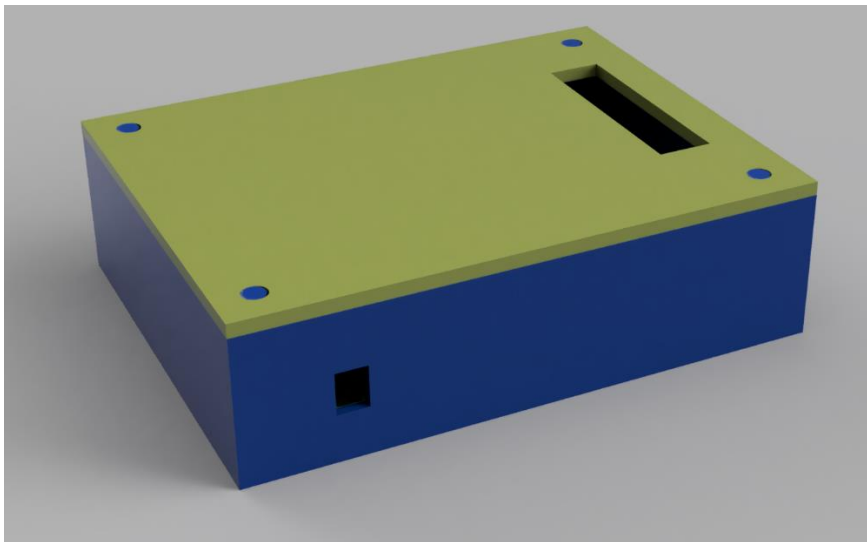
Aufgrund zeitlicher Schwierigkeiten konnte nur noch ein Gehäuseentwurf entwickelt, aber nicht mehr realisiert werden.

Das Gehäuse setzt sich aus einem Deckel (gelb) und einem Boden mit Wänden (blau) zusammen. Auf der Unterseite des Bodens sind Stäbe befestigt, welche perfekt mit den Löchern auf der Leiterplatte übereinstimmen und diese dann wie auf einer Schiene einführen lässt, sodass diese sicher drin liegt.

Die Stäbe werden auch dafür genutzt den Deckel zu befestigen, in Zukunft ist der Plan ein Einrastsystem zu designen, da der Deckel momentan noch abfallen kann, wenn man das Gehäuse auf den Kopf dreht.

Das Loch in der Decke dient dazu, dass man gute Sicht auf die LEDs hat und man das gemessene Signal auch ablesen kann.

In der Front ist ein kleines Loch, in welchem der Schalter perfekt reinpasst und umgeschaltet werden kann um das Board ein- und auszuschalten.



Inbetriebnahme

Ablauf

1. Optische Kontrolle
2. Langsam und mit Strombegrenzung
3. 5V nach Boost Converter einspeisen
4. Sensor messen
5. Referenzspannungen messen
6. LED kontrollieren
7. Netzgerät am Booster
8. Vcc kontrollieren
9. Mit Batterie testen

1. Optische Kontrolle bestanden

2. Vor diesem Schritt wurde am Messpunkt für das Vcc ein Draht eingelötet, um Vcc mit einem Netzteil zu simulieren.

3. 5V eingespeist

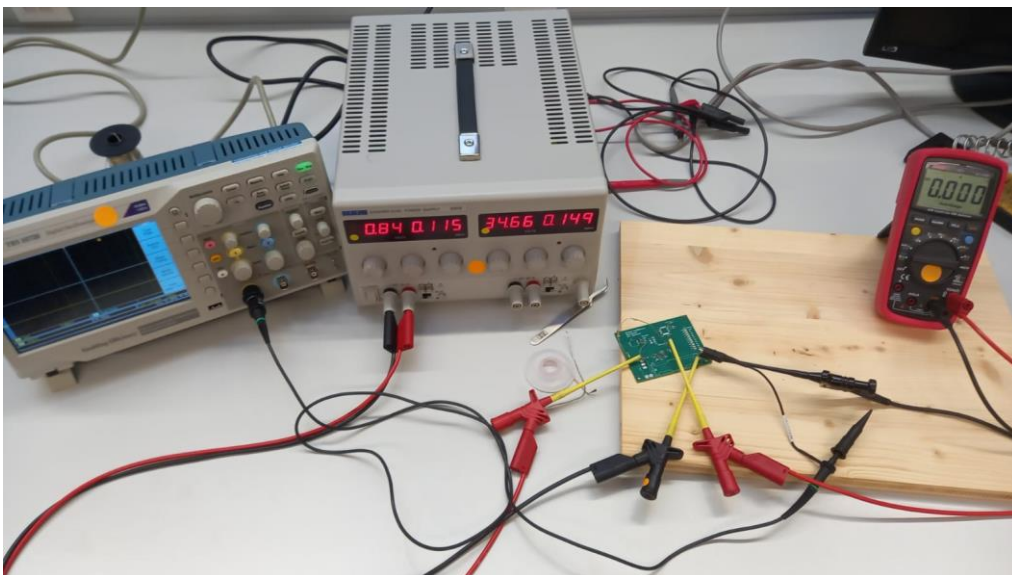
4. Sensor gibt das erwartete Signal von ca. 3,8V aus

5. Referenzspannung ist nicht richtig

6. Leds werden mit 100 Ohm Widerstand nach GND getestet

7.

Messaufbau:



Eingangsspannung: 1,5V

Eingangsstrom: 0,242 A

Ausgangsspannung: 4,8V

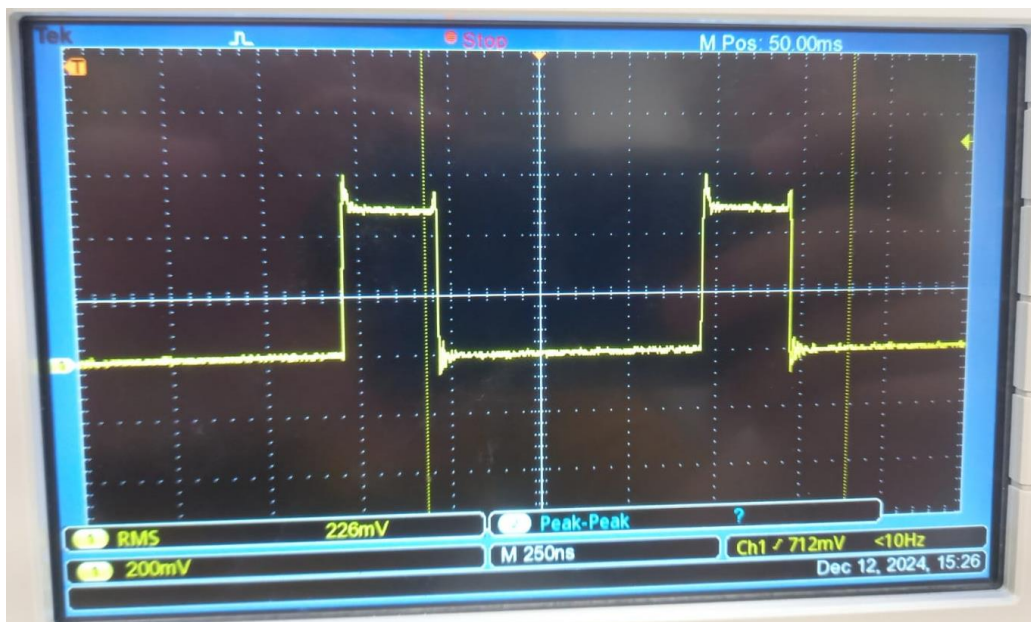
Pin: 0,3 W

Leistung an der Last: 0,23 W

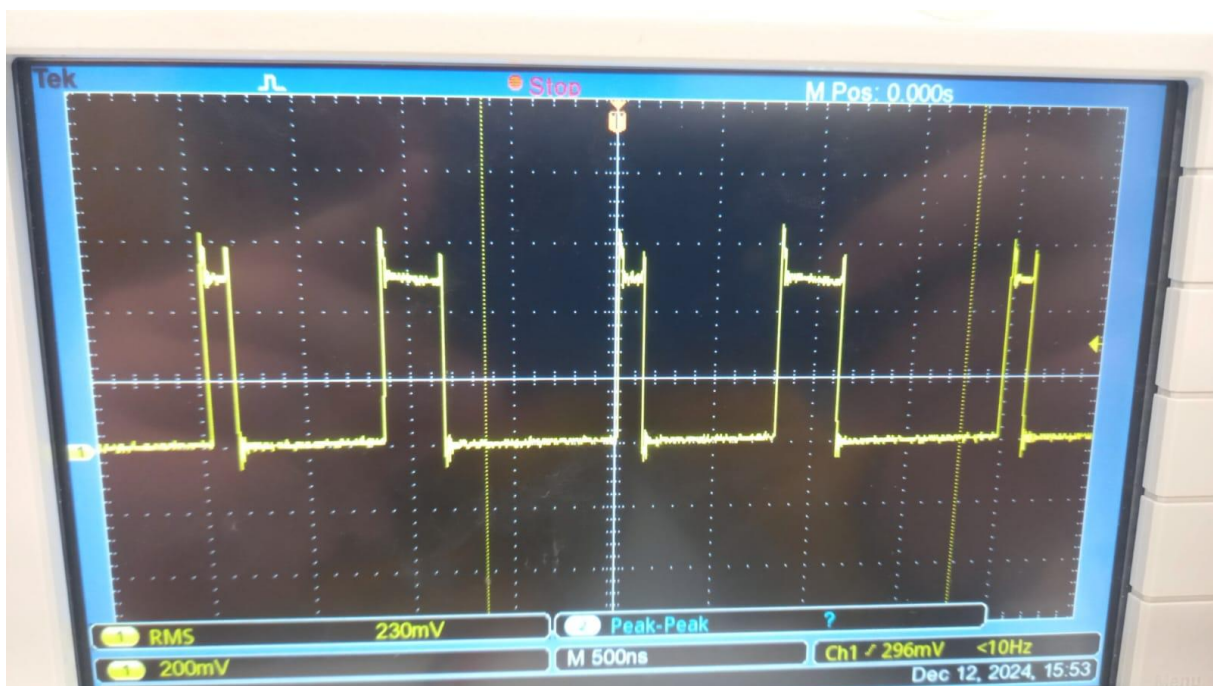
Last: 100 Ohm

BaroSense.docx

Oszilloskop:



Wenn wir die Last von 100 auf 50 Ohm senken, verändert sich das Signal merkbar:



Die Periodendauer und der Duty cycle ändern sich interessanterweise nicht.

Messungen mit Wärmebildkamera des BUC:

Ausgeschalteter Zustand:



Eingeschalteter Zustand:



Die Temperatur nimmt um ca. 15°C zu.

Softwareteil Megacard

Hier rinn uns die Zeit aus und der Code konnte nicht fertig debugged werden, weswegen er nicht funktioniert. Dennoch ist er fertig und es müsste nur auf Fehler überprüft werden.

Code:

```
/*-----*\
| Datei:   main.c
| Version: 1.0
| Projekt: Einfuehrung in das Programmieren des Atmega16
| Beschreibung: Grundgeruest fuer die Anzeigebibliothek
| Schaltung: MEGACARD V5.5
| Autor:
| Erstellung:
|
| Aenderung:
\*-----*/

#include <avr/io.h>
#include <util/delay.h>
#include "display.h"
#include <avr/interrupt.h>
#define Z_MAX 255 //maximaler ADC-Wert
#define Vs_max 5
ISR(ADC_vect)
{
    unsigned char Z = ADCH;
    float a = 0.009;
    float b = 0.095;
    unsigned char Vdd = 5;
    float Vsensor = Vs_max * Z/(Z_MAX +1);
    float P = (Vsensor/Vdd *a) - (b/a);
    display_printf_pos(0,6, "V: %2f", P);
}

void init(void)
{
    DDRA &= ~(1<<PA5); //Pin PA5 auf Eingang stellen
    char kanal = 5;

    ADCSRA |= (1<<ADSC); //Betriebsmodus auf Dauerwandlung stellen
    ADCSRA |= (1<<ADPS1) | (1<<ADPS2); //ADC Vorteiler auf 64 (f_ADC = 187,5 kHz)
    ADMUX |= kanal; //Kanal einstellen (unipolarer Messmodus)
    ADMUX |= (1<<ADLAR); //ADCW linksbündig lesen
    ADMUX |= (1<<REFS0); //Referenzspannung auf Vcc ~5V stellen
    ADCSRA |= (1<<ADEN); //ADC einschalten
    ADCSRA |= (1<<ADSC); //Wandlung starten
    ADCSRA |= (1<<ADIF); //Interrupt spezifisch freigeben
```

```
sei();                                //Interrupts global freigeben
}

int main (void)
{
    init ();    // Aufruf der Grundinitialisierungen
    display_init(); // Initialisierung der Anzeige

    display_string_pos (0, 4, " BaroSense sensing...");

    while (1)
    {

    }

    return 0;
}
```

Korrekturen

Es stellt sich heraus das im KiCad für den LED-Treiber das falsche Symbol verwendet wurde, bei welchem die Pins anders belegt sind. Deshalb musste die Platine angepasst werden und neu bestellt werden.

Als nächstes wurde die Referenzspannungsschaltung angepasst, da die Widerstandswerte falsch berechnet wurden was dazu führte das keiner der LEDs mehr leuchtete.

Es musste auch der Boost-Konverter ausgetauscht werden, welcher aus uns unbekannten Gründen kaputt gegen ist. Wir vermuten das er uns beim Löten kaputt gegangen ist.

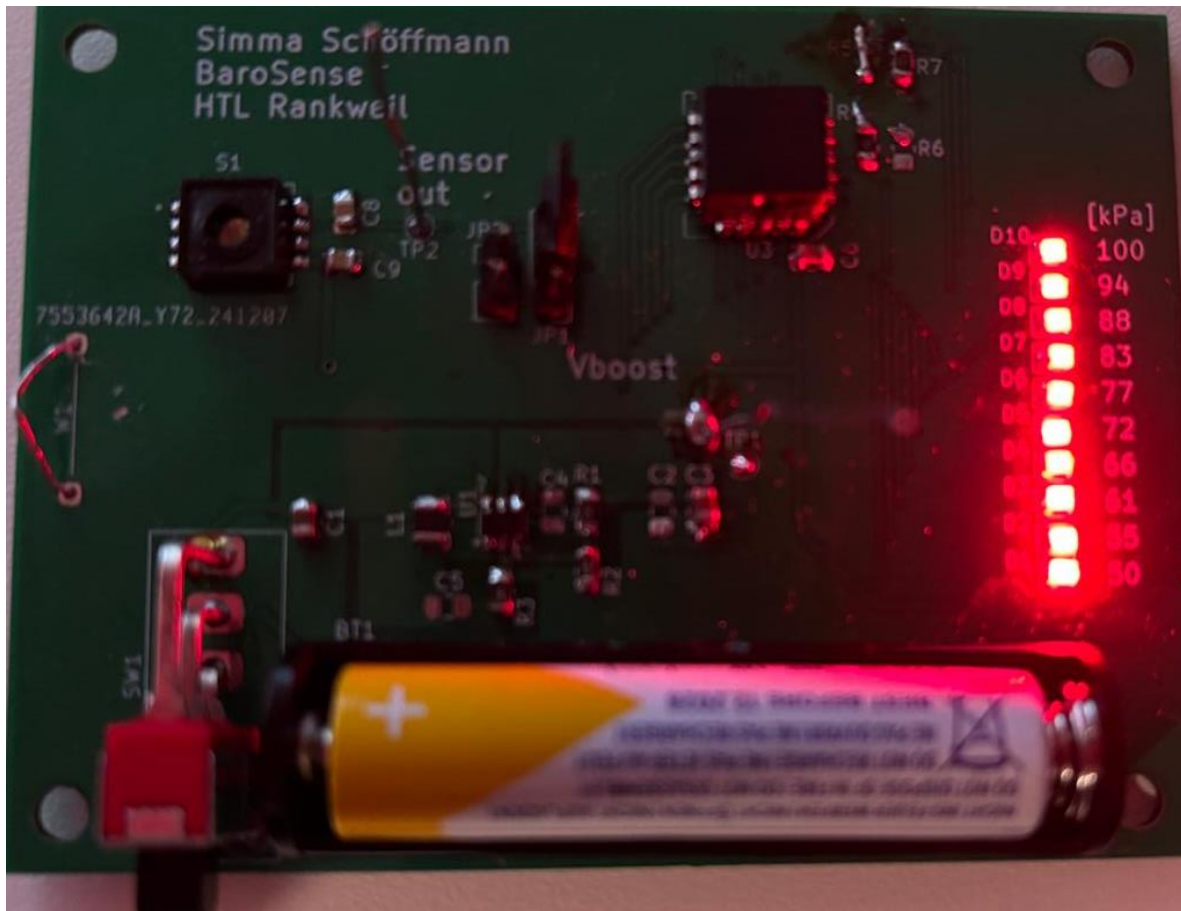
Zuletzt passten wir die Referenzspannungsschaltung erneut an, um einen kleineren Messbereich von 90kPa bis 100kPa zu erreichen.

In den letzten Augenblicken vor der Abgabe dämmert es mir, dass ich schon wieder vergessen habe das der Vadj 1.25V beträgt, welches laut Simulation nicht möglich ist, und deshalb vielleicht einer der Gründe ist, weshalb die Sensorspannung nicht richtig angezeigt ist.

Festhaltend lässt sich sagen, dass wir zwar leider kein Einwandfreies Projekt auf die Beine gebracht haben, allerdings eine Menge dazu gelernt haben, vor allem was Fehlersuche und Behebung angeht.

Ergebnisse

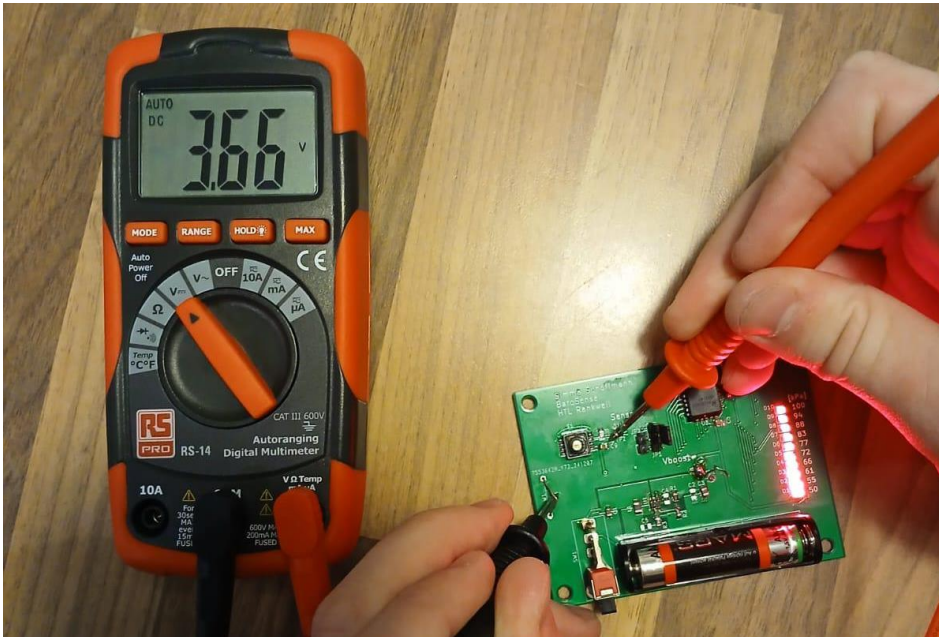
Fertig Bestückte und in Betrieb genommene Leiterplatte im angeschalteten Zustand.



Ein Test mit eingespeister Spannung als Video: https://youtu.be/A_37r9Q3tx8.

In Rankweil wurde das Ergebnis wie auf dem obigen Bild getestet, welches wir erwartet haben. Es war ursprünglich der Plan, dass Raphael einen Zeitraffer oder Bilder macht während langsam immer weniger LEDs leuchten, weil er nach Hause fährt und nach oben fährt. Bis er zuhause ankam ist allerdings nichts passiert, weswegen nochmal das Sensor Signal getestet wurde (Bild auf folgender Seite).

[Hier](#) sind wir zum Schluss gekommen, dass das Sensorsignal zwar nicht genau damit übereinstimmt mit was wir gerechnet haben, allerdings sollte auf den LEDs trotzdem der Wert angezeigt werden können, was hier aber nicht mehr der Fall ist, der Fehler ist [hier](#) beschrieben.

Messung des Sensor Signals am Sulzberg:

Ideen für zukünftige Projekte

Geschwindigkeitsmessung über Beschleunigungssensor und OP-Schaltung

Leistungsbeitrag Raphael

Schaltplan und Layout

SMD-Aufbau und bei Fehlern umlöten THT

Gehäusedesign

Fehlerbehebung

Dokumentation (40%)

Kilian unterstützen

Leistungsbeitrag Kilian

Berechnung und Simulation

Inbetriebnahme und Messungen

THT-Aufbau und teilweise SMD

Fehlerbehebung

Dokumentation (60%)

Raphael unterstützen