



LINZER TECHNIKUM

htl paul-hahn-straße

Höhere Technische Bundeslehranstalt

Höhere Lehranstalt für Elektrotechnik
Ausbildungsschwerpunkt industrielle Elektronik

HTL Diplomarbeit

FalloutMeter - stationäres Radioaktivitätsmessgerät

ausgeführt im

Schuljahr 2015/16

eingereicht von

Mihajlo Krečar 5AHET

Christian Haun 5AHET

Betreuer

Prof. Dr.
Dipl.-Ing. Christoph Lackner

Inhaltsverzeichnis

Eidesstattliche Erklärung	V
Danksagung	VII
1 Problem-Ziel-Lösung	1
1.1 Problem	1
1.2 Ziel	1
1.3 Lösung	1
2 Arbeitspakete	3
3 Erfassung der ionisierenden Strahlung	5
3.1 ionisierende Strahlung	5
3.1.1 Vorbemerkungen	5
3.1.2 physikalische Grundlagen	5
3.2 Messung	5
3.2.1 Auswahl des Sensors	5
3.2.2 Funktionsweise des Sensors	6
3.2.3 Kalibrierung	6
4 Elektronischer Aufbau	7
4.1 Vorbemerkungen	7
4.2 Finaler Entwurf	7
4.2.1 Spannungsversorgung	7
4.2.2 Heizung und Kühlung	8
4.2.3 Messen der ionisierenden Strahlung	14
4.2.4 Berechnung der Zuleitungsquerschnitte	15
4.3 Fertigung	16
4.3.1 Layoutentwurf	17
4.3.2 Belichten	37
4.3.3 Entwickeln	38
4.3.4 Ätzen	38
4.3.5 Aufbringen der Lötstopmmaske	39
4.3.6 Bohren	40
4.3.7 Löten	41

5	Mechanischer Aufbau	43
5.1	Die ersten Modellentwürfe	43
5.1.1	Solid Edge ST4	43
5.1.2	Autodesk Inventor 2016	43
5.2	Gehäusebau	46
5.2.1	Gehäuseoberteil	46
5.2.2	Gehäuseunterteil	48
5.2.3	Vollendung des mechanischen Aufbaus	51
6	Softwareentwicklung der Messschaltung	59
6.1	Software für den ATMEL ATmega328P	59
6.1.1	Bestimmen der Regelparameter der Gehäusetemperierung	60
7	Kommunikation	63
7.1	KNX	63
7.1.1	Allgemein	63
7.1.2	Topologie	63
7.1.3	Telegrammaufbau	65
7.1.4	Weinzierl BAOS-Modul	66
7.2	GSM	69
7.2.1	Allgemein	69
7.2.2	GSM Alarmierung	71
7.3	Push-Benachrichtigung mit Prowl	74
7.3.1	Vorbemerkungen	74
7.3.2	Umsetzung der Push-Mitteilung	75
8	Visualisierung der Messwerte	77
8.1	Smarthome.py	77
8.1.1	Raspberry Pi vorbereiten	77
8.1.2	Smarthome.py einrichten	77
8.1.3	SmartVisu einrichten	81
8.1.4	eibd einrichten	81
8.1.5	SmartVisu Webseite konfigurieren	85
8.2	CubeVision	95
8.2.1	Konfiguration	95
8.2.2	Topologie	96
8.2.3	Parametrierung der Werte	97
	Verzeichnisse	I
	Literaturverzeichnis	V
A	Anhang	VII
A.1	Pflichtenheft	VIII

A.2	Kostenaufstellung FalloutMeter	XVII
A.3	Kostenaufstellung Visualisierung	XX
B	weitere Dokumente	1
B.1	Projekttagbücher	1
B.1.1	Projekttagbuch von Mihajlo Krečar	1
B.1.2	Projekttagbuch von Christian Haun	5

Eidesstattliche Erklärung

Wir erklären Eides statt, dass wir die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht verwendet und die den benutzten Quellen inhaltlich beziehungsweise wörtlich entnommenen Stellen als solche kenntlich gemacht haben.

Linz, April 2016

Christian Haun

Mihajlo Krečar

Danksagung

Zunächst möchten wir uns an dieser Stelle bei all denjenigen bedanken, die uns während der Anfertigung dieser Abschlussarbeit unterstützt und motiviert haben. Dazu gehört, neben unseren Eltern, Daniel Haun, der in vielen anregenden Gesprächen wichtige und hilfreiche Tipps sowie Ideen zu dieser Diplomarbeit beigetragen hat. Weiters bedanken wir uns für die Bereitstellung diverser Komponenten, wie etwa Messinstrumente, da es nur durch sie möglich war, dieses Projekt in vollem Umfang zu realisieren.

Ganz großen Dank wollen wir noch unserer Betreuungslehrkraft Herrn Prof. Dr. Dipl.-Ing. Christoph Lackner aussprechen. Nicht nur gab er uns wertvolle Hinweise und Anregungen, sondern auch moralische Unterstützung und Motivation wurden uns zuteil.

Ganz herzlichen Dank auch an Žarko Krečar, der uns tatkräftig beim mechanischen Aufbau des Gehäuses unterstützt hat und uns immer wieder mit Verbesserungsvorschlägen bereichert hat.

Auch Fabian Plaimauer gilt für seine Fähigkeiten im Bereich des Platinendesigns, sowie für seine Hilfe bei der Erstellung des 3D-Modells der Platine großer Dank .

Weiters gilt besonderer Dank der Firma Weinzierl Engineering GmbH für die großzügige Spende der Komponenten zur KNX-Anbindung.

Vorstellung der Autoren



Mihajlo Krečar

Hauptverantwortlicher für den
mechanischen Aufbau des Gehäuses
weilers verantwortlich für Visualisierung
und GSM-Alarmierung mit dem BabTec
eibPort



Christian Haun

Verantwortlich für Fertigung der Platine
Hauptverantwortlicher für Programmierung
und KNX-Kommunikation
weilers verantwortlich für die Visualisierung
mit der SmartVisu

Kurzfassung

Diese Diplomarbeit behandelt die Entwicklung und den Bau eines stationären Strahlungsmessgerätes für ionisierende Strahlung.

Dieses Messgerät verfügt über eine KNX-Busanbindung, den weltweiten Standard für Haus- und Gebäudeautomatisierung, und ist somit der weltweit erste KNX-Umweltsensor für ionisierende Strahlung.

Wir alle wissen, dass die Gefahr die von Radioaktivität herrührt, nicht zu unterschätzen ist. Wir sehen sie nicht, wir riechen sie nicht, gerade deshalb ist es unabdingbar unsere eigenen Messinstrumente zu installieren, um sie zu erkennen. Wie sollte so etwas besser machbar sein, als mit einem stationären Messinstrument, welches mit der Hausautomatisierung gekoppelt ist und auch Alarmierungen über das GSM-Netz versenden kann? Mithilfe dieses Messgerätes ist es möglich, die Messwerte gut verständlich auf Tablet oder Smartphone als Diagramm anzeigen zu lassen.

Diese Diplomarbeit umfasst somit sämtliche erforderlichen Schritte der Fertigung eines komplett neuen Produktes, sprich Brainstorming, Marktanalyse, Prototyping, Softwareentwicklung und anschließender Funktionstest des Produktes.

Abstract

This diploma thesis discusses the development and the construction of a steady radiation gauge for ionizing radiation. It provides a KNX-Bus-connection, the worldwide standard for building automation, therefore this is the first radiation gauge for KNX. We all know, that the threat of radioactivity shall not be underrated. We can't see it, we can't smell it, due to this, it is necessary to install our own gauges to detect it. How can this be better done, than with a steady gauge, which is coupled with the home automation and is able to send alarms over the GSM-network?

With the help of this gauge, you can view the measurement data on any smartphone or tablet in a chart.

This diploma thesis includes every step of product development, i. e. brainstorming, market analysis, prototyping, software development and test of the product.

1 Problem-Ziel-Lösung

1.1 Problem

Das Problem war das Fehlen einer Lösung zur Überwachung der aktuellen Strahlenbelastung im Heimumfeld welches geeignete Kommunikationsmöglichkeiten, wie etwa KNX oder GSM, bietet. Am Markt war zum Zeitpunkt der Recherche kein für den Heimgebrauch taugliches stationäres Messgerät verfügbar, welches an die KNX-Hausautomatisierung angebunden werden kann.

1.2 Ziel

Das Ziel war die Entwicklung und der Aufbau eines stationären Strahlenmessgerätes. Es soll über eine KNX-, sowie GSM-Anbindung verfügen. KNX soll deshalb verwendet werden, da es den Standard in der Gebäudeautomatisierung darstellt, sicher funktioniert und da die Visualisierung der Messwerte leicht realisiert werden kann. Mit KNX kann man sich auch einen guten Überblick über das Eigenheim verschaffen, indem lokale Regen-, Temperatur- und nun auch Strahlenmesswerte übersichtlich dargestellt werden können. Mithilfe der GSM-Anbindung kann man auch unterwegs sicher informiert werden, da das GSM-Netz große Teile des Landes abdeckt.

1.3 Lösung

Die Lösung war die Messstation mit einem Mikrocontroller und einem Halbleitersensor aufzubauen. Zur KNX-Anbindung soll ein Weinzierl BAOS-Modul 820 verwendet werden.

2 Arbeitspakete

Nr.	Datum	Beschreibung	Bearbeiter
0	30.10.2015	Gesamtkonzept erstellen	Krečar, Haun
1	27.11.2015	Hardwareentwicklung (Schaltplan, tlw. Simulation sofern notwendig, Layout, Fertigung Prototypen und Produkt)	Krečar, Haun
2	01.01.2016	Programmierung	Haun
3	02.02.2016	Testen, Kalibrieren	Krečar, Haun
4	25.03.2016	Visualisierung und Inbetriebnahme	Krečar, Haun

3 Erfassung der ionisierenden Strahlung

3.1 ionisierende Strahlung

3.1.1 Vorbemerkungen

Im Land Österreich gibt es Messstationen zum Messen von ionisierender Strahlung. [1]

Diese Messwerte befinden sich hierbei laut [2] im Bereich zwischen $70 \frac{nSv}{h}$ und $200 \frac{nSv}{h}$. Dies entspricht dem Bereich $0,07 \frac{\mu Sv}{h}$ bis $0,2 \frac{\mu Sv}{h}$

3.1.2 physikalische Grundlagen

Wenn sich im Atomkern eines Stoffes eine für die Protonenzahl geeignete Anzahl von Neutronen befindet, ist der Kern in seinem Grundzustand stabil. Weicht der Wert der Neutronen jedoch ab, wird dieser Kern instabil und es setzt somit ein radioaktiver Zerfall ein.

Die wichtigsten Zerfallsarten sind hierbei: α -, β - und γ -Zerfall.

Beim α -Zerfall werden Heliumkerne ausgesendet, beim β -Zerfall Elektronen (selten auch aus Positronen) und beim γ -Zerfall Photonenstrahlung mit hoher Frequenz. [3]

Wenn der menschliche Körper ionisierende Strahlung absorbiert, entstehen dadurch im Körper aus neutralen Teilchen elektrisch geladene Ionen. Diese Ionen sind sehr reaktionsfreudig, wollen also eine stabile Verbindung eingehen. Die dabei entstehenden Verbindungen können für den Körper gefährlich sein.

Dies kann dazu führen, dass Enzyme ihre Funktion einbüßen, oder sogar ganze Zellen zerstört werden. Die Strahlung kann auch zur Beschädigung des Erbmaterials führen, was wiederum Krebs auslösen kann. [4]

3.2 Messung

3.2.1 Auswahl des Sensors

In dieser Arbeit wurde ein halbleiterbasierender Sensor gewählt, da dieser sich leichter in die Schaltung integrieren und sich gut mit einem Mikrocontroller auswerten lässt.

Nach ausgiebiger Marktanalyse haben wir uns für einen Sensor der Firma Teviso Sensor Technologies Ltd. entschieden, das Modell RD2014 erfüllt alle notwendigen Anforderungen.

3.2.2 Funktionsweise des Sensors

Der Sensor Teviso RD2014 misst die Strahlung mittels PIN¹-Dioden. Bei diesen Halbleitern handelt es sich um Dioden, welche eine zusätzliche undotierte Schicht zwischen den p-, und n-Schichten besitzt. Dies führt zu einer höheren Durchlassspannung, als bei normalen Dioden. Trifft nun ein Strahlungsimpuls auf solch eine Diode, leitet sie für einen kurzen Moment. Dieser Impuls wird verstärkt und mittels einer integrierten Komparatorschaltung mit einer Referenzspannung verglichen. Ist der Impuls stark genug, gibt der Sensor einen messbaren Zählimpuls aus. Aus der Summe dieser Impulse über einer gewissen Zeit lässt sich nun der Messwert berechnen.

In Abbildung 3.2.1 sieht man das Blockschaltbild des Sensors.

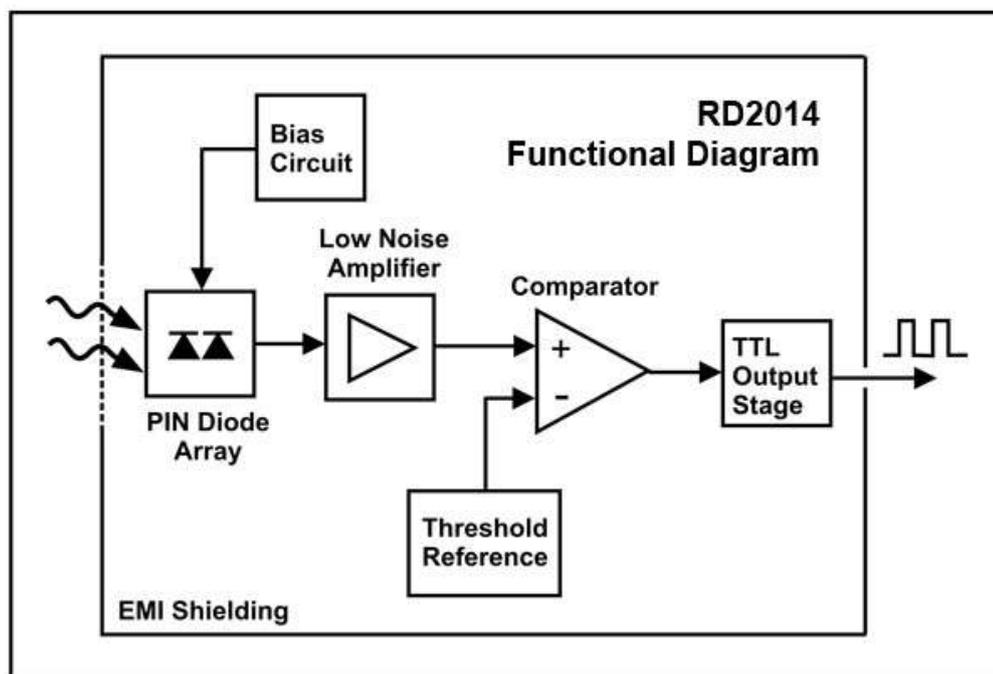


Abbildung 3.2.1: Teviso RD2014 Blockdiagramm

3.2.3 Kalibrierung

Wie uns spät zuteil wurde, ist der verwendete Strahlensensor zur Kalibrierung in der EU nicht zugelassen. Nichtsdestotrotz ist es uns erlaubt, diesen Sensor zu verwenden, das dies nicht unter kommerzieller Absicht geschieht.

¹PIN-Diode Englisch für positive intrinsic negative diode

4 Elektronischer Aufbau

4.1 Vorbemerkungen

Der erste Entwurf der Messschaltung wurde in der Software Eagle erstellt. Dieser umfasste das BAOS-Modul 820 der Firma Weinzierl (notwendig zur KNX-Anbindung), den Radioaktivitätssensor RD2014 der Firma Teviso, einen I2C Luftfeuchtigkeits-, und Temperatursensor AM2321 der Firma Aosong sowie den Mikrocontroller ATmega328P der Firma Atmel. Die Kühlung sowie Heizung der Schaltung wurde auch bedacht, da diese sich ja im Normalfall im Außenbereich befindet. So wurde noch ein Lüfter und ein Heizelement hinzugefügt. Das Heizelement sollte direkt mit einem MOSFET vom Mikrocontroller angesteuert werden, wobei für den Lüfter eine I2C-fähige Regelung mit dem MAX6650 der Firma Maxim vorgesehen war.

4.2 Finaler Entwurf

4.2.1 Spannungsversorgung

Zur Spannungsversorgung wird wegen seiner recht hohen Leistung der Step-Up-Wandler LMR14050 der Firma Texas Instruments verwendet. Diese setzt die 24 V SELV-Spannung am Eingang in 5 V und maximal 5 A um.

Berechnung der Bauteilwerte

Die Bauteilwerte wurden mithilfe der Formeln im Datenblatt des LMR14050 [5] berechnet. Unsere vorgegebenen Werte seien:

- $V_{OUT} = 5 V$
- $f_{SW} = 300 kHz$
- $V_{IN_MAX} = 24 V$
- $I_{OUT} = 5 A$
- $R_{FBT} = 100 k\Omega$
- $K_{IND} = 0,4$

Stellt man nun Gleichung 4.1 auf R_{FBB} um, lässt sich dessen Wert ermitteln.

$$R_{FBT} = \frac{V_{OUT} - 0,75}{0,75} \cdot R_{FBB} \longrightarrow R_{FBB} = R_{FBT} \cdot \frac{0,75}{V_{OUT} - 0,75} = 17,65 k\Omega \quad (4.1)$$

Man kann hier den nächsten verfügbaren Wert wählen, in diesem Fall wäre dies 17,8 kΩ.

R_T berechnet sich nach Gleichung 4.2.

$$R_T(k\Omega) = 32537 \cdot f_{SW}(kHz)^{-1,045} = 83,91 k\Omega \quad (4.2)$$

Die benötigte Induktivität des Step-Down-Converters kann man nach Gleichung 4.3 auswählen.

$$L_{MIN} = \frac{V_{IN_MAX} - V_{OUT}}{I_{OUT} \cdot K_{IND}} \cdot \frac{V_{OUT}}{V_{IN_MAX} \cdot f_{SW}} = 6,597 \mu H \quad (4.3)$$

Es wurde eine Spule mit einer Induktivität von $8,2 \mu H$ und einem maximalen Strom von $7 A$ gewählt.

Die restlichen Bauteilwerte wurden aus den Empfehlungen des Datenblattes entnommen, folgende Werte wurden also gewählt:

- $C_{OUT} = 180 \mu F$
- $C_{IN} = 4,7 \mu F$
- $C_{SS} = 22 nF$
- $C_{BOOT} = 100 nF$

Zuletzt muss noch die Schottky-Diode ausgewählt werden. Der Durchlassstrom I_F muss lediglich hoch genug gewählt werden, in diesem Fall beträgt dieser Wert $I_F = 7 A$.

4.2.2 Heizung und Kühlung

Peltier-Element

Das Heizelement, sowie der Lüfter wurden im finalen Entwurf wieder verworfen, da die Möglichkeit durch ein Peltier-Element heizen und kühlen zu können besser erschien. Im ersten Entwurf war die Heizung und die Kühlung auf zwei getrennte Systeme aufgeteilt, was nun jedoch, durch den Einsatz eines Peltier-Elements überflüssig wurde.

Ein Peltier-Element erzeugt bei Stromfluss eine Temperaturdifferenz zwischen den beiden Seiten des Bauteils, damit arbeitet es praktisch als Wärmepumpe. Die Entstehung dieser Temperaturdifferenz basiert auf den nach Jean Peltier benannten »Peltier-Effekt«.

Dieser Effekt geht folgendermaßen vonstatten: Verbindet man 2 unterschiedlich stark elektrisch leitfähige Materialien an zwei Enden und legt eine elektrische Spannung an, so kommt es zum Stromfluss. Dieser Stromfluss transportiert Wärme von einer Verbindungsstelle zur anderen. Somit wird eine kalt und die andere warm. Gute Stromleiter sind jedoch meist auch gute Wärmeleiter, deshalb kommen für Peltier-Elemente nur Materialien mit einer hohen elektrischen Leitfähigkeit, jedoch geringer thermischen Leitfähigkeit in Frage. Hierbei kommen in der Regel Halbleiter zum Einsatz.

Das Peltier-Element soll durch eine mit PWM angesteuerte H-Brücke mit entsprechender Drosselspule geregelt werden. Es gibt zwar vom Hersteller empfohlene Methoden, das Peltier-Element nur mit PWM anzusteuern, die Methode den Strom entsprechend zu glätten bringt jedoch einen höheren Wirkungsgrad und eine höhere Lebensdauer mit

sich. Mit der Methode der H-Brücke lässt sich das Peltier-Element zum Heizen, sowie zum Kühlen verwenden, da man einen positiven, als auch negativen Strom über die Last schicken kann.

Eine Seite des Peltier-Elements ist gut leitend mit dem Gehäuse verbunden, auf der anderen Seite befindet sich ein Kühlkörper. Das Peltier-Element wurde an beiden Seiten mit Wärmeleitkleber angebracht. Die Temperatur im Gehäuseinneren darf sich im Bereich von -5°C bis 45°C bewegen, ansonsten befinden sich die eingesetzten Bauteile nicht mehr im erlaubten Temperaturbereich und könnten fehlerhaft arbeiten.

Messung von Temperatur und Luftfeuchtigkeit

Zum Messen der Temperatur, sowie der Luftfeuchtigkeit im Gehäuseinneren wird der Aosong AM2321 verwendet. Dieser Sensor kommuniziert über I2C, somit ist eine leichte Anbindung an den Mikrocontroller möglich.

Um trotz der softwaremäßigen Beschränkung die Spannung und somit den Strom am Peltier-Element nicht zu groß werden zu lassen, wird eine Feinsicherung von $3,5\text{ A}$ in der Zuleitung des Peltier-Elements platziert.

Berechnung der Bauteilwerte

Die nötige Induktivität zum Glätten des Stromes kann laut Datenblatt des H-Brücken-IC »TI DVR592« [6] berechnet werden, siehe Gleichung 4.4.

Hierbei ist V_O die Spitzenspannung des PWM-Signals, V_{TEC} der Mittelwert dieser Spannung, D das Tastverhältnis, T_S die Periodendauer des PWM-Signals und L die Induktivität.

$$\Delta I_L = \frac{(V_O - V_{TEC}) \cdot D \cdot T_S}{L} \quad (4.4)$$

Stellt man nun diese Gleichung auf L um, so erhält man Gleichung 4.5.

$$L = \frac{(V_O - V_{TEC}) \cdot D \cdot T_S}{\Delta I_L} \quad (4.5)$$

Laut [7] ist beim Ansteuern eines TEC ein Rippelstrom von 2% zu vertreten, der Rippelstrom ΔI_L berechnet sich also nach Verhältnis 4.6. Der Maximalstrom I_{Lmax} durch das Peltier-Element ist in diesem Fall $3,7\text{ A}$.

$$\frac{\Delta I_L}{2\%} = \frac{I_{Lmax}}{100\%} \rightarrow \frac{\Delta I_L}{2\%} = \frac{3,7\text{ A}}{100\%} \quad (4.6)$$

Stellt man nun das Verhältnis auf ΔI_L um, berechnet sich der Wert nach Gleichung 4.7.

$$\Delta I_L = 2\% \cdot \frac{3,7\text{ A}}{100\%} = 0,074\text{ A} \quad (4.7)$$

Der maximale Ripplestrom beträgt also 74 mA .

Mit einem PWM-Signal mit $V_O = 5\text{ V}$, $V_{TEC} = 1,9\text{ V}$, $D = 0,38$ und $T_S = \frac{1}{28,8\text{ kHz}} = 34,72\mu\text{s}$ kann man nun L , laut Gleichung 4.8, berechnen.

$$L = \frac{(5\text{ V} - 1,9\text{ V}) \cdot 0,38 \cdot 34,72\mu\text{s}}{0,074\text{ A}} = 552,74\mu\text{H} \quad (4.8)$$

Da im Schaltungsaufbau 2 Induktivitäten verwendet werden, werden die nötigen Werte laut Gleichung 4.9 berechnet.

$$L_1 = L_2 = \frac{L}{2} = \frac{552,74\mu\text{H}}{2} = 276,37\mu\text{H} \quad (4.9)$$

Wir wählten pro Induktivität einen Wert von $330\mu\text{H}$, da dieser leicht erhältlich ist.

Simulation der Bauteilwerte

Vor dem tatsächlichen Aufbau der Schaltung wurde diese in LTSpice simuliert.

Wie man in Abbildung 4.2.1 sieht, wurde in der Simulation die Ausgangsstufe der H-Brücke aufgebaut. Der ohmsche Widerstand des Peltier-Elementes beträgt laut Datenblatt [8] $0,75\Omega$

Folgende Simulation zeigt die Ausgangsstufe zuerst ohne die eingebauten Kondensatoren. In Abbildung 4.2.2 ist das aufgenommene Diagramm ersichtlich.

Abbildung 4.2.3 zeigt, dass sich bei dieser Simulation ein Ripplestrom von $55,6137\text{ mA}$ einstellt.

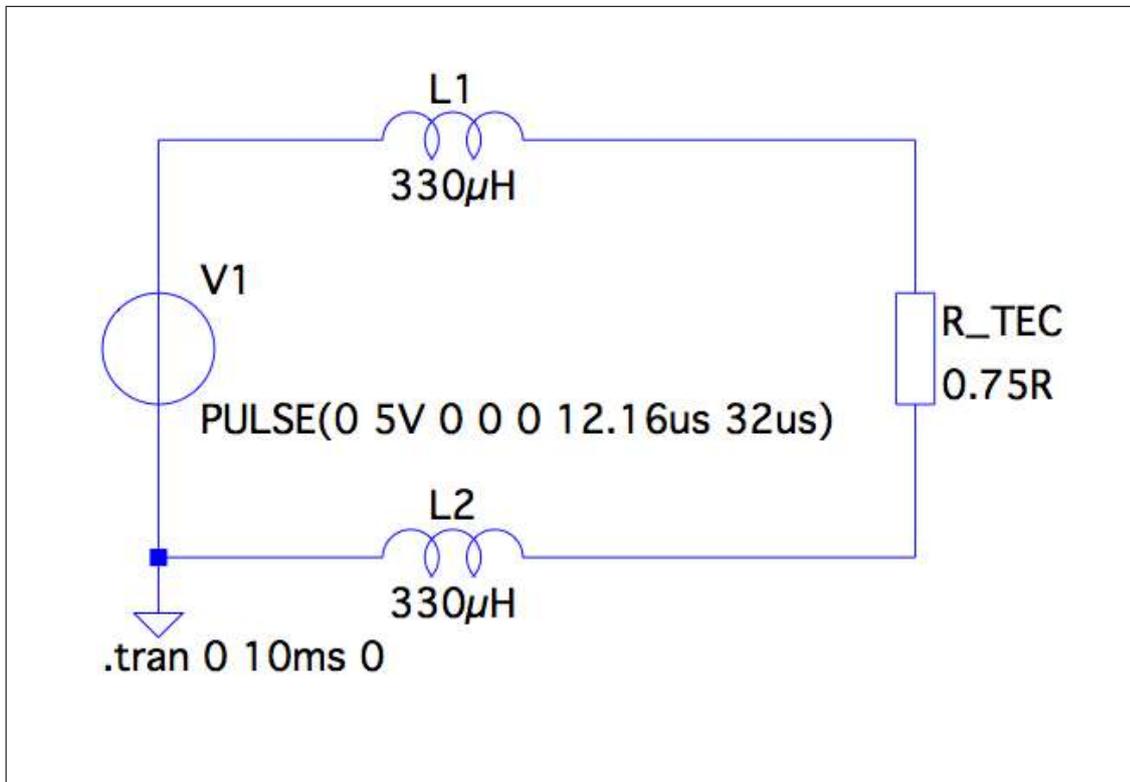


Abbildung 4.2.1: TEC ohne Kondensatoren Ausgangsstufe

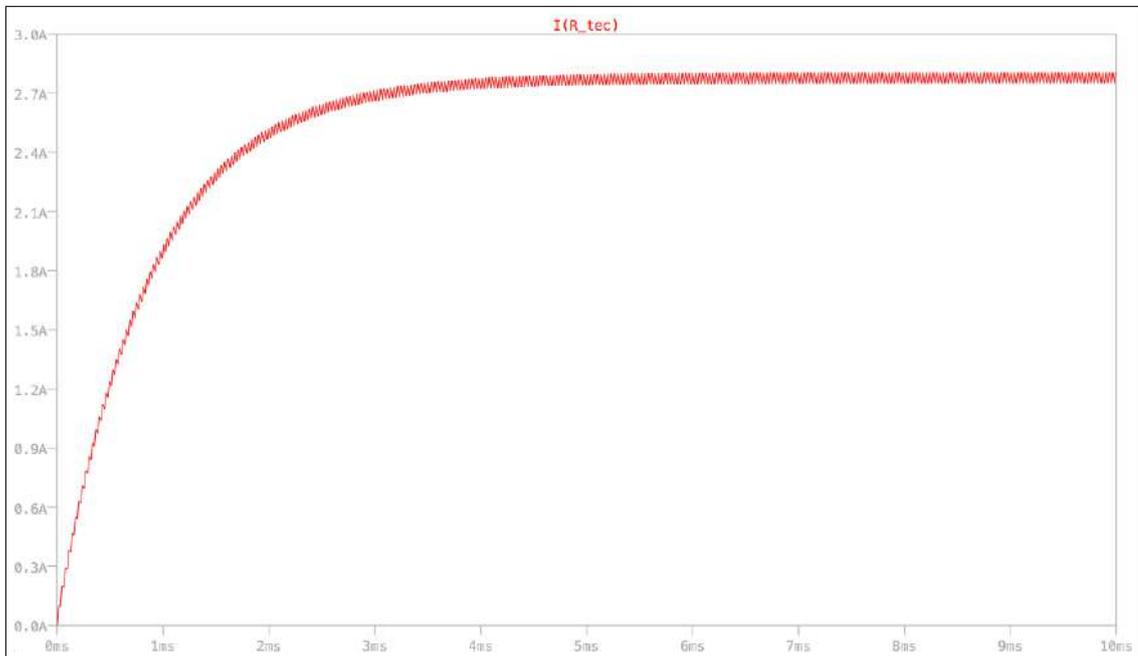


Abbildung 4.2.2: TEC ohne Kondensatoren Ausgangsdiagramm

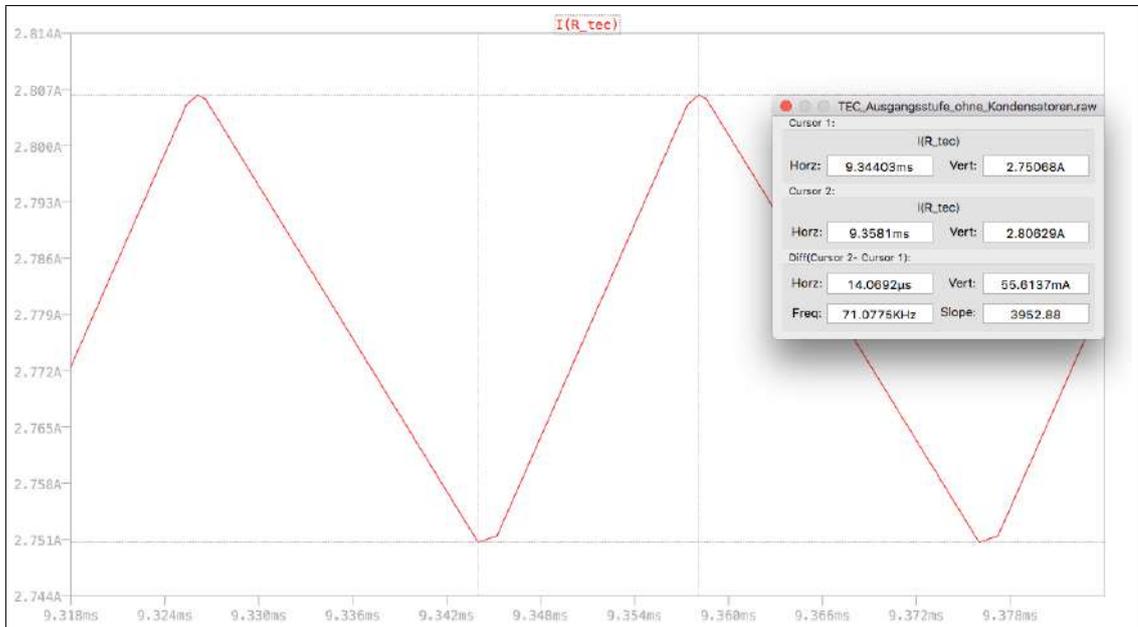


Abbildung 4.2.3: TEC ohne Kondensatoren Rippelstrom

In Abbildung 4.2.4 und 4.2.5 sieht man nun die Simulation mit Kondensatoren.

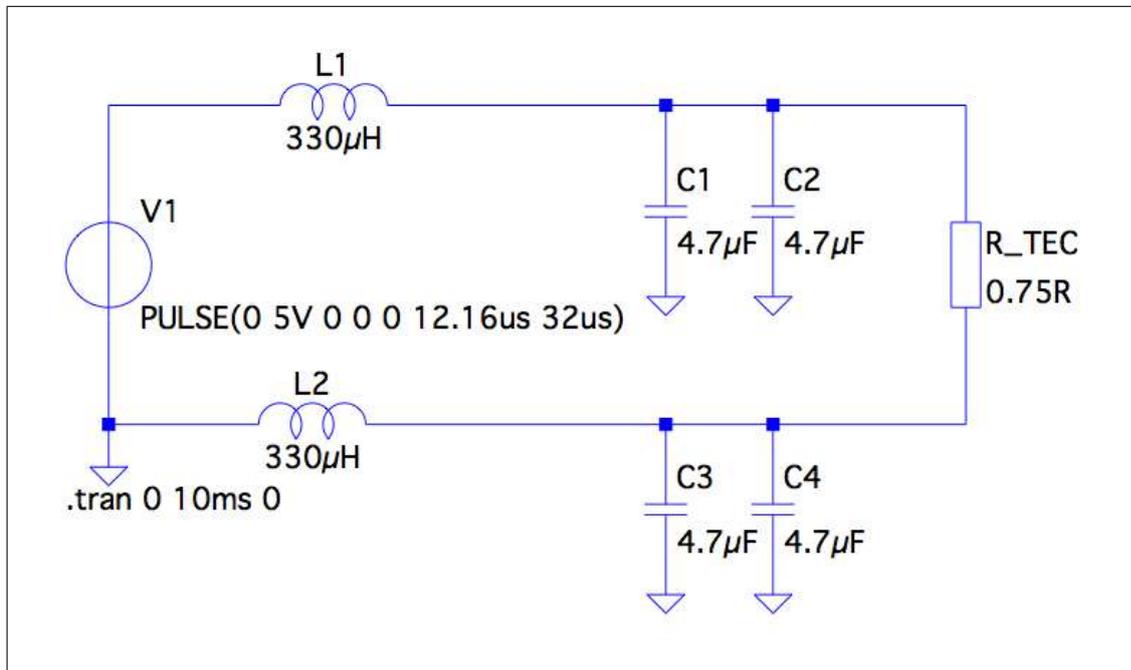


Abbildung 4.2.4: TEC mit Kondensatoren Ausgangsstufe

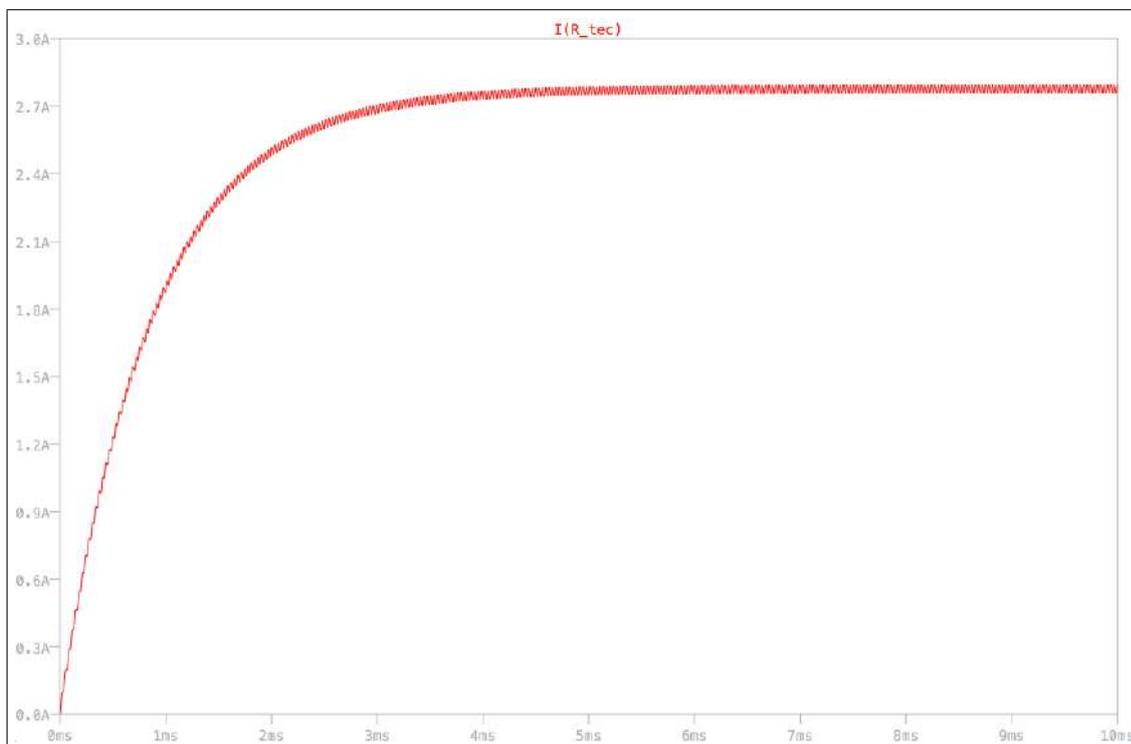


Abbildung 4.2.5: TEC mit Kondensatoren Ausgangsdiagramm

Erfassung der Temperatur und der Luftfeuchtigkeit

Zum Messen der Luftfeuchtigkeit sowie der Temperatur wird der Aosong AM2321 I2C-Sensor [9] verwendet. Anhand dieser Messwerte wird die Heizung, bzw. Kühlung der Schaltung geregelt sowie ein Störtelegramm auf den KNX-Bus gesendet, falls diese die Maximalwerte überschreiten.

Auslegung des Kühlkörpers für das Peltier-Element

Laut den Peltier-Handhabungshinweisen [10] kann Formel 4.10 zur Kühlkörperauslegung herangezogen werden.

Unsere Werte seien $Q_{max} = 8,5 W$ und $U_{max} = 3,8 V$.

$$R_{th} \leq \frac{15 K}{Q_{max}} = 1,76 \frac{K}{W} \quad (4.10)$$

Unser Kühlkörper hat einen thermischen Widerstand von $R_{th} = 1,45 \frac{K}{W}$, er ist also für diese Anwendung geeignet.

4.2.3 Messen der ionisierenden Strahlung

Nach ausgiebiger Internetrecherche erschienen die Sensoren der schweizer Firma Teviso als am besten geeignet für dieses Projekt. Wir haben uns für den Sensor RD2014 entschieden, da er einen Kompromiss aus Genauigkeit und Temperaturfestigkeit darstellt.

Die Firmenbeschreibung klang vielversprechend:

»Teviso Sensor Technologies Ltd. entwickelt Strahlungssensoren auf Halbleiter Basis, die für die Messung von Beta-, Gamma- und Röntgenstrahlung bestimmt sind. Diese Sensoren sind kostengünstig und dank ihrer Robustheit geeignet, in vielen Anwendungen das traditionelle Geiger-Müller Zählrohr abzulösen.

Sensoren für radioaktive Strahlung in Festkörper Technologie Ltd. werden in der Forschung bereits seit mehreren Jahren eingesetzt. Sie sind relativ groß und komplex und vor allem sehr teuer. Mit den neuen Produkten ist es Teviso gelungen, kleine und vor allem kostengünstige Sensoren zu entwickeln, die primär als Herzstück in batteriebetriebenen Strahlungsmessgeräten verwendet werden können.

Teviso hat sich zum Ziel gesetzt, innovative Produkte auf hohem Stand der Technik zu entwickeln und zu produzieren. Der Technologiestandort Schweiz bietet für TEVISO ideale Voraussetzungen, den hohen Ansprüchen an Präzision und Zuverlässigkeit gerecht zu werden.« [11]

Mit diesen Sensoren ist es möglich, die Messwerte mit dem Mikrocontroller optimal auszuwerten, da der Sensor über einen TTL-Zählausgang verfügt. Abbildung 4.2.6 zeigt den Sensor.



Abbildung 4.2.6: Teviso RD2014

4.2.4 Berechnung der Zuleitungsquerschnitte

Die Zuleitung besteht aus einem KNX-Kabel, auf dessen Beilaufdrähten die Versorgungsspannung von 24 V Gleichspannung für die Messschaltung angelegt wird. Die Lizen dieses Kabels bestehen aus Kupfer.

Der maximale Strom sei $I_{1max} = 589\text{ mA}$. Dieser wurde gemessen, während die Schaltung unter maximaler Last arbeitete. Die Nennspannung $U_1 = 24\text{ V}$, der spezifische Widerstand von Kupfer $\rho_{Cu} = 1,721 \cdot 10^{-2} \frac{\Omega \cdot \text{mm}^2}{\text{m}}$, die Leiterlänge $l = 25\text{ m}$ und der Querschnitt der Leitung $A = 0,75\text{ mm}^2$.

Der Widerstand dieser Leitung lässt sich nun mit Gleichung 4.11 berechnen.

$$R = \frac{\rho \cdot l}{A} = 0,57\ \Omega \quad (4.11)$$

Nun lässt sich der Spannungsabfall der Leitung nach Gleichung 4.12 bestimmen.

$$\Delta U = R \cdot I_{1max} = 0,34\text{ V} \quad (4.12)$$

Für eine Anwendung wie diese, ist laut TAEV ein Spannungsabfall von 3% vertretbar, die Gleichungen 4.13 und 4.14 zeigen, dass die gewählte Leitung ausreichend ist.

$$\frac{U_1}{\Delta U_{max}} = \frac{100}{3} \longrightarrow \Delta U_{max} = \frac{U_1 \cdot 3}{100} = 0,72\text{ V} \quad (4.13)$$

$$\Delta U < \Delta U_{max} = 1 \quad (4.14)$$

Der Spannungsabfall liegt im erlaubten Bereich und die Leitung ist somit geeignet. Entsprechend des maximalen Zuleitungsstroms kann auch die Eingangssicherung der Schaltung gewählt werden, in diesem Fall bietet sich ein Wert von 600 mA an.

4.3 Fertigung

Die Herstellung von Platinen gliedert sich grundlegend in 7 Arbeitsschritte:

1. Layoutentwurf
2. Belichten
3. Entwickeln
4. Ätzen
5. Aufbringen der Lötstopmmaske
6. Bohren
7. Löten

Um Platinen wie folgend beschrieben fertigen zu können, sind folgende Komponenten vonnöten.

- fotobeschichtetes Basismaterial
- Natriumpersulfat
- Natriumhydroxid
- Natriumcarbonat
- Lötstopplaminat

Es ist auch ratsam, über die folgenden Geräte zu verfügen.

- UV Belichtungsgerät
- Ätzküvette mit Aquarienpumpe und Aquarienheizstab
- Laminiergerät
- Bohrschleifer mit Bohrstände
- Hartmetallbohrer

4.3.1 Layoutentwurf

Das Layout wurde mit der Software EAGLE der Firma CADsoft erstellt. Die Schaltung ist in 2 separate Platinen aufgeteilt, auf der ersten, dem »Control-Board«, befindet sich die Auswerteschaltung, die Regelung der Heizung bzw. Kühlung sowie die Spannungsversorgung. Auf der zweiten Platine, dem »Sensor Board«, befindet sich lediglich der Radioaktivitätssensor mit seiner üblichen Filterbeschaltung, welche laut Datenblatt [12] empfohlen wird.

Hauptplatine

Tabelle 4.3.1: FalloutMeter Stückliste

Part	Value	Package	Library
C1	100 <i>nF</i>	C1206	rcl
C2	47 <i>nF</i>	C1206	rcl
C3	100 <i>nF</i>	C1206	rcl
C4	100 <i>nF</i>	C1206	rcl
C5	100 <i>nF</i>	C1206	rcl
C6	100 <i>nF</i>	C1206	rcl
C7	100 <i>nF</i>	C1206	rcl
C8	4.7 μ F	C1206	rcl
C9	4.7 μ F	C1206	rcl
C10	10 μ F	C1206	rcl
C11	4.7 μ F	C1206	rcl
C12	4.7 μ F	C1206	rcl
C13	10 μ F	C1206	rcl
C14	100 <i>nF</i>	C1206	rcl
C15	100 <i>nF</i>	C1206	rcl
C17	100 <i>nF</i>	C1206	rcl
C20	100 <i>nF</i>	C1206	rcl
C_BOOT	100 <i>nF</i>	C1206	rcl
C_IN	4,7 μ F	C1206	rcl
C_LOAD	15 <i>pF</i>	C1206	rcl
C_OUT	180 μ F	C1206	rcl

Tabelle 4.3.1: FalloutMeter Stückliste

Part	Value	Package	Library
C_SS	22 <i>nF</i>	C1206	rcl
D1	1N4148	DIODE_SMB	eagle-ltspice
D2	LD117ADT-TR	DPACK	schottky_diode_dpak
D3	BAS40	SOT23	diode
D4	BAS40	SOT23	diode
F1	0,6 A/T	SH22,5A	fuse
IC1	MEGA328P-AU	TQFP32-08	avr-6
IC2	LMR-14050	HSOIC-8	LMR14050
IC3	IR2104S	IR2104S	ir-2104
IC4	IR2104S	IR2104S	ir-2104
L1	8,2 μH	DO3316P	inductor-coilcraft
L2	330 μH	SFT1040	inductors
L3	330 μH	SFT1040	inductors
LED1	KNX-LED	CHIPLED_1206	led
Q1	N-MOSFET	SOT223	transistor-power
Q2	N-MOSFET	SOT223	transistor-power
Q3	N-MOSFET	SOT223	transistor-power
Q4	N-MOSFET	SOT223	transistor-power
QG1	7,3728 <i>MHz</i>	SM77H	crystal
R1	10 <i>k</i> Ω	M1206	resistor
R2	4,7 <i>k</i> Ω	M1206	resistor
R3	4,7 <i>k</i> Ω	M1206	resistor
R4	VARISTORCN1206	CT/CN1206	varistor
R5	22 Ω	M1206	resistor
R6	22 Ω	M1206	resistor
R7	100 <i>k</i> Ω	M1206	resistor
R8	22 Ω	M1206	resistor
R9	22 Ω	M1206	resistor
R100	0 Ω	R1206	resistor

Tabelle 4.3.1: FalloutMeter Stückliste

Part	Value	Package	Library
R_FBB	17,8 k Ω	R1206	rcl
R_FBT	100 k Ω	R1206	rcl
R_LOAD	10 k Ω	M1206	resistor
R_T	84,5 k Ω	R1206	rcl
S1	KNX-Button	SWITCH-SMD	SparkFun-Retired
SV1	AVRISPMK2	MA03-2	con-1stb
U2	BAOS-820	BAOS-820	weinzierl-baos
U3	Aosong AM2321	AOSONG_AM2321	Aosong_AM2321
X1	MKDSN1,5/3-5,08	MKDSN1,5/3-5,08	con-phoenix-508
X2	MKDSN1,5/2-5,08	MKDSN1,5/2-5,08	con-phoenix-508
X3	243-211	243-211	wago-243-1
X4	MKDSN1,5/2-5,08	MKDSN1,5/2-5,08	con-phoenix-508

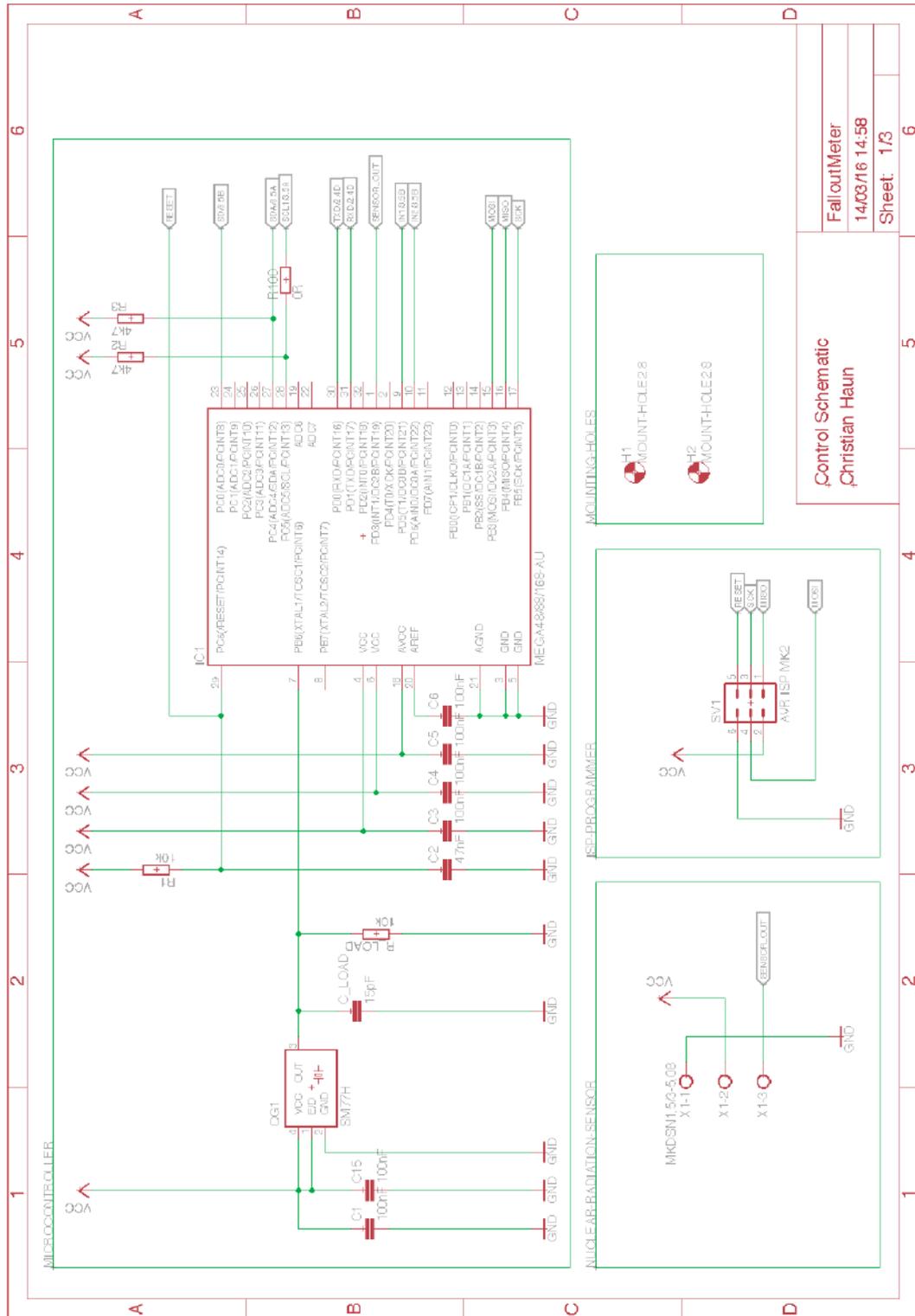


Abbildung 4.3.1: FalloutMeter Schaltplan Seite 1

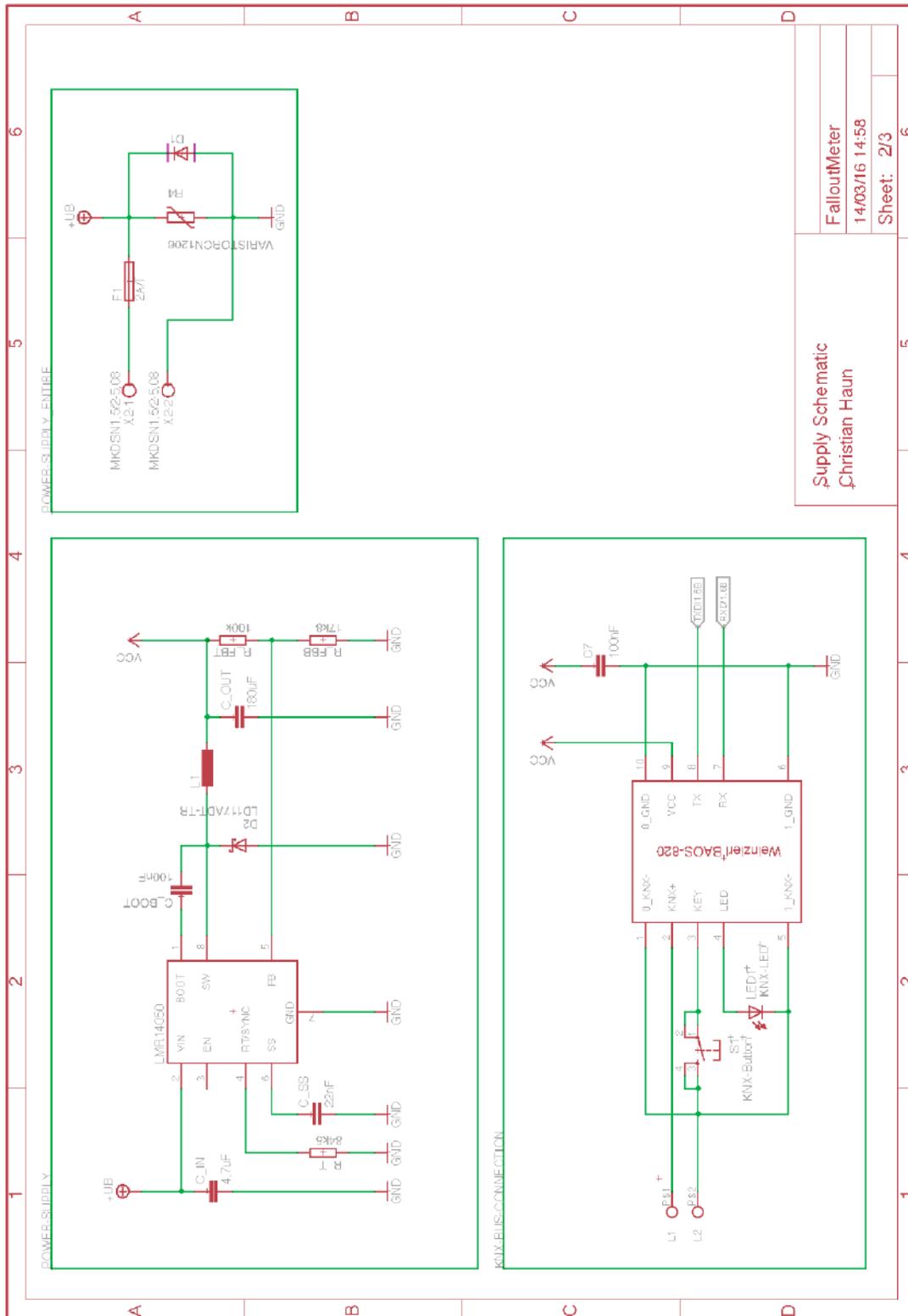
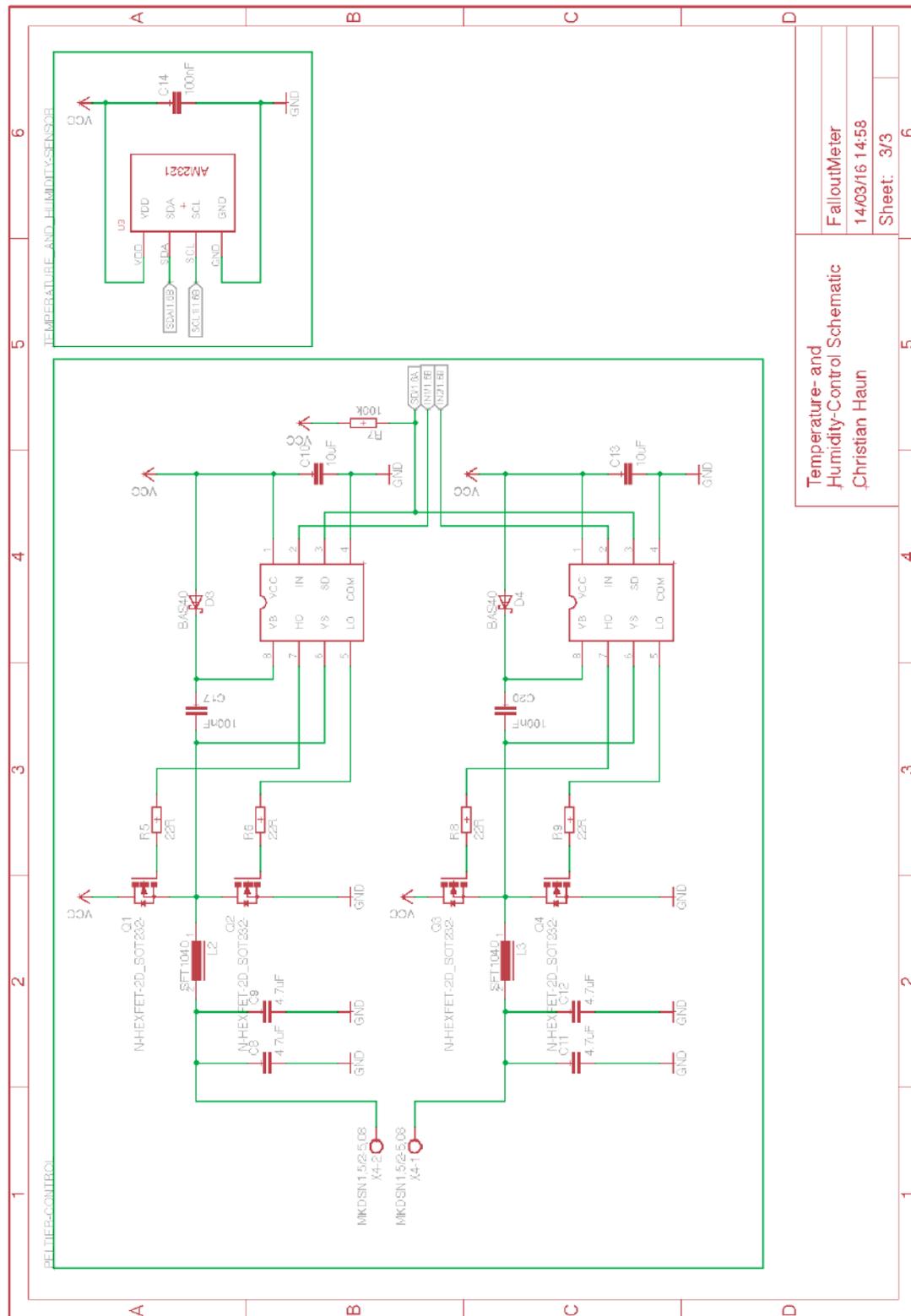


Abbildung 4.3.2: FalloutMeter Schaltplan Seite 2



Temperature- and Humidity-Control Schematic		FalloutMeter	
Christian Haun		14/03/16 14:58	
Sheet: 3/3			

Abbildung 4.3.3: FalloutMeter Schaltplan Seite 3

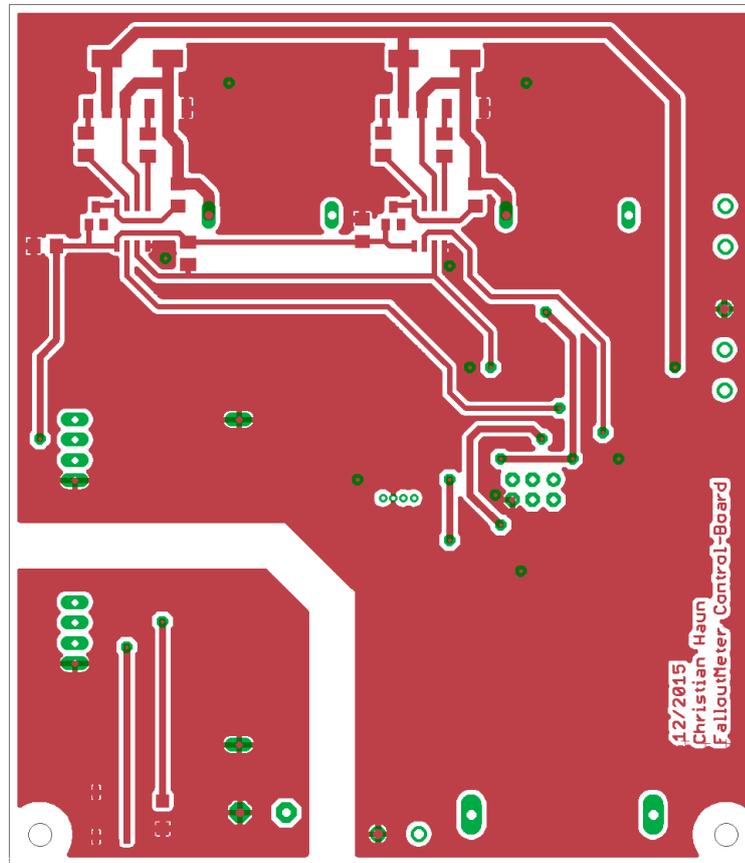


Abbildung 4.3.4: FalloutMeter Layout Top

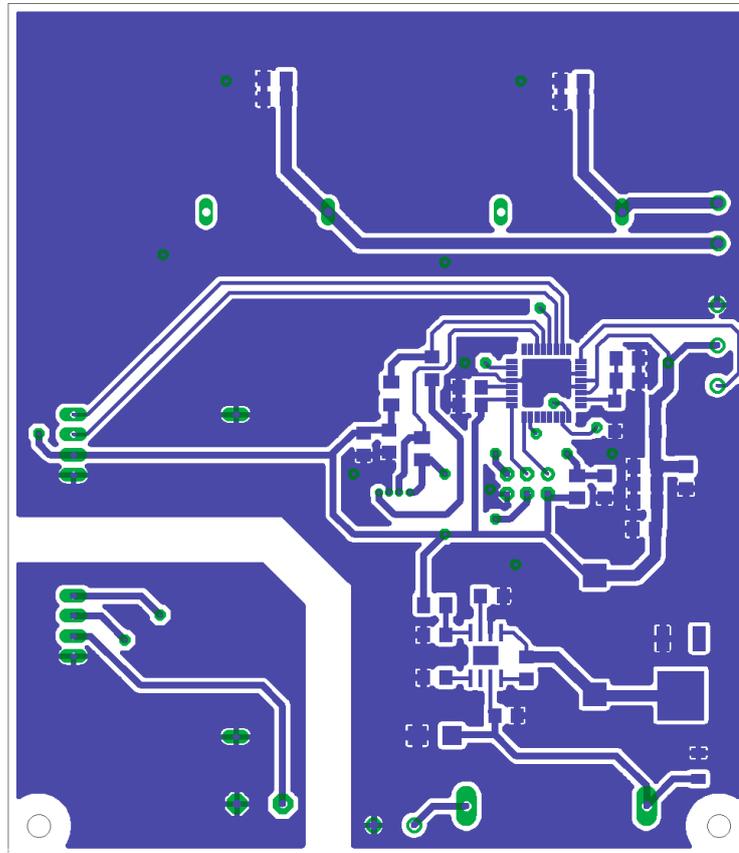


Abbildung 4.3.5: FalloutMeter Layout Bottom

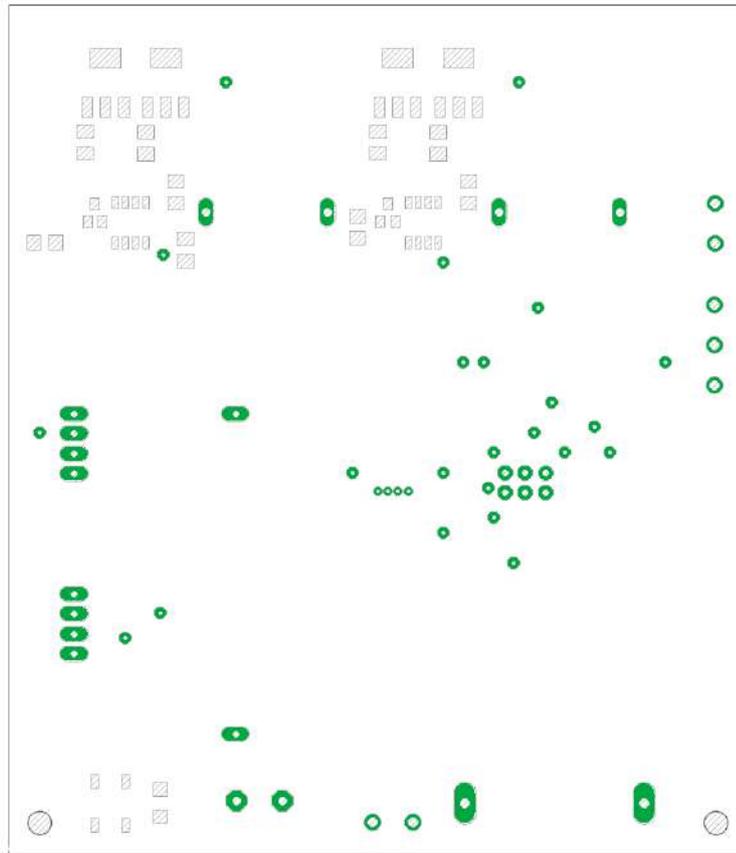


Abbildung 4.3.6: FalloutMeter Lötstopmmaske Top

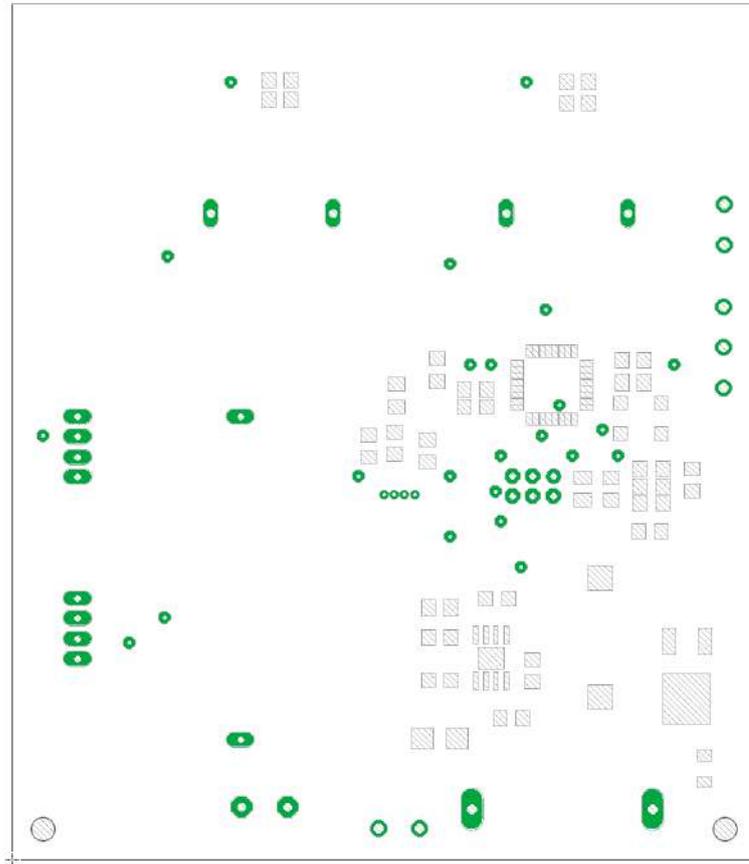


Abbildung 4.3.7: FalloutMeter Lötstopmmaske Bottom

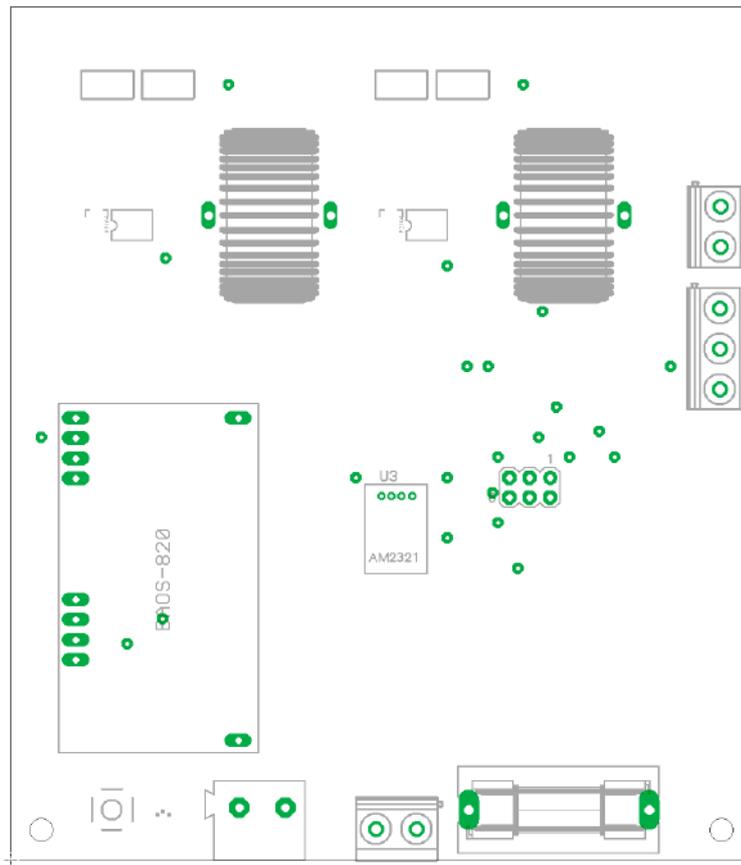


Abbildung 4.3.8: FalloutMeter Bestückungsplan Top

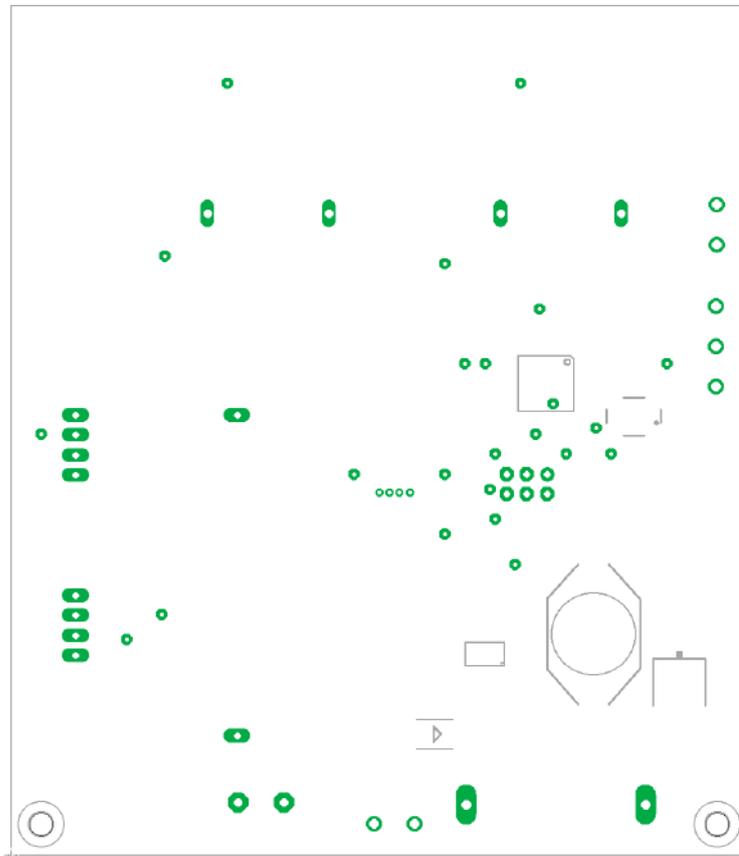


Abbildung 4.3.9: FalloutMeter Bestückungsplan Bottom

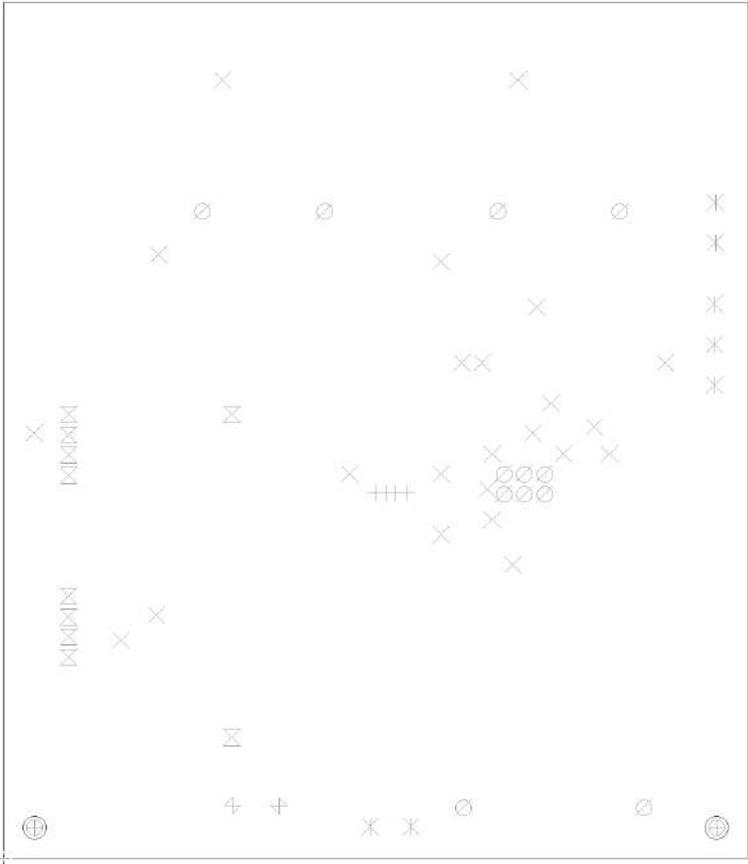
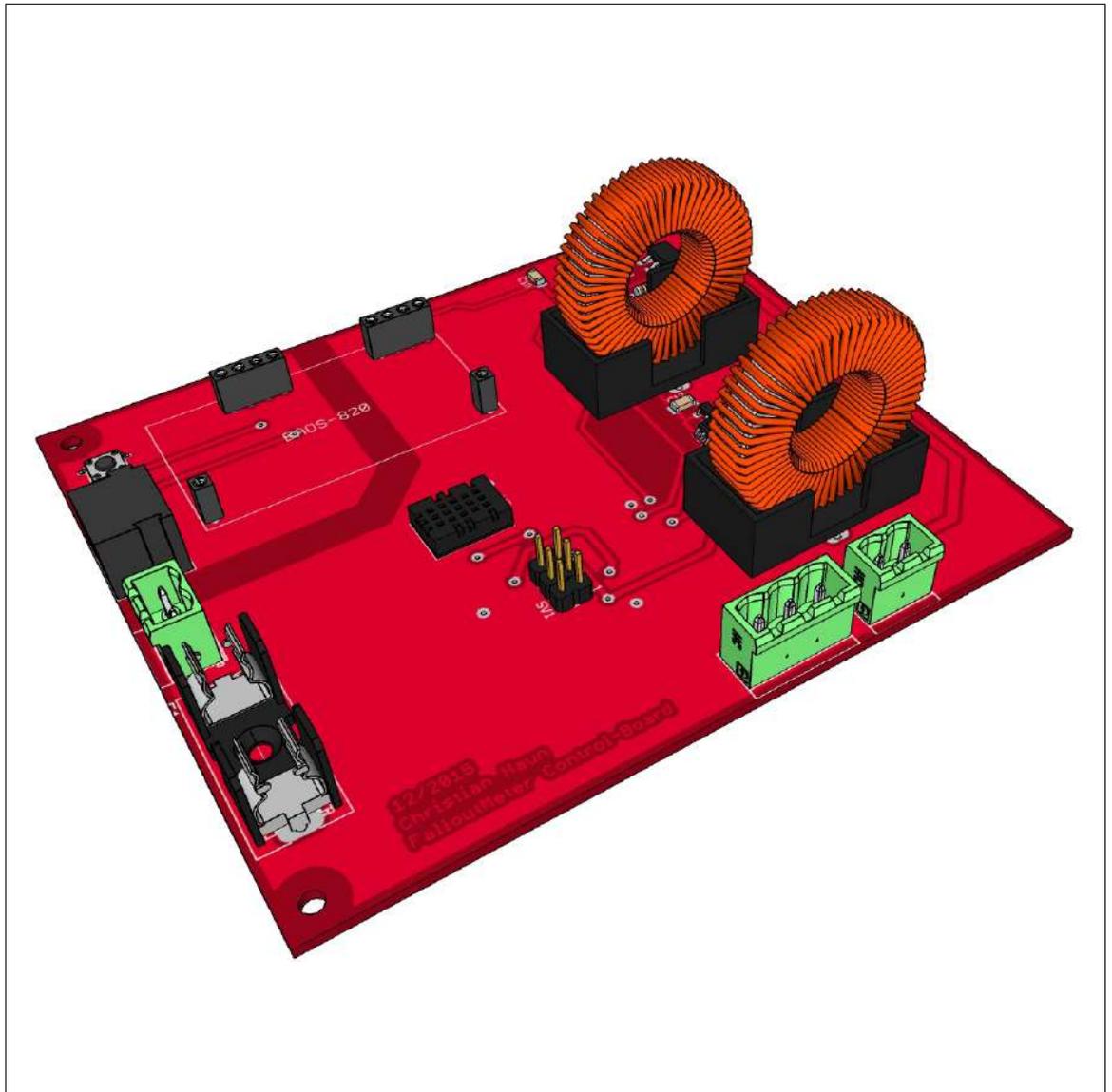


Abbildung 4.3.10: FalloutMeter Bohrplan

3D-Modell



Sensorplatine**Tabelle 4.3.2:** FalloutMeter Sensor Stückliste

Part	Value	Package	Library
C1	100 μF	153CLV-0405	rcl
R1	150 Ω	M1206	resistor
U\$1	RD2014	RD2014	Teviso_RD2014
X1	MKDSN1,5/3-5,08	MKDSN1,5/3-5,08	con-phoenix-508

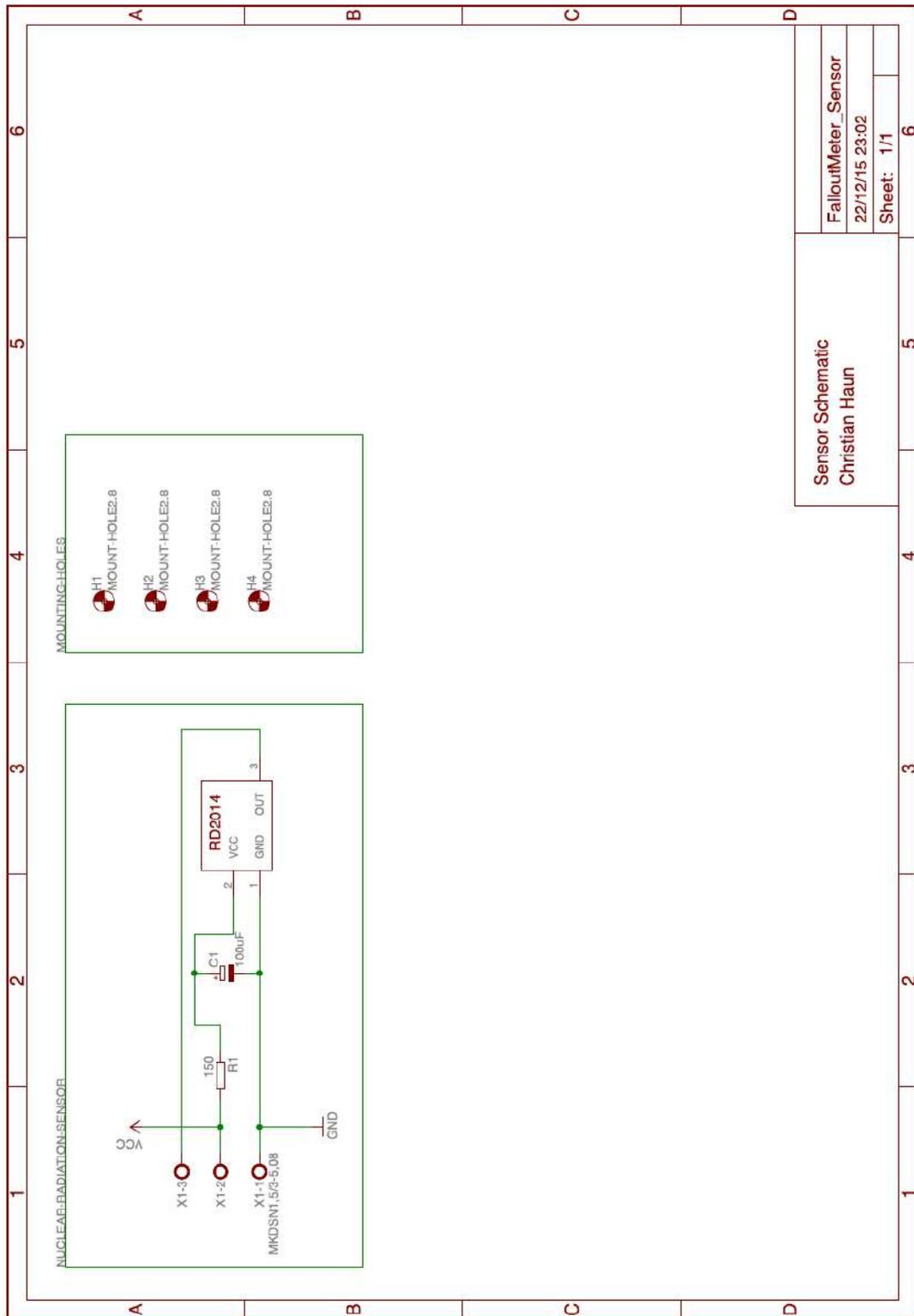


Abbildung 4.3.11: FalloutMeter Sensor Schaltplan

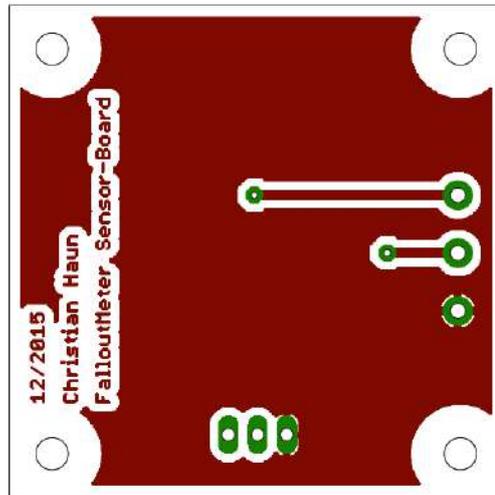


Abbildung 4.3.12: FalloutMeter Sensor Layout Top

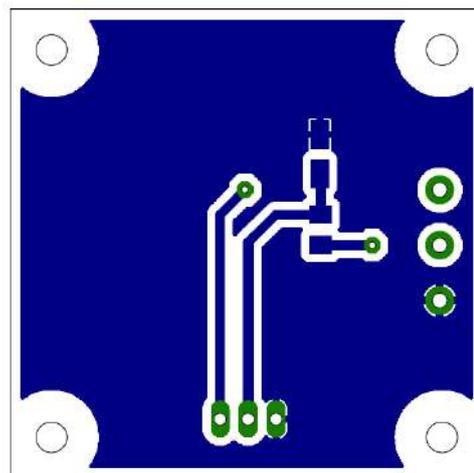


Abbildung 4.3.13: FalloutMeter Sensor Layout Bottom

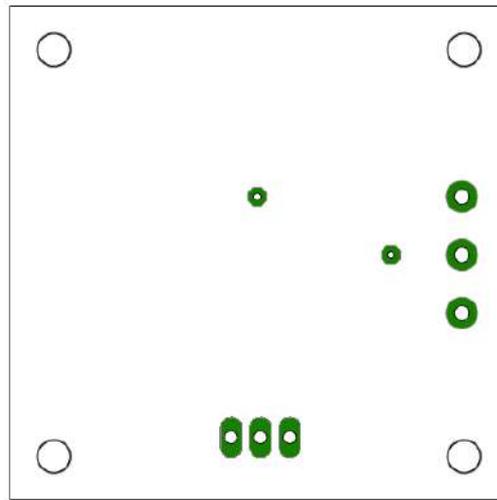


Abbildung 4.3.14: FalloutMeter Sensor Lötstopmmaske Top

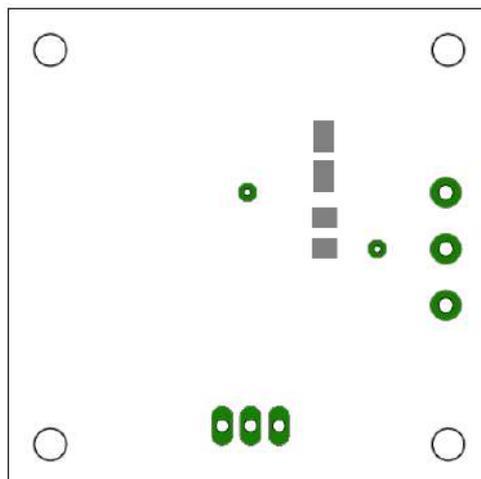


Abbildung 4.3.15: FalloutMeter Sensor Lötstopmmaske Bottom

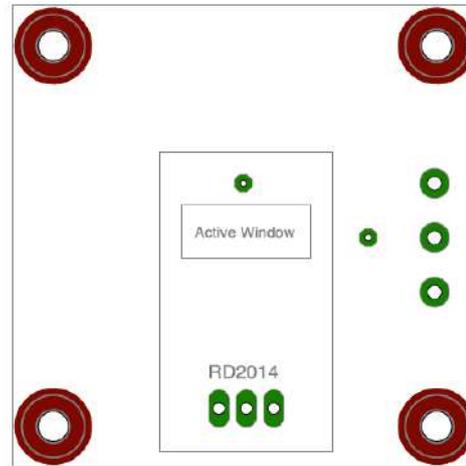


Abbildung 4.3.16: FalloutMeter Sensor Bestückungsplan Top

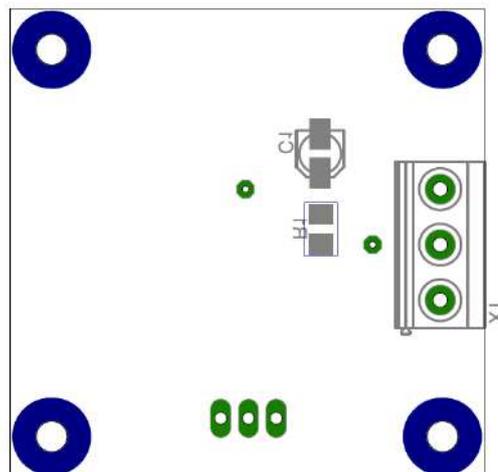


Abbildung 4.3.17: FalloutMeter Sensor Bestückungsplan Bottom

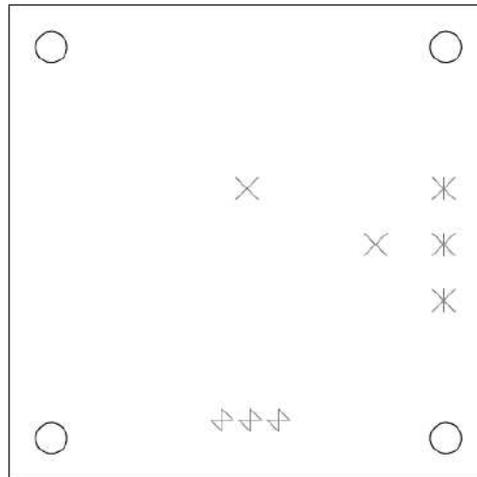


Abbildung 4.3.18: FalloutMeter Sensor Bohrplan

3D-Modell



4.3.2 Belichten

Zuerst wird das Layout auf eine transparente Folie aufgedruckt. Nun wird die Platine unter UV-Licht in einem Belichtungsgerät für 2:30 min belichtet. In Abbildung 4.3.19 ist das verwendete Belichtungsgerät abgebildet.



Abbildung 4.3.19: Platinen belichten

4.3.3 Entwickeln

Nach dem Belichten wird in einem Entwicklungsbad in einer Mischung aus 1 l Wasser und 20 g Natriumhydroxid die Platine entwickelt. Dies sollte am besten unter Zufuhr von Sauerstoff geschehen, dazu wird die Platine öfters aus dem Entwicklungsbad gezogen und geschwenkt. Ist das Layout gut auf der belichteten Seite der Platine erkennbar, ist der Entwicklungsprozess abgeschlossen. Danach wird die Platine mit Wasser von der Entwicklerflüssigkeit befreit. Abbildung 4.3.20 zeigt das Entwicklerbad.



Abbildung 4.3.20: Platinen entwickeln

4.3.4 Ätzen

Da die Platine nun fertig entwickelt ist, kann das Ätzen beginnen. Hierfür wird ein Ätzbad aus 1,5 l Wasser und 300 g Natriumpersulfat vorbereitet. Der Ätzvorgang geht

bei einer Temperatur von $45\text{ }^{\circ}\text{C}$ am schnellsten vorstatten, dies kann man beispielsweise mit einem Aquarienheizstab realisieren.

Weiters sollte man auf eine geeignete Sauerstoffzufuhr achten. Dies wird mit einem Luftverteilerschlauch am Boden der Ätzküvette, welcher mit einer Aquariumpumpe verbunden ist, sichergestellt. Die Platine wird nun so lange im Ätzbad belassen, bis sich sämtliches belichtetes Kupfer abgelöst hat.

Nun wird die Platine aus dem Ätzbad genommen und gut mit Wasser abgespült. Danach wird die Leiterplatte mit Ethanol oder Dimethylketon gereinigt, um den restlichen Fotolack zu entfernen. Abbildung 4.3.21 zeigt den Ätzvorgang.

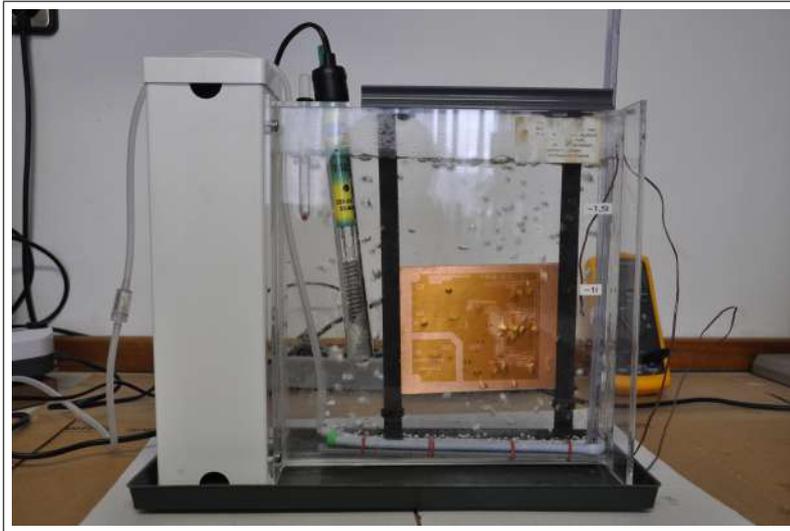


Abbildung 4.3.21: Platinen ätzen

4.3.5 Aufbringen der Lötstopmmaske

Dieser Schritt ist nicht zwingend notwendig, jedoch ist die Lötstopmmaske sehr empfehlenswert, da sie das Löten stark erleichtert und die Platine vor Oxidation schützt.

Hierfür wird das Lötstopplaminat mit einem Laminiergerät zuerst kalt auflaminiert, so kann man es im Fehlerfall noch abziehen. Danach wird heiß auflaminiert. Nun wird die Lötstopmmaske auf transparenter Folie aufgedruckt und die Platine wird mit dieser im Belichtungsgerät unter UV-Licht 50 s lang belichtet.

Danach wird eine Lösung aus 1 l Wasser und 10 g Natriumcarbonat angerührt und die Platine dort so lange belassen, bis sich die unbelichteten Stellen abgelöst haben. Dieser Vorgang kann durch bearbeiten der Platinenoberfläche mit einem Pinsel beschleunigt werden. Hierbei ist auf die zusätzliche Schutzfolie des Lötstopplaminats zu achten, diese muss natürlich vorher abgezogen werden.

Ist dies geschehen, sollte die Platine noch 15 min im Belichtungsgerät belichtet werden und einige Tage lang unter Tageslicht belassen werden, wird darauf verzichtet, kann es geschehen, dass sich beim Löten einige Teile der Lötstopmmaske ablösen.

Auf Abbildung 4.3.22 sehen Sie die Platine mit aufgebrachtener Lötstopmmaske.

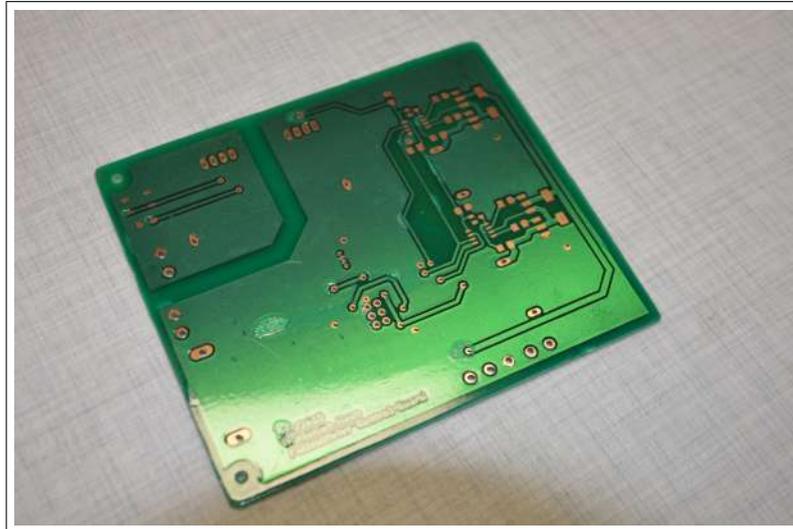


Abbildung 4.3.22: Lötstopmmaske aufbringen

4.3.6 Bohren

Zum Bohren der Platinen wird am besten ein Bohrschleifer mit dazugehörigem Bohrstander verwendet, diesen gibt es beispielsweise von der Firma Proxxon. Weiters wurden die Hartmetallbohrer der Firma Bungard verwendet, da diese in den benötigten Größen leicht erhältlich sind.

In Abbildung 4.3.23 sehen Sie die Legende zum Bohrplan der Platinen.

LAYER-STACK	Sym	Nº	Mils	MM	Qty	Plated
01-16	+	1	20	0.50	4	YES
01-20	×	2	24	0.60	23	YES
	□	3	35	0.90	10	YES
	◇	4	40	1.02	6	YES
	⊗	5	43	1.10	1	YES
	⊗	6	44	1.12	4	YES
	+	7	47	1.20	1	YES
	+	8	51	1.30	7	YES
	*	9	52	1.32	2	YES
	*	10	110	2.80	2	NOT

Abbildung 4.3.23: Bohrplan Legende

4.3.7 Löten

Zuletzt wird die Platine noch per Hand oder im Reflow-Ofen bestückt. Auf Abbildung 4.3.24 sehen sie nun die fertig bestückte Platine.

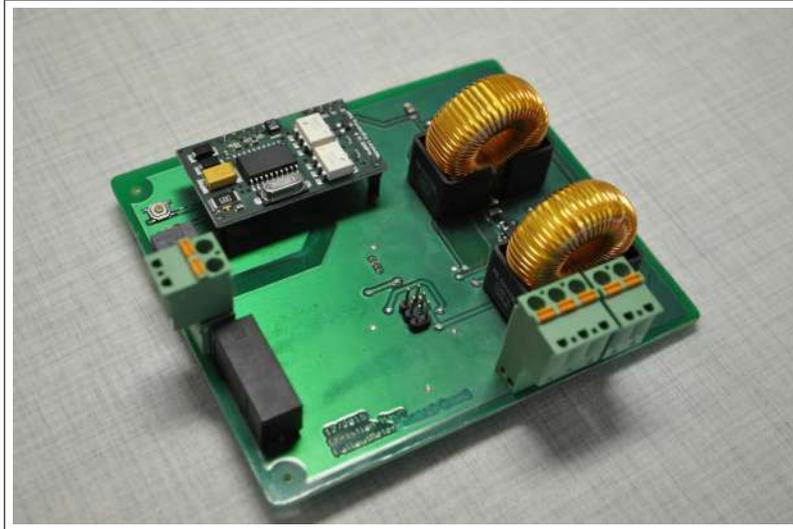


Abbildung 4.3.24: Platinen löten

5 Mechanischer Aufbau

5.1 Die ersten Modellentwürfe

Für die Modellierung des Gehäuses für die Platine und sämtliche Heizungs- und Kühlkörper standen verschiedene CAD-Programme zur Verfügung. Das Modell sollte als Hilfe zur vereinfachten Darstellung und Veranschaulichung für den tatsächlichen Bau dienen.

5.1.1 Solid Edge ST4

Zu Beginn wurde das Programm Solid Edge ST4 verwendet, da für diese Software einiges an Erfahrung in der Schule gesammelt wurde. Nach einigen Stunden Arbeit musste jedoch festgestellt werden das Solid Edge kein geeignetes Programm für die Gehäuseentwicklung war.

5.1.2 Autodesk Inventor 2016

Letztendlich wurde das Modell mit der Software Inventor Professional 2016 der Firma Autodesk erstellt. Dieses Programm bietet eine Vielzahl an Funktionen. Im Gegensatz zu Solid Edge ST4 ist die Blechbearbeitung bei Autodesk Inventor 2016 einfacher zu bedienen. Abbildung 5.1.1 zeigt das 3D-Modell des Gehäuses.

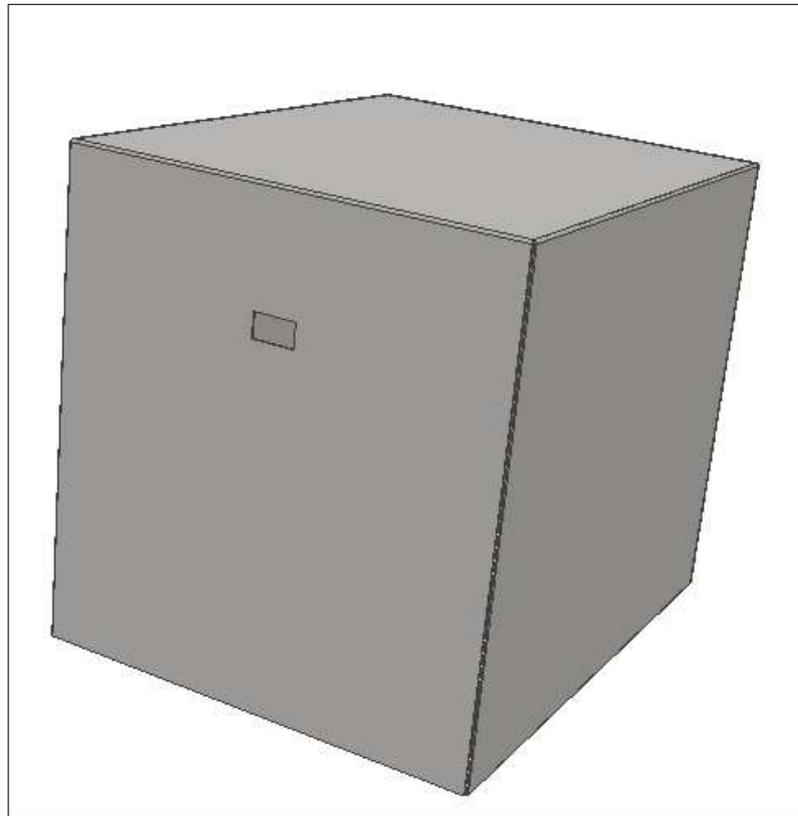


Abbildung 5.1.1: 3D-Modell Gehäuse

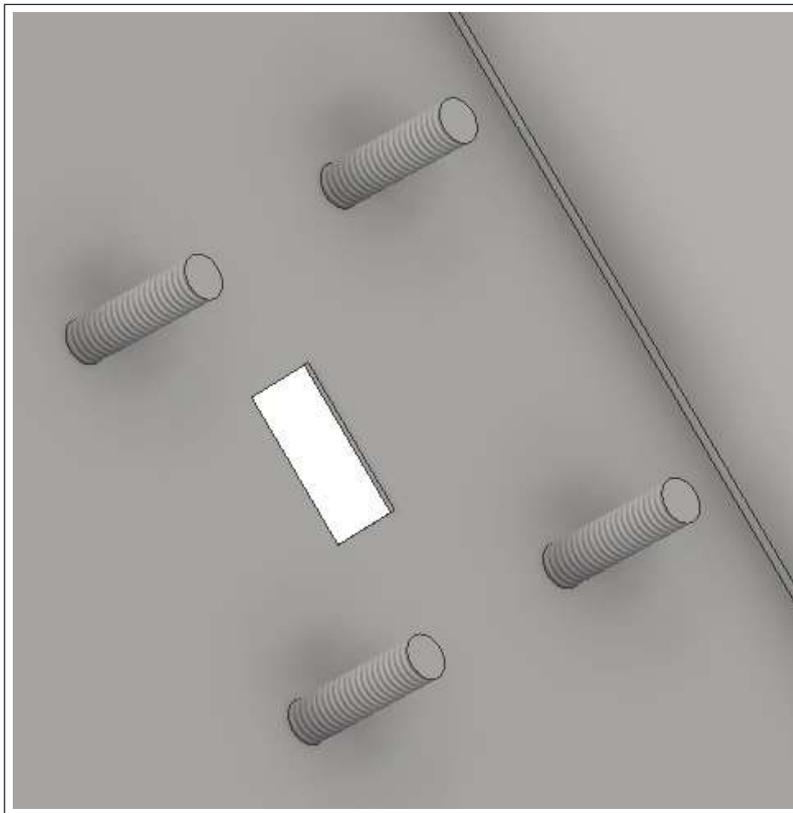


Abbildung 5.1.2: 3D Modell der Gewindestangen für die Sensorplatine

5.2 Gehäusebau

Um das Gehäuse möglichst langlebig und stabil zu halten wurde als Material Baustahl verwendet. Eine weitere positive Eigenschaft ist, dass Baustahl sehr gut verarbeitbar ist. Das Ziel war es, ein Gehäuse zu entwickeln, welches von allen Seiten, ausgenommen auf der Unterseite, schraubenlos ist.

5.2.1 Gehäuseoberteil

Damit das Gehäuse stabil genug ist, wurde eine Blechdicke von $1,5\text{ mm}$ verwendet. Um ausreichend Platz für die Platine im Innenraum vom Gehäuse zu schaffen, wurde eine Länge von 150 mm , eine Breite von 200 mm und eine Höhe von 160 mm gewählt.

Blechausschnitt

Um die Strahlung mit dem Mikrocontroller auszuwerten, darf das Sensorfeld nicht vom Blech beeinträchtigt werden. Aus diesem Grund wurde ein Blechausschnitt auf der Vorderseite des Gehäuses durchgeführt. Der Ausschnitt hat eine Länge von 14 mm und einer Breite von 8 mm . Diese entsprechen in etwa den Bemaßungen des Sensorfeldes. Damit das Sensorfeld vor Eindringen von Regenwasser oder Schnee geschützt ist, wurde eine kleine Plexiglasplatte hinter dem Blechausschnitt befestigt, welche mit Dichtmasse abgedichtet wurde. Abbildung 5.2.1 zeigt den Ausschnitt für das Sensorfeld.



Abbildung 5.2.1: Blechausschnitt

Gewindestangen für die Sensorplatine

Damit eine sichere Anschraubung für die Sensorplatine gewährleistet ist, wurden rund um den Blechausschnitt vier Gewindestangen mit einer Länge von 15 mm und einem Durchmesser von 4 mm auf der Innenseite vom Gehäuse hinzugefügt. Die Sensorplatine wurde mit M4 Muttern befestigt. Um die Sensorplatine beim Anschrauben nicht zu beschädigen, wurden M4 Beilagscheiben benutzt. Abbildung 5.2.2 zeigt die Gewindestangen der Sensorplatine.



Abbildung 5.2.2: Gewindestangen für die Sensorplatine

5.2.2 Gehäuseunterteil

Die 2 mm dicke Blechplatte aus Baustahl hat eine Länge von 150 mm und eine Breite von 200 mm . Sie dient als Grundplatte des Gehäuses auf welchem sämtliche Elemente befestigt werden.

Halterung für die Hauptplatine

Die metallische Fassung sorgt für eine stabile Befestigung der Hauptplatine. Die Platine wird sozusagen zwischen den Schlitzen »gesteckt«.



Abbildung 5.2.3: Halterung ohne Platine

Heiz- und Kühlkörper

Ein LED-Kühlkörper der Marke »Fischer Elektronik« und das Peltier-Element sorgen für die ideale Temperatur im Innenraum des Gehäuses. Der Kühlkörper besteht aus Aluminium und hat einen Durchmesser von 60 mm bei einer Höhe von 50 mm . Das Peltier Element wird mittels Wärmeleitkleber am Gehäuseunterteil befestigt. Um die Temperatur optimal leiten zu können wird der Kühlkörper auf das Peltier Element geklebt. Zwei M3 Gewindestangen mit Anschraubung sorgen für eine zusätzliche Befestigung der Elemente.

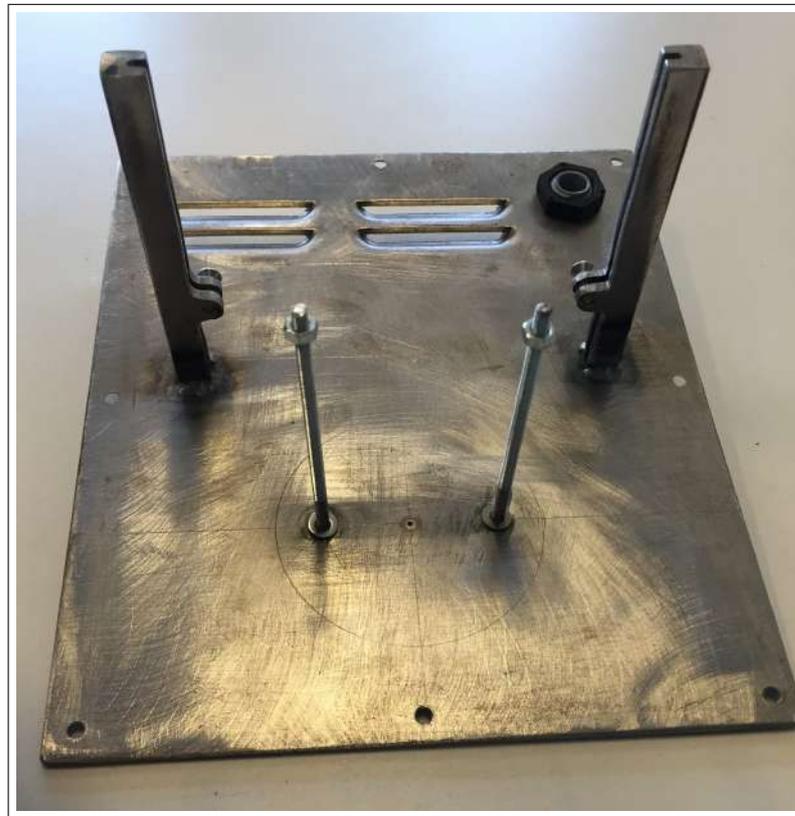


Abbildung 5.2.4: Befestigung des Kühlkörpers und des Peltier Elementes

Bohrung für KNX-Buskabel

Um das Kabel fest und dicht mit dem Gehäuse zu verbinden, wurde eine M12 Kabelverschraubung aus Metall verwendet. Sie dient ebenfalls als Schutz vor Feuchtigkeit, Staub und mechanischen Einflüssen. Damit die Kabelverschraubung ordnungsgemäß am Unterteil vom Gehäuse angesetzt werden kann, wurde ein Loch mit einem Durchmesser von 12 mm gebohrt. Mit Hilfe einer Gegenmutter wurde die Kabelverschraubung befestigt.

Lüftungsschlitze

Damit eine optimale Luftzirkulation gegeben ist, wurden Lüftungsschlitze am Unterteil vom Gehäuse eingepresst. Diese sollen auch eine Bildung von Kondenswasser im Gehäuseinneren verhindern.



Abbildung 5.2.5: Lüftungsschlitze

5.2.3 Vollendung des mechanischen Aufbaus

Für das Zusammenbauen von Ober- und Unterteil wurde ein Rahmen am unteren Ende vom Gehäuseoberteil angeschweißt. Der Rahmen verfügt über acht Gewindebohrungen für Kreuzschlitzschrauben welche dazu dienen, dass das Gehäuseunterteil mit dem Gehäuseoberteil fest verschraubt wird. Eine Gummidichtung zwischen Gehäuseober- und Gehäuseunterteil verhindert ungewolltes Eindringen von Staub und Flüssigkeiten. Abbildung 5.2.6 zeigt den angeschweißten Rahmen und die acht Gewindebohrungen.



Abbildung 5.2.6: Rahmen mit Gewindebohrungen

Halterung für das Gehäuse

Um ein möglichst optimales Messergebnis zu erreichen, muss das Gehäuse im Freien angebracht werden. In diesem Fall wird es an einem Rohrgelände mit einem Durchmesser von 40 mm befestigt. Daher wurden mittig auf der Rückseite vom Gehäuse zwei Rohrhalterungen angeschweißt.

Verzinken

Da das Gehäuse im Freien verbaut wird, muss es vor Korrosion geschützt werden. Aus diesem Grund wurde es verzinkt. Das Verzinken vom Metallgehäuse erhöht die Lebensdauer und sorgt für ein hohes Maß an Qualität.

Folgende Schritte beschreiben den Ablauf der galvanischen Verzinkung:

1. Gehäuse alkalisch entfetten und spülen
2. in Salzsäure beizen um Rost zu entfernen

3. elektrolytisch entfetten und spülen
4. Aufhellen in verdünnter Salzsäure
5. Das Verzinken erfolgt 40 *min* lang im sauren Glanzzinkbad bei $2 \frac{A}{dm^2}$.
6. Anschließend wird das Gehäuse einer 25-minütigen Chromatierung unterzogen.
7. spülen und trocknen

Abbildung 5.2.7 zeigt das verzinkte Gehäuseunterteil.

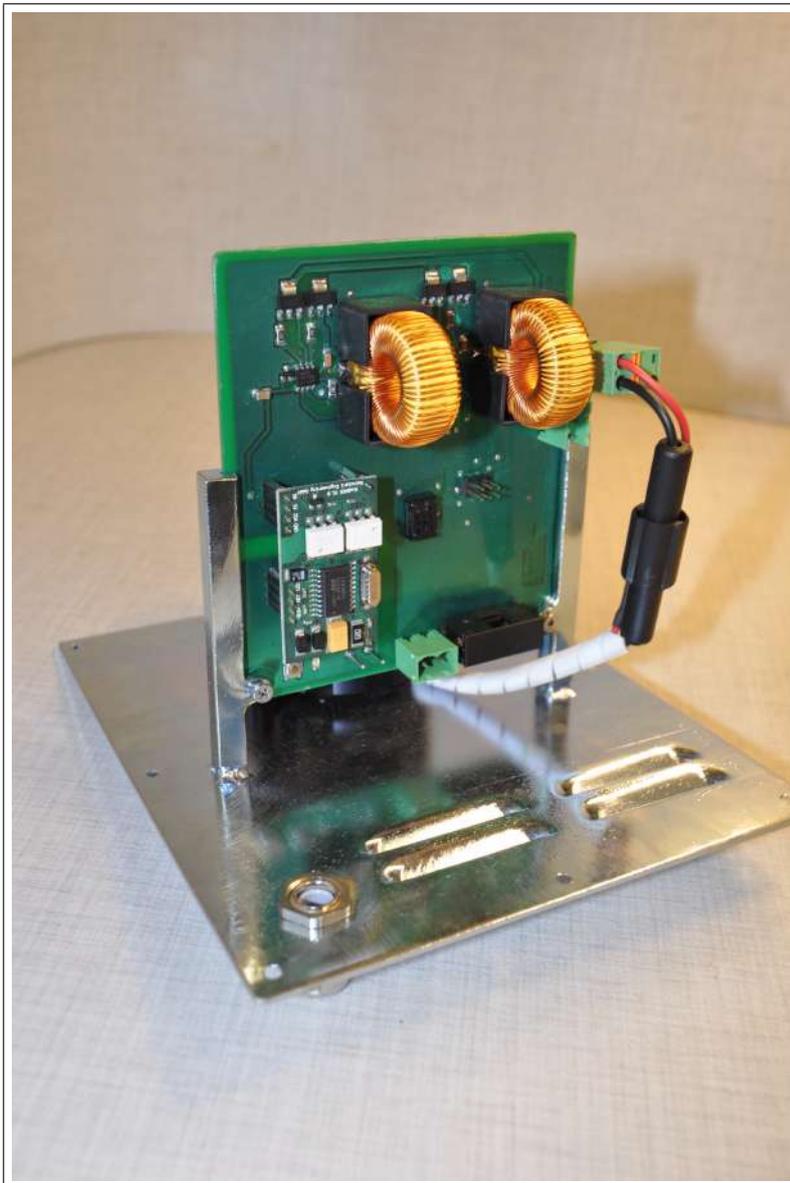


Abbildung 5.2.7: verzinktes Gehäuseunterteil

In Abbildung 5.2.8 sehen Sie die fertige Befestigung für den Kühlkörper.

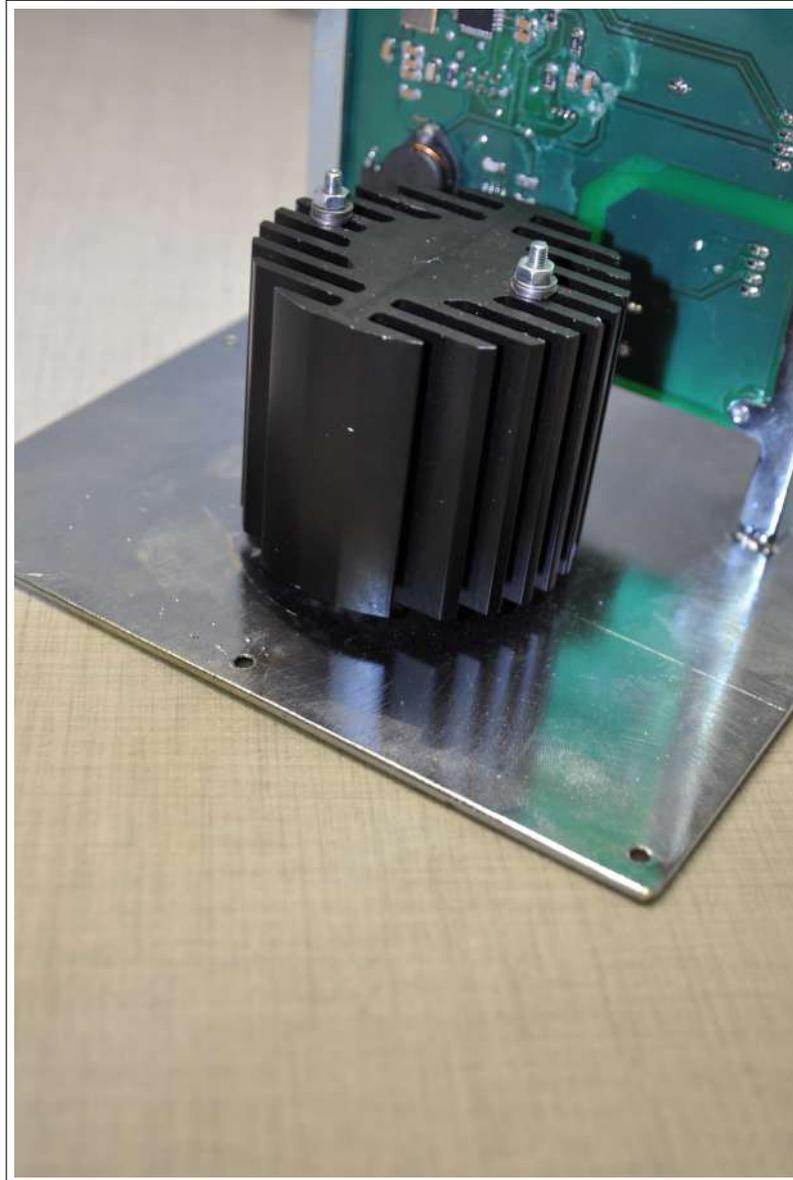


Abbildung 5.2.8: Befestigung des Kühlkörpers

Abbildung 5.2.9 zeigt das verzinkte Gehäuseoberteil.

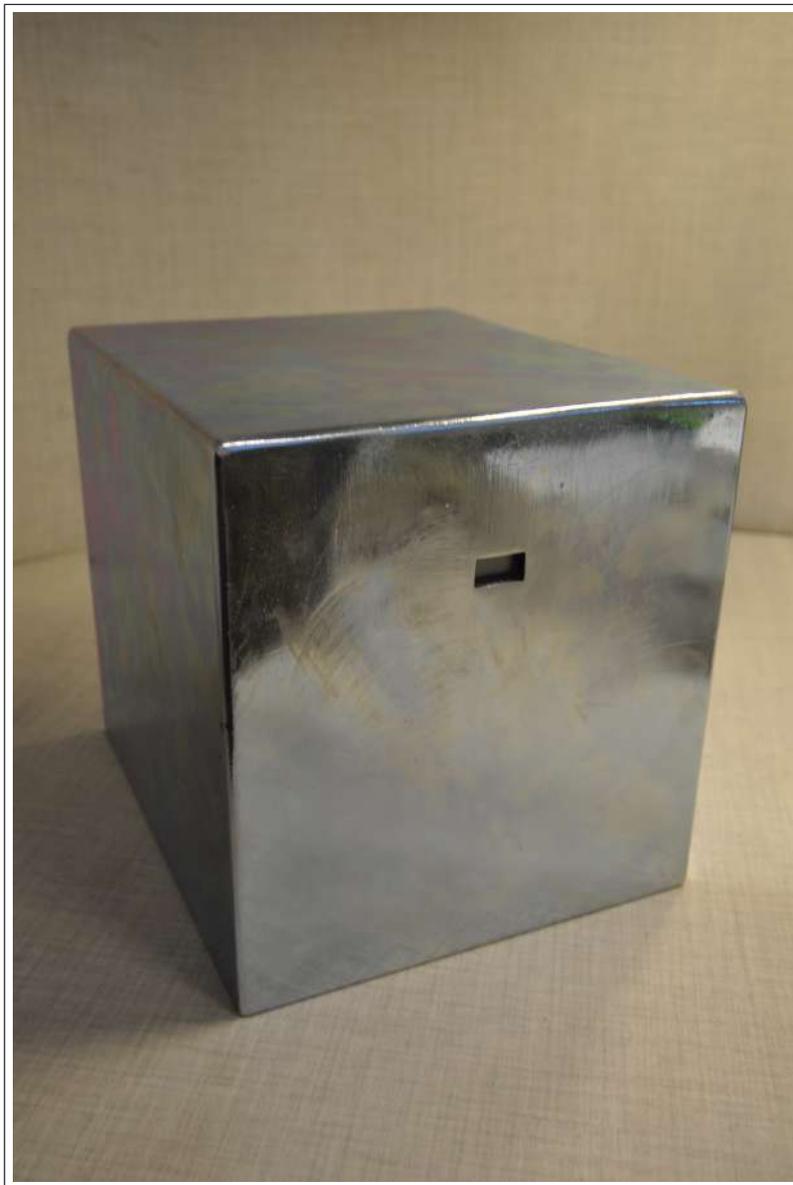


Abbildung 5.2.9: verzinktes Gehäuseoberteil

Abbildung 5.2.10 zeigt die Halterung für den Sensor samt Sensorplatine.



Abbildung 5.2.10: Befestigung des Sensors

In Abbildung 5.2.10 ist die Wandhalterung für das Gehäuse zu sehen. Da es an einer Brüstung befestigt wird, wurde entschieden, die auf der Abbildung zu sehenden metallenen Halbreise am Gehäuse anzubringen, um es später mittels Schlauchschellen, wie sie auch im Sanitärbereich verwendet werden, an jener Brüstung zu befestigen. Dies hat den Vorteil, dass das Gehäuse auch an verschiedenen anderen Orten befestigt werden kann, beispielsweise an einer eigens dafür platzierten Metallstange.



Abbildung 5.2.11: Wandhalterung des Gehäuses

6 Softwareentwicklung der Messschaltung

6.1 Software für den ATMEL ATmega328P

Die Software wurde in C für den Mikrocontroller ATmega328P der Firma Atmel erstellt. Das Programm, das zur KNX-Kommunikation über den UART¹ des Mikrocontroller mit dem Weinzierl BAOS-820-Modul dient, steht auf der Herstellerwebseite zum Download zur Verfügung. [13] Diese Software ist jene für das Weinzierl Development- oder Demo-Board. Da jedoch am Demo-Board ein Atmel ATmega328 verbaut ist, eignet sich diese Software zur weiteren Verwendung.

Der erste Entwurf wurde auf dem Weinzierl Development Board getestet, welches uns freundlicherweise zur Verfügung gestellt wurde.

Wichtig ist, beim Programmieren auf die passende Einstellung der Fuses Rücksicht zu nehmen. Im Falle des ATmega328P müssen diese laut Tabelle 6.1.1 eingestellt werden.

Tabelle 6.1.1: ATmega328P Fuses

ATmega328P Fuses	
Beschreibung	Wert
BODLEVEL	2V7
RSTDISBL	DEACTIVATED
DWEN	DEACTIVATED
SPIEN	ACTIVATED
WDTON	DEACTIVATED
EESAVE	DEACTIVATED
BOOTSZ	2048W_3800
BOOTRST	DEACTIVATED
CKDIV8	DEACTIVATED
CKOUT	DEACTIVATED
SUT_CKSEL	EXTXOSC_8MHZ_XX_16KCK_14CK_65MS
EXTENDED	0xFD
HIGH	0xD9
LOW	0xFF

¹UART Universal Asynchronous Receiver Transmitter

Der Radioaktivitätssensor hat eine TTL-Zählausgang, welcher über einen externen Interrupt in den Mikrocontroller eingelesen wird. Mit den Informationen aus dem Datenblatt ([12]) lässt sich nun der jeweilige Messwert berechnen.

```
1 uSvh = MAXCNT * 60 / (float)dt / CalFactor;
```

Listing 6.1: Berechnung des Messwertes

Hierbei ist $MAXCNT = 10$, $CalFactor = 5.8$ und dt jene Zeitspanne in s , bis $MAXCNT$ erreicht ist.

Hat der Mikrocontroller 10 Sensorimpulse ($MAXCNT$) gezählt, wird aus der bis dahin verstrichenen Zeit ein Messwert errechnet.

Die Software ist so programmiert, dass sie zyklisch und/oder auf Anfrage den Messwert an den KNX-Bus sendet. Desweiteren verfügt das Messgerät über eine zyklische Aktivitätsanzeige, welche die Funktionsweise garantiert. Sollte dieses Signal ausbleiben, kann die GSM-Schnittstelle eine Störmeldung abgeben.

6.1.1 Bestimmen der Regelparameter der Gehäusetemperierung

Zur Regelung des Peltier-Elements wurde ein digitaler PI-Regler gewählt, dessen Parameter mittels Einstellregeln nach Chien, Hrones und Reswick ermittelt wurden, auch »Wendetangentenverfahren« genannt.

Im Programm wurden 2 separate Regler erstellt, einen für die obere Temperaturgrenze, ein anderer für die untere. Aus Energiespargründen bleiben beide Regler im Programm gesperrt, es sei denn die Temperatur im Gehäuse nähert sich einem der beiden Grenzwerte. In diesem Fall wird der jeweilige Regler aktiviert und regelt die Temperatur auf einen der beiden Sollwerte.

Auf Abbildung 6.1.1 sehen sie die aufgenommene Sprungantwort der Regelstrecke.

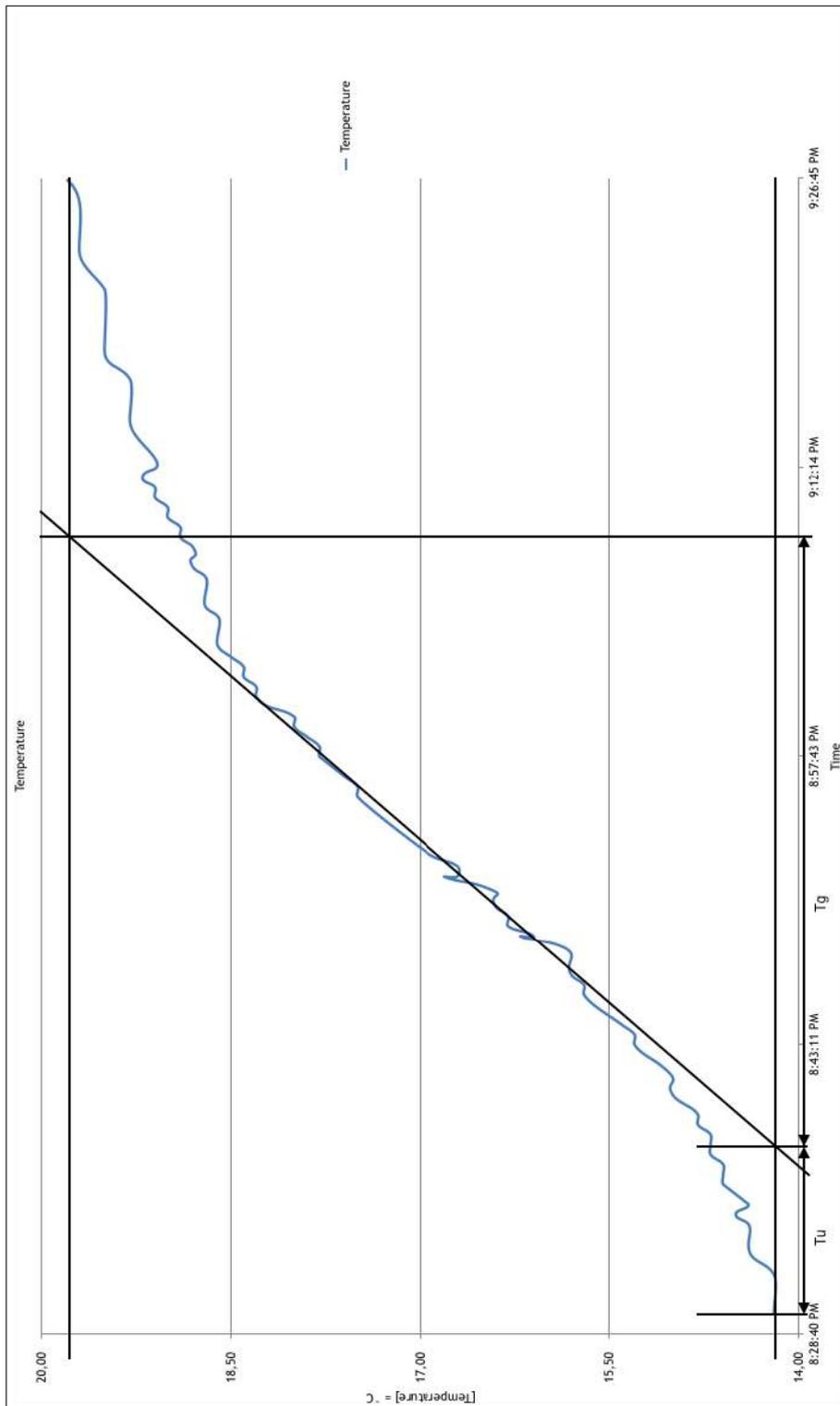


Abbildung 6.1.1: Peltier-Element Sprungantwort

Der Regler soll auf eine Optimierung des Führungsverhaltens ausgelegt sein. In den Gleichungen 6.1 und 6.2 werden nun die Parameter des PI-Reglers berechnet. Die aus der Sprungantwort ermittelten Werte sind: $T_u = 504,26 s$, $T_g = 1833,68 s$ und $K_S = 1$.

$$K_R = 0,35 \cdot \frac{T_g}{T_u \cdot K_S} = 1,27 \quad (6.1)$$

$$T_N = 1,2 \cdot T_g = 2200,42 s \quad (6.2)$$

Mithilfe dieser Temperaturregelung kann die Schaltung auch unter, für unsere Breitengrade, extreme Temperaturen arbeiten.

7 Kommunikation

7.1 KNX

Bei KNX, Nachfolger des bekannten EIB¹, handelt es sich um den weltweiten Standard in der Gebäudeautomatisierung. Er vereint auch die Mechanismen des BatiBus und EHS². Das KNX-Bussystem ist dezentral aufgebaut, sprich jeder Busteilnehmer hat seine eigene Intelligenz. Dies führt dazu, dass im Falle des Ausfalls eines Gerätes die restlichen Geräte ohne Unterbrechung weiterarbeiten können. Die Geräte werden mithilfe der ETS³ programmiert.

7.1.1 Allgemein

Beim KNX System handelt es sich um ein Bussystem für die Gebäudesteuerung. Dies bedeutet, dass alle Geräte das gleiche Übertragungsverfahren benutzen und über eine gemeinsame Busleitung Daten austauschen. Daraus ergeben sich folgende Konsequenzen. [14]

- Der Zugriff auf die Busleitung muss eindeutig geregelt sein (Buszugriffsverfahren).
- Ein großer Anteil der übertragenen Daten sind nicht Nutzdaten (z.B. Licht an/Licht aus), sondern Adressinformationen (Von wem kommen die Daten, an wen sind sie gerichtet?).

Neben den Systemgeräten (Spannungsversorgung, Busleitung usw.) wird beim KNX System generell zwischen Sensoren und Aktoren unterschieden. Sensoren sind Geräte, die Ereignisse im Gebäude erkennen (Tastenbetätigung, Bewegung, Über-/Unterschreitung eines Temperaturwerts etc.) und in Telegramme umwandeln. Anschließend versenden sie diese Telegramme (Datenpakete). Geräte, die Telegramme empfangen und die darin enthaltenen Befehle in Aktionen umwandeln, bezeichnet man als Aktoren.

7.1.2 Topologie

Der KNX-Bus wird üblicherweise bedrahtet ausgeführt. Hierfür verwendet man spezielle verdrehte Leitungen (TP, »Twisted Pair«) welche für die Verlegung in selben Rohren mit Niederspannungsleitungen zugelassen sind. Diese Leitungen erfüllen zwei Aufgaben:

¹EIB Europäischer Installations-Bus

²EHS European Home Systems

³ETS Engineering Tool Software

Sie versorgt die Teilnehmer mit der Versorgungsspannung 29VDC und ermöglicht den Informationsaustausch zwischen den Teilnehmern.

Die kleinste Installationseinheit bildet dabei die sogenannte Linie: Maximal 64 Teilnehmer werden an eine Busleitung angeschlossen und von einer Spannungsversorgung gespeist. Das Buskabel lässt sich hierbei beliebig verlegen und kann an jeder Stelle verzweigt werden. Als Busstruktur ergibt sich damit eine freie Baumstruktur. Das erlaubt einen sehr flexiblen Aufbau. Bei mehr als 64 Teilnehmern können über so genannte Linienverstärker bis zu vier Liniensegmente zu einer Linie mit maximal 255 Teilnehmern zusammen geschaltet werden, wobei die Linienverstärker als Teilnehmer mitgerechnet werden müssen. In der Praxis wird dieser maximale Linienausbau in der Regel nicht ausgenutzt, sondern beim Überschreiten von 64 Teilnehmern wird eine neue Linie angelegt. Dies macht einerseits die Anlage übersichtlicher und andererseits hat man später die Möglichkeit, die Anzahl der Telegramme in jeder Linie zu reduzieren, indem man die Filterfunktion der Linienkoppler nutzt. Der Linienkoppler zum Verbinden der beiden Linien kann auch als Linienverstärker parametrisiert werden. [14]

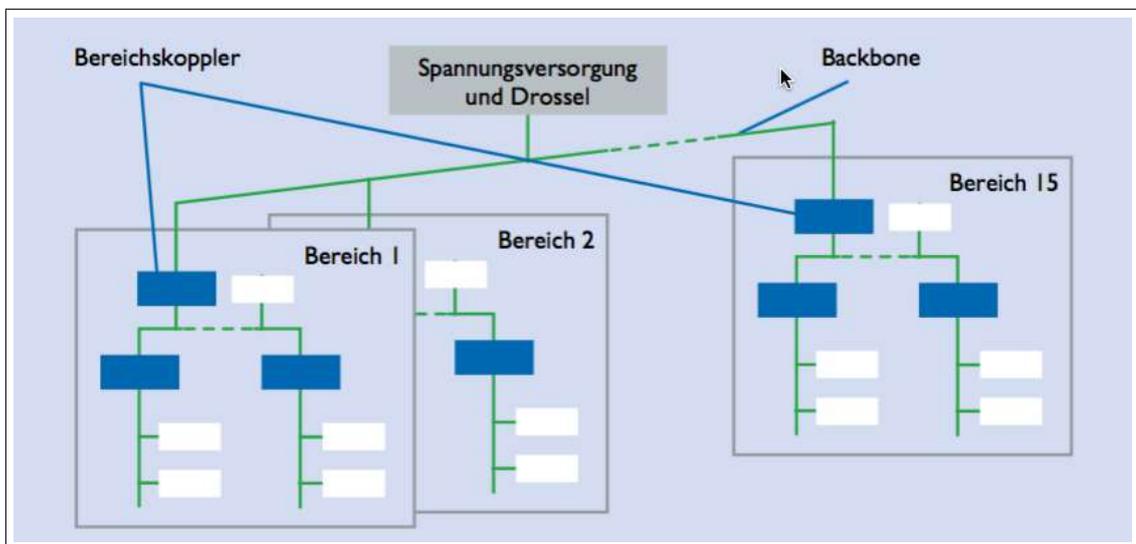


Abbildung 7.1.1: KNX-Topologie [14]

7.1.3 Telegrammaufbau

Der Informationsaustausch zwischen den Busteilnehmern erfolgt über sogenannte Telegramme. Ein Telegramm besteht aus einer Folge von Zeichen, wobei ein Zeichen aus einem Byte gebildet wird. Oft werden mehrere Zeichen zu einem Feld zusammengefasst. So ergibt sich ein Telegrammaufbau laut Abbildung 7.1.2.

Die Kommunikation lässt sich in folgende Schritte einteilen.

1. Als Erstes wird das Kontrollfeld übertragen. Das Kontrollfeld regelt Aufgaben wie die Priorität des Buszugriffs und die Telegrammwiederholung.
2. Es folgt das Adressfeld. Dieses Feld enthält die Absender und Adressdaten von Sender und Empfänger.
3. Als drittes Feld kommt das Datenfeld mit den Nutzdaten. Es ist je nach Bedarf unterschiedlich lang. Maximal kann es aus 16 Zeichen (16 Byte) bestehen.
4. Den Abschluss bildet ein Sicherungsfeld.

[14]

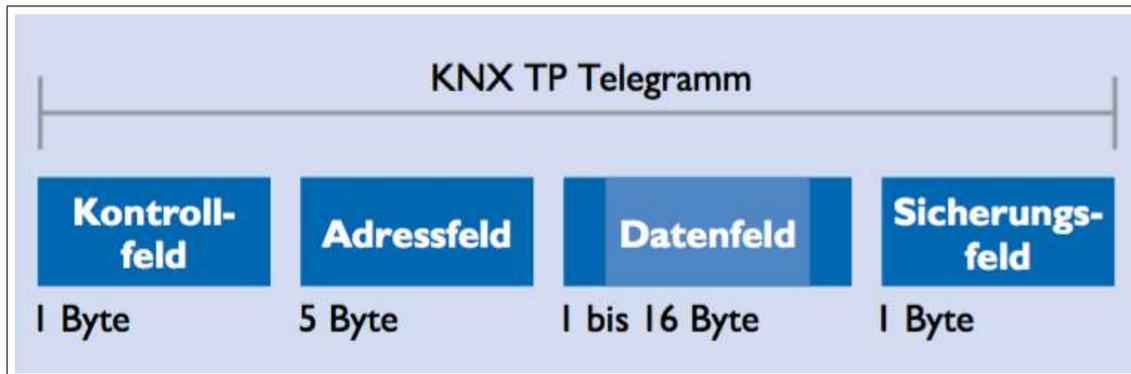


Abbildung 7.1.2: KNX TP Telegrammaufbau [14]

7.1.4 Weinzierl BAOS-Modul

Wir haben freundlicherweise von der Firma Weinzierl zwei Development-Boards, sowie die KNX-Schnittstellen »Weinzierl BAOS-Modul 820« und das »Weinzierl BAOS-Modul 822« zur Verfügung gestellt bekommen.

Hierbei handelt es sich um eine KNX-Schnittstelle, basierend auf einem Atmel ATmega128 Mikrocontroller.

Es arbeitet sekundärseitig mit dem sogenannten FT 1.2-Protokoll.

Der Name »BAOS-Modul« rührt vom BAOS-Objektserver her, mit welchem das Modul die Kommunikation regelt. Mit diesem lässt sich eine logische Entkopplung zwischen der Kommunikation mit dem Mikrocontroller und dem KNX-Bus herstellen. Damit muss der Mikrocontroller keine Kenntnis über die aktuellen Gruppenadressen der KNX-Anlage besitzen. Er kann auch kommunizieren, wenn das Senden auf den Bus, z.B. wegen kurzzeitig großer Buslast, nicht möglich ist. Abbildung 7.1.3 zeigt das Weinzierl BAOS-Modul 820.

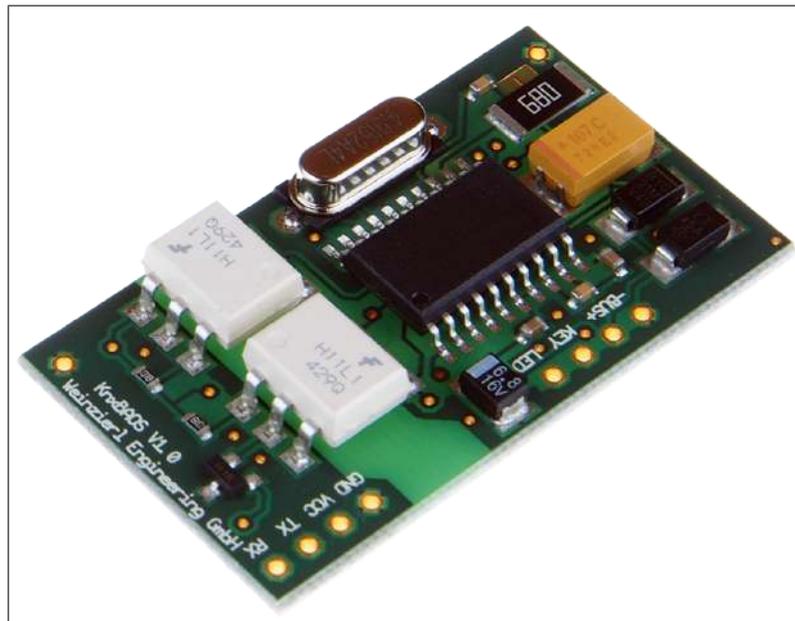


Abbildung 7.1.3: Weinzierl BAOS-Modul 820 [15]

Parametrierung in der ETS

Zur Parametrierung des Weinzierl BAOS-Moduls 820 wurde die vom Hersteller zur Verfügung gestellte ETS-Applikation verwendet. Diese kann auf der Herstellerseite unter den Downloads heruntergeladen werden. [16]

Die Tabelle 7.1.1 zeigt die verwendeten Gruppenadressen.

Tabelle 7.1.1: verwendete Gruppenadressen

verwendete Gruppenadressen			
Nr.	Datenpunkt	Gruppenadresse	Beschreibung
1	DPT 14	0/0/1	aktueller Strahlungswert in $\frac{\mu\text{Sv}}{\text{h}}$
2	DPT 01	0/0/2	Alarm Grenzwert überschritten (1 = überschritten)
3	DPT 01	0/0/3	Umschalten interner/externer Grenzwert (0 = intern, 1 = extern)
4	DPT 14	0/0/4	externen Grenzwert vorgeben $\frac{\mu\text{Sv}}{\text{h}}$
5	DPT 09	0/0/5	aktueller Temperaturwert in $^{\circ}\text{C}$
6	DPT 09	0/0/6	aktueller Luftfeuchtigkeitswert in %
7	DPT 01	0/0/7	Heizung aktiv (1 = aktiv, 0 = inaktiv)
8	DPT 01	0/0/8	Kühlung aktiv (1 = aktiv, 0 = inaktiv)
9	DPT 01	0/0/9	Summenstörung (1 = Störung)
10	DPT 01	0/0/10	zyklische Aktivitätsanzeige (alle 60 s wird ein Telegramm mit dem Inhalt »1« gesendet)

In Tabelle 7.1.2 sehen Sie die nötigen Parameterbytes.

Tabelle 7.1.2: verwendete Parameterbytes

verwendete Parameterbytes		
Nr.	Parameterbyte	Beschreibung
1	200	Grenzwert der Strahlenbelastung in $\frac{n\text{Sv}}{\text{h}}$
2	1	zyklisches Senden (1 = aktiv, 0 = inaktiv)
3	1	Senden bei Änderung (1 = aktiv, 0 = inaktiv)
4	60	Zykluszeit für zyklisches Senden des Messwerts in s
5	10	Messwert für Senden bei Änderung in $\frac{n\text{Sv}}{\text{h}}$

Hat man die Applikation in das ETS-Projekt importiert, wird das BAOS-Modul unter den Einstellungen im Reiter »Parameter« laut den Abbildungen 7.1.4 und 7.1.5 konfiguriert.

Typ von Datenpunkt 1	DPT 14 - Gleitkommawert - 4 Bytes
Beschreibung von Datenpunkt 1	aktuelle Strahlenbelastung
Typ von Datenpunkt 2	DPT 01 - Binär - 1 Bit
Beschreibung von Datenpunkt 2	Alarm Grenzwert überschritten
Typ von Datenpunkt 3	DPT 01 - Binär - 1 Bit
Beschreibung von Datenpunkt 3	interner/externer Grenzwert
Typ von Datenpunkt 4	DPT 14 - Gleitkommawert - 4 Bytes
Beschreibung von Datenpunkt 4	externer Grenzwert vorgeben
Typ von Datenpunkt 5	DPT 04 - Zeichen - 1 Byte
Beschreibung von Datenpunkt 5	aktueller Temperaturwert
Typ von Datenpunkt 6	DPT 05 - Prozentwert - 1 Byte
Beschreibung von Datenpunkt 6	aktuelle Luftfeuchtigkeit
Typ von Datenpunkt 7	DPT 01 - Binär - 1 Bit
Beschreibung von Datenpunkt 7	Heizung aktiv
Typ von Datenpunkt 8	DPT 01 - Binär - 1 Bit
Beschreibung von Datenpunkt 8	Kühlung aktiv
Typ von Datenpunkt 9	DPT 01 - Binär - 1 Bit
Beschreibung von Datenpunkt 9	Summenstörung
Typ von Datenpunkt 10	DPT 01 - Binär - 1 Bit
Beschreibung von Datenpunkt 10	zyklische Aktivitätsanzeige

Abbildung 7.1.4: ETS5 Datenpunkte

Parameter byte 1	200
Beschreibung von Parameter byte 1	Grenzwert in nSv/h
Parameter byte 2	1
Beschreibung von Parameter byte 2	zyklisches Senden an/aus
Parameter byte 3	10
Beschreibung von Parameter byte 3	Zykluszeit für zyklisch Senden
Parameter byte 4	10
Beschreibung von Parameter byte 4	Wert für Senden bei Änderung

Abbildung 7.1.5: ETS5 Parameter

Anschließend wird das Gerät programmiert. Wird eine KNX-Teilnehmer zum ersten Mal programmiert, muss zunächst die KNX-Programmiertaste am Gerät gedrückt werden, da man hierbei die Applikation, als auch die physikalische Adresse überträgt. Bei jedem weiteren Programmiervorgang muss diese Taste nicht mehr gedrückt werden, da die Adressierung ab diesem Zeitpunkt bereits über die festgelegte physikalische Adresse erfolgt.

7.2 GSM

7.2.1 Allgemein

Funkzellen

Das Mobilfunknetz ist in Zellen aufgeteilt. Der Durchmesser einer Zelle beträgt mehrere Kilometer. In jeder Zelle hat der Mobilfunknetzbetreiber eine oder mehrere Basisstationen im Einsatz. Bewegt sich nun ein Funkteilnehmer durch das Mobilfunknetz, so bewegt er sich auch durch die verschiedenen Zellen.

Das Mobiltelefon strahlt Funkwellen in alle Richtungen aus. Bei den Basisstationen unterscheidet man zwischen der omnidirektionalen und der sektorisierten Basisstation.

- Die omnidirektionale Basisstation steht im Zentrum einer Funkzelle und strahlt ihre Funkwellen in alle Richtungen (360° Abstrahlwinkel der Antenne) aus.
- Die sektorisierte Basisstation wird zur Erhöhung der Gesprächskapazität eingesetzt. Sie strahlt ihre Funkwellen nur in einem von drei Sektoren einer Funkzelle aus.

Bewegt sich ein Mobilfunkkunde jedoch am Rand einer Zelle, so erkennt das Netz, wann es besser ist, eine neue Verbindung zu einer anderen Basisstation aufzunehmen.

Das Netz entscheidet dann anhand der Verbindungsqualität, welche Basisstation für eine Verbindung besser geeignet ist.

Die Verbindungsqualität zu den Basisstationen wird ständig geprüft. Bei Bedarf wird die Basisstation gewechselt. Dabei wird die Verbindung zur alten Basisstation erst abgebrochen, wenn die neue Verbindung steht. Der Mobilfunkkunde merkt davon nichts. Seine Sprach- und Datenverbindungen werden unterbrechungsfrei fortgeführt. Abbildung 7.2.1 zeigt den prinzipiellen Aufbau eines GSM-Netzes. [17]

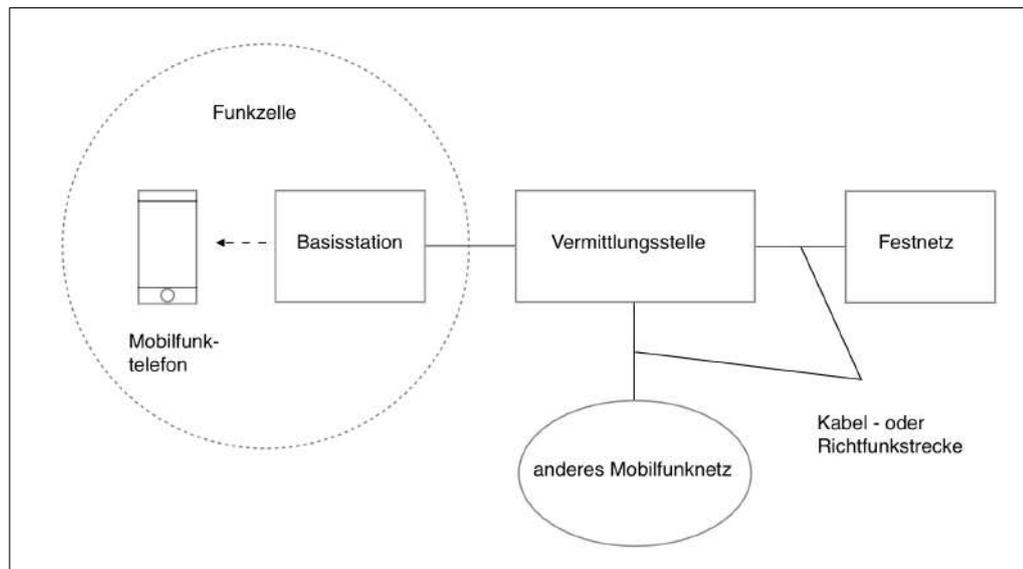


Abbildung 7.2.1: GSM Funktionsweise

Modulation

Die aus der digitalen Sprachumwandlung hervorgegangenen Signale werden in eine hochfrequente elektromagnetische Welle gewandelt, die als Transportmedium von Antenne zu Antenne dient. Diesen Vorgang nennt man Modulation.

Wegen der geringen Bandbreite und der teuren Frequenzressourcen wird beim GSM-Standard die Sprache auf viel weniger $\frac{b}{s}$ («Bits pro Sekunde») zusammengefasst als dies bei der Übertragung im Festnetz der Fall ist. Während es dort $64 \frac{kb}{s}$ sind, erfolgt die Sprachkodierung von GSM mit $24,7 \frac{kb}{s}$. [18]

Übertragung

GSM-Netze übertragen die Daten nicht kontinuierlich, sondern »pulsen« nur in bestimmten wiederkehrenden Zeiträumen, auch Zeitschlitze genannt. Während nachfolgender Zeitschlitze können andere Nutzer senden, bis der erste Nutzer wieder an der Reihe ist. Innerhalb eines Zeitschlitzes wird die Information mit Hilfe elektromagnetischer Felder als Transportmedium übertragen. Die Datenpakete werden in einem festen zeitlichen Abstand 217 mal pro Sekunde gesendet.

Beim Mobiltelefon und bei der Mobilfunkbasisstation funktioniert die Abfolge der Datenpakete unterschiedlich. Das Handy sendet die Datenpakete der Sprachinformation immer im gleichen zeitlichen Abstand nacheinander. Die Beschaffenheit der von der Basisstation gesendeten Signale ist dagegen abhängig vom tatsächlichen Gesprächsaufkommen.

In den GSM-900-Netzen findet die Funkübertragung im Frequenzbereich um 900 MHz statt, genauer gesagt in den Teilbereichen 890 MHz bis 915 MHz und 935 MHz bis 960 MHz . Bei den GSM-1800-Netzen liegen die Frequenzen zwischen 1710 MHz bis 1785 MHz und 1.805 MHz bis 1.880 MHz . Dabei wird jeweils der untere Frequenzbereich für die Übertragung vom Mobiltelefon zur Basisstation genutzt (»Uplink«), der obere Bereich dient der Übertragung in der entgegengesetzten Richtung (»Downlink«). [18]

7.2.2 GSM Alarmierung

Die GSM Anbindung kann über das EibPort der Firma BabTech erfolgen. Dieser bietet eine grafische Oberfläche, sowie KNX-Visualisierung. Es gibt verschiedene Versionen dieses Servers, wobei unserer eine GSM-Anbindung mitbringt. Den EibPort mit GSM-Kommunikation sehen Sie in Abbildung 7.2.2.

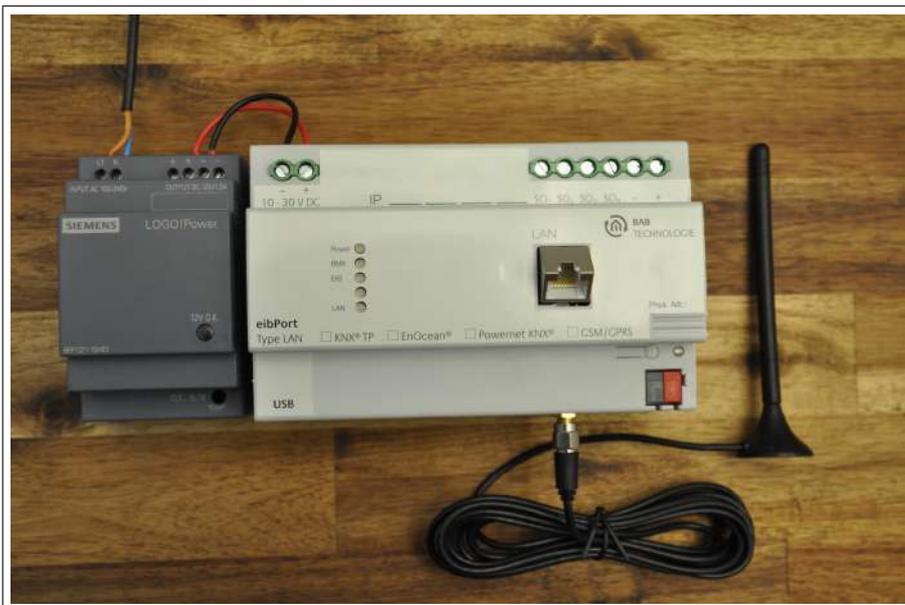


Abbildung 7.2.2: BABtec eibPort mit GSM

eibPort Editor [19]

In Abbildung 7.2.3 wird der Ablauf für die Konfiguration einer SMS-Alarmierung gezeigt.

1. Über den Editor wird der eibPort programmiert und die Visualisierung erstellt.
2. Im Job-Editor werden die Dienste konfiguriert die der eibPort ausführen soll.
3. Nun wählt man die Jobkategorie »SMS Sender« links im Fenster aus.

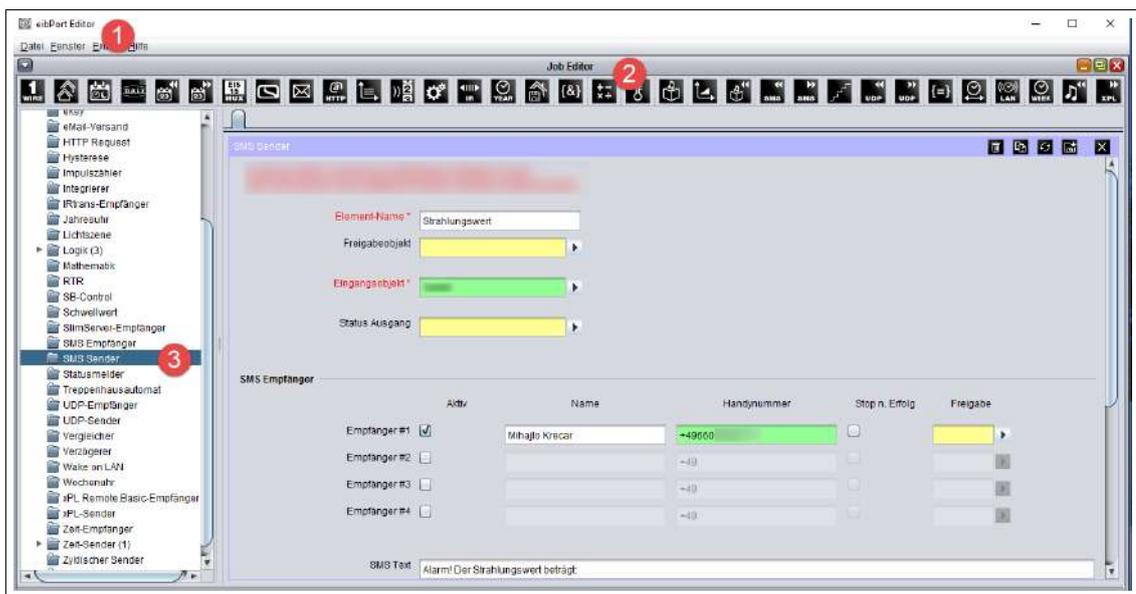


Abbildung 7.2.3: Job-Editor

Als Element-Name wurde »Strahlungswert« gewählt. Für das Eingangsobjekt muss die richtige KNX Gruppenadresse verwendet werden. In unserem Falle ist diese 0/0/1.

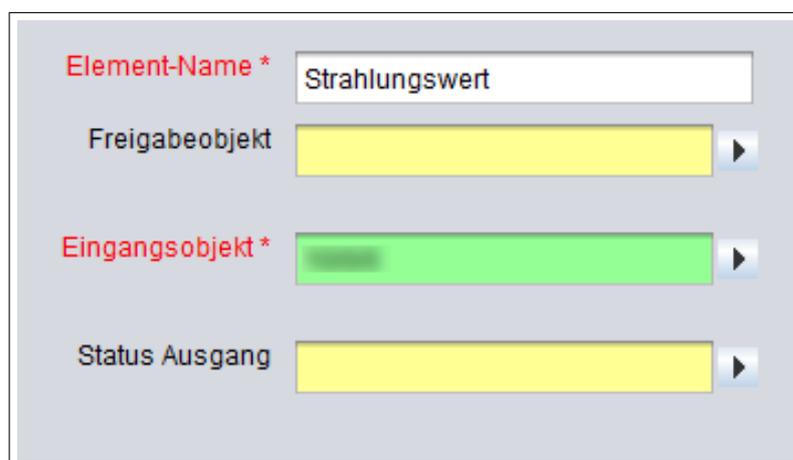


Abbildung 7.2.4: SMS Sender Einstellungen

Anschließend folgen die Schritte in Abbildung 7.2.5.

1. Es können bis zu 4 SMS Empfänger eingetragen werden. Bei der Eingabe der Telefonnummer ist zu berücksichtigen das die Ländervorwahl ebenfalls eingetragen werden muss.
2. Hier kann der gewünschte Text eingegeben werden, der dem Empfänger zugesendet wird. Nach dem Text wird automatisch der Wert angezeigt, der in den nächsten Schritten definiert wird.
3. Die Adresse beträgt 0/0/1.
4. Der Datentyp ist ein DPT 14 (früher auch EIS 9). Dieser gibt eine 32 Bit Gleitkommazahl an.
5. Unter »Format« wird die Einheit der gewichteten Dosisleistung (Äquivalentdosisleistung) in $\frac{\mu\text{Sv}}{\text{h}}$ angegeben.



SMS Empfänger

	Aktiv	Name	Handnummer	Stop n. Erfolg	Freigabe
Empfänger #1	<input checked="" type="checkbox"/>	Mihajlo Krečar	+49660	<input type="checkbox"/>	
Empfänger #2	<input type="checkbox"/>		+49	<input type="checkbox"/>	
Empfänger #3	<input type="checkbox"/>		+49	<input type="checkbox"/>	
Empfänger #4	<input type="checkbox"/>		+49	<input type="checkbox"/>	

SMS Text: Alarm! Der Strahlungswert beträgt

Werte Senden

Wert #1 ("0"):

Adresse: EIS Typ: Format (EIS 1 ein / aus): Faktor: Offset:

Abbildung 7.2.5: Werte senden

7.3 Push-Benachrichtigung mit Prowl

7.3.1 Vorbemerkungen

Beim Einsatz eines derart sensiblen Messgerätes ist es natürlich unabdingbar stets über den aktuellen Messwert benachrichtigt zu werden, um über aufgetretene Messwertüberschreitungen stets informiert zu sein. Eine übliche und stabile Möglichkeit der Benachrichtigung ist mittels GSM möglich und in Kapitel 7.2 näher erklärt. Mittlerweile sind, durch den immer häufiger werdenden Einsatz von Smartphones, jedoch besser an mobile Betriebssysteme angepasste Lösungen, eine Statusmeldung an einen Nutzer zu senden, möglich.

Durch den Einsatz von Push-Benachrichtigungen ist es möglich, einen täglichen oder wöchentlichen Statusbericht abzusetzen, ohne jedes Mal ein gebührenpflichtiges SMS abzusetzen. Die Lösung der Wahl hört auf den Namen »Prowl«. Auf Abbildung 7.3.1 sehen sie ein Beispiel einer solchen Statusmitteilung.

Prowl ist ein GROWL⁴-Client für iOS-Smartphones und wird benötigt, um Push-Nachrichten abzusetzen.

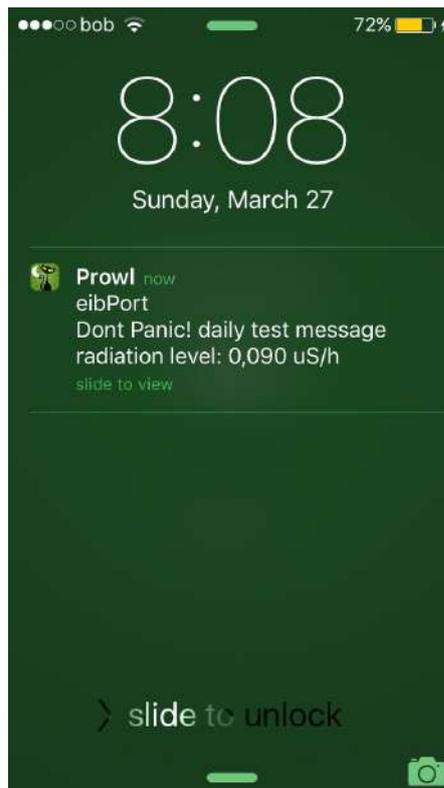


Abbildung 7.3.1: Push-Meldung am Lock-Screen

⁴Growl ist ein globales Benachrichtigungssystem für Mac OS X, siehe [20]

7.3.2 Umsetzung der Push-Mitteilung

Es gibt mehrere Möglichkeiten, eine Push-Mitteilung abzusetzen, als Auslöser der Benachrichtigung wird eine KNX-Telegramm verwendet, welches vom FalloutMeter absetzt wird. Da bereits zur GSM-Anbindung, sowie zur Visualisierung, ein babTec eibPort 3 eingesetzt wurde, haben wir uns dazu entschlossen, die Möglichkeiten des integrierten HTTP⁵-Request Logik-Jobs zu nutzen. In Verbindung mit der kostenpflichtigen iOS-App Prowl [21] kann eine Push-Nachricht generiert werden. Für Android sind ähnliche Apps verfügbar. In Ermangelung eines Android-Gerätes wurde eine Umsetzung nur mit iOS getestet.

Als Quelle für den nachfolgenden Text wurde [22] verwendet.

- Als erstes muss bei der Prowl-Webseite[23] ein Benutzerkonto angelegt werden.
- Danach kann ein sogenannter »API⁶-Key« erstellt werden.
- Nun wird noch die Prowl-App installiert, diese ist im App-Store zu finden.

Anschließend wird im eibPort der »Job-Editor« aufgerufen, und ein neuer »HTTP-Request Job« angelegt.

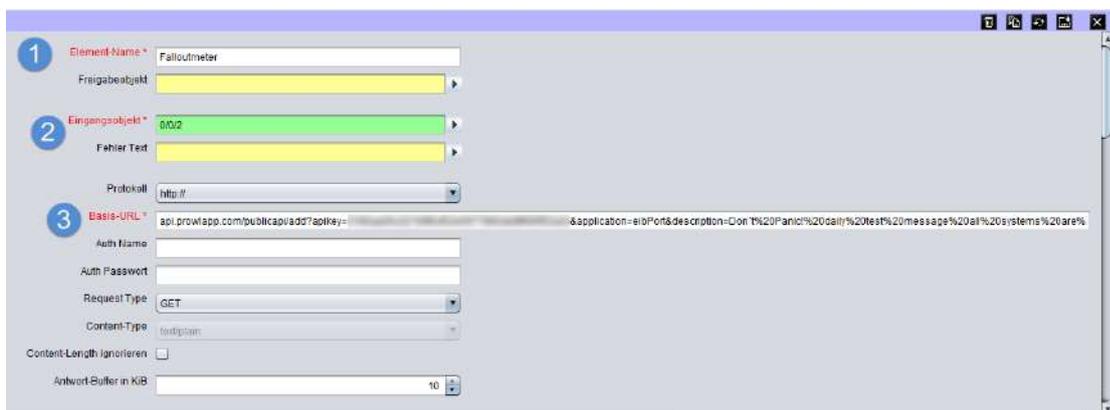


Abbildung 7.3.2: HTTP-Request Job für Push-Benachrichtigung 1

⁵HTTP Hypertext Transfer Protocol

⁶Application Programming Interface



	Adresse	EIS Typ	Format (EIS 1 ein / aus)	Faktor	Offset
4 Wert1 (*0*)	0/0/1	EIS 9 (4 Byte FP)	%f uS/h	1	0
Wert2 (*1*)		EIS 1 (1 Bit)			
Wert3 (*2*)		EIS 1 (1 Bit)			
Wert4 (*3*)		EIS 1 (1 Bit)			
Wert5 (*4*)		EIS 1 (1 Bit)			
Wert6 (*5*)		EIS 1 (1 Bit)			
Wert7 (*6*)		EIS 1 (1 Bit)			
Wert8 (*7*)		EIS 1 (1 Bit)			
Wert9 (*8*)		EIS 1 (1 Bit)			
Wert10 (*9*)		EIS 1 (1 Bit)			

Abbildung 7.3.3: HTTP-Request Job für Push-Benachrichtigung 2

Beschreibung der Einstellungen zu Abb. 7.3.2 und Abb. 7.3.3

1. Als erstes sollte ein aussagekräftiger Name für den Logikjob vergeben werden.
2. Die hier eingetragene Gruppenadresse ist der Auslöser, hier kann die 1 Bit Adresse »Grenzwert überschritten« des FalloutMeter angegeben werden, sobald diese Adresse gesendet wird, wird der Job aktiv.
3. Die Basis-URL⁷ ist immer: `api.prowlapp.com/publicapi/add?apikey=<API-Key>&application=<name>&description=<message>` ⇒ Die Angaben in eckigen Klammern sind individuell anzugeben.
4. Abschließend wird noch eine Messwertübergabe umgesetzt, um stets über die aktuelle Lage informiert zu sein. Bei »Wert1« wird die Gruppenadresse des Messwertes, samt Datentyp und Einheit angegeben.

Abschließend können alle Einstellungen noch im ETS-Gruppenmonitor mit der Funktion »Werte lesen/schreiben« getestet werden. Hierzu wird einfach ein Test-Telegramm auf die Auslöseadresse gesendet. Die übermittelte Benachrichtigung ist nun in Abbildung 7.3.1 zu sehen.

⁷URL Uniform Resource Locator

8 Visualisierung der Messwerte

Zur Visualisierung der Messwerte stehen 2 mögliche KNX-Visualisationen zur Verfügung, die SmartVisu mithilfe der Software Smarthome.py, sowie die CubeVision der Firma BabTech.

8.1 Smarthome.py

Bei Smarthome.py handelt es sich um ein kostenloses, unter GNU lizenziertes Programm, welches einem UNIX-basierenden Rechner erlaubt, auf den KNX-Bus zuzugreifen. Diese Software bringt die SmartVisu als kostenlose Visualisierung mit. In unserem Fall verwenden wir einen Raspberry Pi als Server und ein KNX-IP-Gateway, damit dieser auf den Bus zugreifen kann. Alternativ dazu kann man auch die sogenannte »ROT¹-Extension« benutzen, dies ist eine kleine Platine, die man direkt auf die GPIO's² des Raspberry Pi stecken kann und so eine physikalische Busanbindung bietet. Um die SmartVisu nutzen zu können, muss zuerst ein Server mit Smarthome.py aufgesetzt werden. Die dafür nötigen Schritte werden nun erklärt.

8.1.1 Raspberry Pi vorbereiten

Zuerst muss »Raspbian« auf dem Raspberry Pi eingerichtet werden. Dies ist eine an den Raspberry Pi angepasste Debian-Distribution und das standardmäßige Betriebssystem des Raspberry Pi. Die IMG-Datei des Betriebssystems kann von der Raspberry Pi Homepage heruntergeladen werden und wird dann auf die SD-Karte geflasht. [24]

8.1.2 Smarthome.py einrichten

Nun muss man sich in der Kommandozeile mit »root« anmelden.

Folgende Befehle dienen zum Updaten des Systems sowie herunterladen der nötigen Pakete.

```
1 apt-get update
2 apt-get -y install dialog openssh-server apache2 php5 php5-json openntpd python3 python3-dev
  python3-setuptools git-core unzip wget libawl-php php5-curl build-essential
3 easy_install3 pip
4 pip install ephem
```

Listing 8.1: System updaten

¹ROT RTC-OneWire-TPUART (EIB/KNX)

²GPIO General Purpose Input/Output

Nun wird der User »smarhome« angelegt.

```

1 adduser smarhome
2 usermod -G www-data,sudo -a smarhome

```

Listing 8.2: Benutzer anlegen

Nun werden die nötigen Dateien vom Github-Repository heruntergeladen.

```

1 cd /usr/local
2 git clone git://github.com/mknx/smarhome.git
3 chown -R smarhome:smarhome /usr/local/smarhome

```

Listing 8.3: smarhome downloaden

Jetzt werden die Zugriffs- und Änderungs-Zeitstempel für die Datei angelegt.

```

1 su smarhome
2 cd /usr/local/smarhome/etc
3 touch logic.conf

```

Listing 8.4: Zugriffs- und Änderungs-Zeitstempel bearbeiten

Nun wird die Datei »smarhome.conf« bearbeitet, hier kann der Aufenthaltsort für die Wetterdaten und Koordinaten festgelegt werden.

```

1 sudo nano smarhome.conf

```

Listing 8.5: smarhome.conf bearbeiten

Diese Datei bekommt nun folgenden Inhalt.

```

1 # smarhome.conf
2
3 lat = 48.13
4 lon = 14.10
5 elev = 301
6 tz = 'Europe/Linz'

```

Listing 8.6: smarhome.conf Inhalt

Nun wird die Datei »plugin.conf« bearbeitet.

```

1 sudo nano plugin.conf

```

Listing 8.7: plugin.conf bearbeiten

Standardmäßig bekommt diese Datei folgenden Inhalt.

```

1 # plugin.conf
2
3 [knx]
4     class_name = KNX
5     class_path = plugins.knx
6     host = <IP.des.Raspberry.Pi>
7     port = 6720
8 #     send_time = 600 # update date/time every 600 seconds, default none
9 #     time_ga = 1/1/1 # default none
10 #     date_ga = 1/1/2 # default none
11
12 [visu]

```

8.1 Smarthome.py

```

13 class_name = WebSocket
14 class_path = plugins.visu
15 ip = <IP.des.Raspberry.Pi>
16 port = 2424
17 acl = rw
18 smartvisu_dir = /var/www/smartvisu
19
20 [cli]
21 class_name = CLI
22 class_path = plugins.cli
23 ip = 0.0.0.0
24 update = True
25
26 [sql]
27 class_name = SQL
28 class_path = plugins.sqlite

```

Listing 8.8: plugin.conf Inhalt

Nun wird noch die Datei »smarthome« bearbeitet.

```

1 exit
2 cd /etc/init.d
3 sudo nano smarthome

```

Listing 8.9: smarthome Script anlegen

Diese bekommt nun folgenden Inhalt.

```

1 #!/bin/sh
2 ### BEGIN INIT INFO
3 # Provides:          smarthome
4 # Required-Start:    $syslog $network
5 # Required-Stop:     $syslog $network
6 # Should-Start:      eibd owserver
7 # Should-Stop:       eibd owserver
8 # Default-Start:     2 3 4 5
9 # Default-Stop:      0 1 6
10 # Short-Description: Start SmartHome.py
11 ### END INIT INFO
12
13 DESC="SmartHome.py"
14 NAME=smarthome.py
15 SH_ARGS="-q"
16 SH_UID='smarthome'
17
18 PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin
19 DIR=/usr/local/smarthome/bin
20 DAEMON=$DIR/$NAME
21 SCRIPTNAME=/etc/init.d/$NAME
22 PIDFILE=/var/run/$NAME.pid
23
24 [ -x "$DAEMON" ] || exit 0
25 [ -r /etc/default/$NAME ] && . /etc/default/$NAME
26
27 DAEMON_ARGS="$SH_ARGS"
28
29 do_start()
30 {
31     touch $PIDFILE
32     chown $SH_UID:$SH_UID $PIDFILE

```

```

33 start-stop-daemon --start --user $SH_UID --group $SH_UID --chuid $SH_UID --pidfile $PIDFILE
   --startas $DAEMON --test > /dev/null || return 1
34 start-stop-daemon --start --user $SH_UID --group $SH_UID --chuid $SH_UID --pidfile $PIDFILE
   --startas $DAEMON -- $DAEMON_ARGS || return 2
35     ps ax | grep $DAEMON | grep -v grep | awk '{print $1}' > $PIDFILE
36 }
37
38 do_stop()
39 {
40     start-stop-daemon --stop --retry=TERM/30/KILL/5 --pidfile $PIDFILE
41     RETVAL="$?"
42     [ "$RETVAL" = 2 ] && return 2
43     start-stop-daemon --stop --quiet --oknodo --retry=0/30/KILL/5 --exec $DAEMON
44     [ "$?" = 2 ] && return 2
45     rm -f $PIDFILE 2> /dev/null
46     return "$RETVAL"
47 }
48
49 do_reload() {
50     start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE --name $NAME
51     return 0
52 }
53
54 case "$1" in
55     start)
56         do_start
57         ;;
58     stop)
59         do_stop
60         ;;
61     restart)
62         echo "Restarting $DESC" "$NAME"
63         do_stop
64         sleep 1
65         do_start
66         ;;
67     *)
68         echo "Usage: $SCRIPTNAME {start|stop|restart}" >&2
69         exit 3
70         ;;
71
72 esac
73
74 exit 0

```

Listing 8.10: smarhome Script

Mit folgendem Befehl werden die Dateirechte richtig gesetzt.

```
1 chmod 755 smarhome
```

Listing 8.11: smarhome Dateirechte

Folgender Befehl aktiviert den Autostart.

```
1 inserv smarhome
```

Listing 8.12: smarhome Autostart

8.1.3 SmartVisu einrichten

Zuerst wird die aktuelle Version heruntergeladen.

```
1 cd /var/www
2 rm index.html
3 wget http://smartvisu.de/download/smartVISU_2.7.zip
4 unzip smartVISU_2.7.zip
5 rm smartVISU_2.7.zip
```

Listing 8.13: SmartVisu herunterladen

Nun werden die Rechte gesetzt und der Benutzer angelegt.

```
1 chown -R www-data:www-data smartVISU
2 chmod -R 775 smartVISU
```

Listing 8.14: SmartVisu Dateirechte

Nun wird »smarthome« gestartet.

```
1 /etc/init.d/smarthome start
```

Listing 8.15: smarthome starten

Nun kann man den Log auf Fehlermeldungen kontrollieren, die einzige die erscheinen darf, ist ein Verbindungsfehler mit »eibd« oder »onewire«, da beides noch nicht installiert ist. In dieser Anleitung wird One-Wire nicht installiert, da es nicht gebraucht wird.

```
1 tail /usr/local/smarthome/var/log/smarthome.log
```

Listing 8.16: Fehlerlog

8.1.4 eibd einrichten

Die folgende Befehle fügen die Repository (Packetquelle) für die KNX-Pakete hinzu. Zuerst wird die zu verändernde Datei geöffnet.

```
1 nano /etc/apt/sources.list
```

Listing 8.17: Packetquellen bearbeiten

Nun wird dieser Datei folgender Inhalt am Ende hinzugefügt.

```
1 deb http://www.auto.tuwien.ac.at/~mkoegler/debian eib main
2 deb-src http://www.auto.tuwien.ac.at/~mkoegler/debian eib main
```

Listing 8.18: Packetquelle

Nun wird »eibd« installiert.

```
1 apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 7DC4FF45991BF361
2 apt-get update
3 apt-get -y --force-yes install bcusdk
```

Listing 8.19: eibd installieren

Nun konfigurieren wir die Datei »eibd«.

```
1 nano /etc/default/eibd
```

Listing 8.20: eibd Konfiguration anlegen

Diese Datei bekommt nun folgenden Inhalt.

```
1 EIB_ARGS="--daemon --Server --Tunnelling --Discovery --GroupCache --listen-tcp"
2 EIB_ADDR="0.0.1"
3 EIB_IF="ipt:<ip-des-knx-ip-modules>"
4 EIB_UID="smarthome"
```

Listing 8.21: eibd Konfiguration

Nun wird das Startup-Script angelegt.

```
1 cd /etc/init.d
2 nano eibd
```

Listing 8.22: eibd Script anlegen

Dieses Script bekommt nun diesen Inhalt.

```
1 #!/bin/sh
2 ### BEGIN INIT INFO
3 # Provides:          eibd
4 # Required-Start:    $syslog $network
5 # Required-Stop:     $syslog $network
6 # Default-Start:     2 3 4 5
7 # Default-Stop:      0 1 6
8 # Short-Description: Start the eib daemon.
9 ### END INIT INFO
10
11 DESC="KNX daemon 'eibd'"
12 NAME=eibd
13 EIB_ARGS="--daemon --Server --Tunnelling --Discovery --GroupCache --listen-tcp"
14 EIB_UID="smarthome"
15
16 PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin
17 #Ubuntu: DAEMON=/usr/bin/$NAME
18 #Debian:  DAEMON=/usr/local/bin/$NAME
19 DAEMON=/usr/local/bin/$NAME
20 PIDFILE=/var/run/$NAME.pid
21 SCRIPTNAME=/etc/init.d/$NAME
22
23 [ -x "$DAEMON" ] || exit 0
24
25 [ -r /etc/default/$NAME ] && . /etc/default/$NAME
26
27 DAEMON_ARGS="$EIB_ARGS --pid-file=$PIDFILE --eibaddr=$EIB_ADDR $EIB_IF"
28
29 . /lib/init/vars.sh
30
31 do_start()
32 {
33     touch $PIDFILE
34     chown $EIB_UID $PIDFILE
35     start-stop-daemon --start --quiet --chuid $EIB_UID --pidfile $PIDFILE --exec $DAEMON --test >
36     /dev/null || return 1
37     start-stop-daemon --start --quiet --chuid $EIB_UID --pidfile $PIDFILE --exec $DAEMON --
38     $DAEMON_ARGS || return 2
39 }
```

```

38
39 do_stop()
40 {
41     start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE --name $NAME
42     RETVAL="$?"
43     [ "$RETVAL" = 2 ] && return 2
44     start-stop-daemon --stop --quiet --oknodo --retry=0/30/KILL/5 --exec $DAEMON
45     [ "$?" = 2 ] && return 2
46     rm -f $PIDFILE 2> /dev/null
47     return "$RETVAL"
48 }
49
50 do_reload() {
51     start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE --name $NAME
52     return 0
53 }
54
55 case "$1" in
56     start)
57         do_start
58         ;;
59     stop)
60         do_stop
61         ;;
62     #reload|force-reload)
63         #echo "Reloading $DESC" "$NAME"
64         #do_reload
65         #log_end_msg $?
66         #;;
67     restart)
68         #
69         # If the "reload" option is implemented then remove the
70         # 'force-reload' alias
71         #
72         echo "Restarting $DESC" "$NAME"
73         do_stop
74         sleep 1
75         do_start
76         ;;
77     *)
78         echo "Usage: $SCRIPTNAME {start|stop|restart}" >&2
79         exit 3
80         ;;
81 esac
82 exit 0

```

Listing 8.23: eibd Script Inhalt

Mit folgendem Befehl werden die Dateirechte richtig gesetzt.

```
1 chmod 755 eibd
```

Listing 8.24: eibd Dateirechte

Folgender Befehl aktiviert den Autostart.

```
1 insserv eibd
```

Listing 8.25: eibd Autostart

Nun wird der Daemon «eibd» gestartet.

```
1 /etc/init.d/eibd start
```

Listing 8.26: eibd starten

Zum Verwalten oder Neustarten von »Smarthome.py« kann man im Browser folgende URL aufrufen.

```
1 https://<IP.des.Raspberry.Pi>:8080
```

Listing 8.27: SmartControl

Unter »Show Services« lassen sich nun die verschiedenen Dienste starten bzw. neustarten. Auf Abbildung 8.1.1 sehen die die Übersichtsseite der laufenden Dienste.

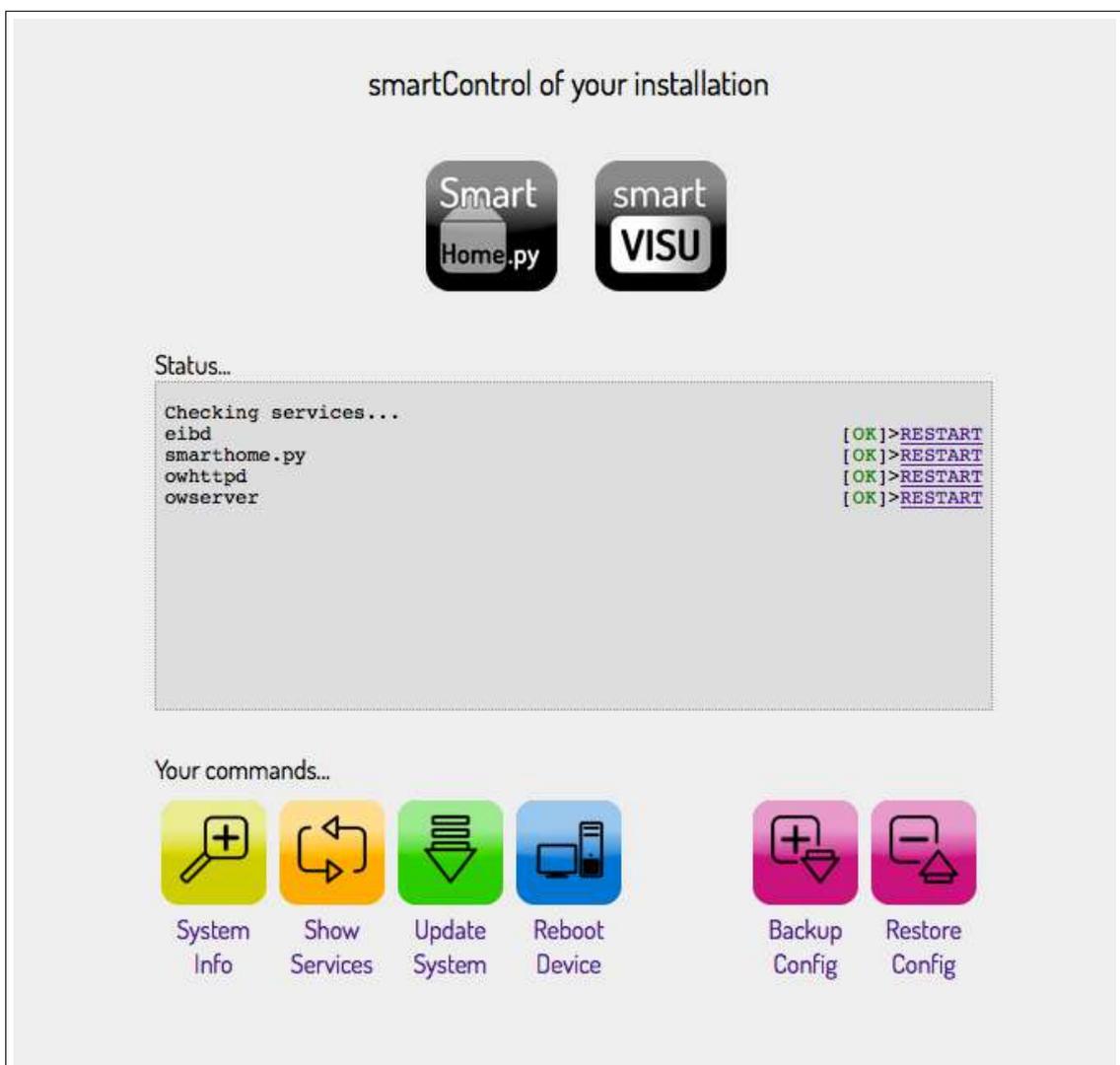


Abbildung 8.1.1: SmartControl

8.1.5 SmartVisu Webseite konfigurieren

Nun müssen im Verzeichnis »/usr/smarthome/items/« in den ».conf« - Dateien die benötigten Gruppenadressen hierarchisch hinterlegt werden.

Dies geschieht in folgender Form.

```
1 [FirstFloor]
2
3 [[LivingRoom]]
4
5     name = Living Room
6
7     [[[CeilingLamp]]]
8
9         [[[[Switch]]]]
10
11             type = bool
12             visu_acl = rw
13             knx_dpt = 1
14             knx_listen = 1/1/1
15             knx_send = 1/1/0
16             knx_init = 1/1/1
17
18         [[[Blinds]]]
19
20             [[[[Move]]]]
21
22                 type = bool
23                 visu_acl = rw
24                 enforce_updates = yes
25                 knx_dpt = 1
26                 knx_send = 2/1/0
27
28             [[[[Stop]]]]
29
30                 type = bool
31                 visu_acl = rw
32                 enforce_updates = yes
33                 knx_dpt = 1
34                 knx_send = 2/1/1
35
36             [[[[Position]]]]
37
38                 type = num
39                 visu_acl = rw
40                 knx_dpt = 5
41                 knx_listen = 2/1/3
42                 knx_send = 2/1/2
43                 knx_init = 2/1/3
```

Listing 8.28: items

Nun wird im Verzeichnis »var/www/smartvisu/« die Datei »_template« kopiert und unter einem neuem Namen wieder eingefügt, dies stellt die Grundlage für die neue Web-Visualisation.

Jetzt können die einzelnen HTML-Seiten nach eigenen Wünschen verändert, bzw. neu geschrieben werden.

Schalter

Folgender Befehl steht für einen einfachen Schalter.

```
1 {{ basic.switch('id', 'gad', 'pic_on', 'pic_off', 'val_on', 'val_off') }}
```

Listing 8.29: Schalter

- »id« steht für die einmalige ID des Elements.
- »gad« steht für die Adresse des zu schaltenden Elements.
- »pic_on« steht für das darzustellende Bild im Zustand 1.
- »pic_off« steht für das darzustellende Bild im Zustand 0.
- »val_on« steht für den Wert im eingeschalteten Zustand.
- »val_off« steht für den Wert im ausgeschalteten Zustand.

Jalousieschalter

Zum Erstellen eines Jalousieschalters wird folgender Code verwendet.

```
1 {{ basic.shutter('id', 'gad_pos', 'gad_angle', 'min', 'max', 'step', 'mode') }}
```

Listing 8.30: Jalousieschalter

- »id« ist die einmalige ID der Funktion.
- »gad_pos« ist die Sollposition der Jalousie.
- »gad_angle« steht für den gewollten Neigungsgrad.
- »min« ist jener Wert, bei dem die Jalousien schließen sollen.
- »step« ist die Schrittweite der Werte.
- »mode« ist der Modus der Jalousien. (Hierbei steht »half« für Neigungen von -1 bis 1 und »full« für Neigungen von 0 bis 1.)

Messwerte

Zum Darstellen eines einzelnen Messwertes wird folgender Code verwendet.

```
1 {{ basic.value('id', 'gad', 'unit', 'tag') }}
```

Listing 8.31: Messwerte

- »id« steht für die einmalige ID.

- »gad« ist der darzustellende Messwert.
- »unit« ist die Bezeichnung der verwendeten Einheit.
- »tag« ist der beigefügte HTML-Tag.

Bei unserem Messwert der aktuellen ionisierenden Strahlung in $\frac{\mu\text{Sv}}{\text{h}}$ handelt es sich um einen Messwert vom Datenpunkt 14, also einer 4-Byte Gleitkommazahl. Diese kann mithilfe obigen Codes angezeigt werden.

Diagramm

Zum Darstellen eines Diagrammes wird folgender Code verwendet.

```
1 {{ plot.period('id', 'gad', 'mode', 'tmin', 'tmax', 'ymin', 'ymax', 'step', 'label', 'color',
  'exposure', 'axes', 'zoom') }}
```

Listing 8.32: Diagramm

Hierbei gilt folgende Legende:

- »id« steht für die einmalige ID.
- »gad« ist der zu sendende Wert.
- »mode« ist der Betriebsmodus
- »tmin« ist das minimale Zeitintervall der X-Achse.
- »tmax« ist das maximale Zeitintervall der X-Achse.
- »ymin« ist der minimale Wert auf der Y-Achse.
- »ymax« ist der maximale Wert auf der Y-Achse.
- »step« ist die Schrittweite.
- »label« ist die Beschriftung der einzelnen Messwerte.
- »color« ist die Achsenfarbe.
- »exposure« ist die Art der Achse. (z.B. »line«, »stair«, »spline«, »area«, »areaspline« oder »column«)
- »axes« ist die Achsenbeschriftung.
- »zoom« ist die Größe des möglichen Zoom.

Diese Art von Diagramm kann auch zur Darstellung des Messwertes der Strahlenbelastung verwendet werden.

Farbwahlschalter

Zum Erstellen eines Farbwahlschalters für RGB wird folgender Code verwendet.

```
1 {{ basic.colordisc('id', 'gad_r', 'gad_g', 'gad_b', 'min', 'max', 'step', 'colors') }}
```

Listing 8.33: Farbwahlschalter

Hierbei gilt folgende Legende:

- »id« ist die einmalige ID des Elements.
- »gad_r« ist der Wert der Farbe ROT.
- »gad_g« ist der Wert der Farbe GRÜN.
- »gad_b« ist der Wert der Farbe BLAU.
- »min« ist der minimale Wert.
- »max« ist der maximale Wert.
- »step« ist die Schrittweite.
- »colors« ist die Anzahl der Farbflächen.

Webseite

Der Weg, den Funktionen die Werte der Gruppenadressen zu übermitteln, erfolgt in der gleichen Struktur, in der man in den ».conf« - Dateien die KNX-Gruppenadressen hinterlegt hat. Um auf die einzelnen Gruppenadressen zuzugreifen, werden hier die einzelnen Teile durch Punkte getrennt. Dies kann folgendermaßen aussehen.

```
1 FirstFloor.LivingRoom.CeilingLamp.Switch
```

Listing 8.34: Verweis auf GA

Diese Elemente werden üblicherweise in einer Blockdarstellung auf der Webseite aufgeteilt. Der Code hierfür ist folgender.

```
1 <div class="block">
2   <div class="set-2" data-role="collapsible-set" data-theme="c" data-content-theme="a"
3     data-mini="true">
4     <div data-role="collapsible" data-collapsed="false">
5
6       <h3>Light</h3>
7
8       <table width="100%">
9         <td align="left">
10
11           </td>
12         </table>
13
14     </div>
15 </div>
```

Listing 8.35: Blockdarstellung

Letztendlich könnte eine HTML-Seite der Visualisation beispielsweise so aussehen.

```
1 /**
2 * -----
3 * @package    smartVISU
4 * @author     Christian Haun
5 * @license    GPL [http://www.gnu.de]
6 * -----
7 */
8
9 {% import "widget_shutter.html" as shutter %}
10 {% import "multimedia.html" as multi %}
11 {% import "widget_uzsu.html" as uzsu %}
12
13 {% extends "base.html" %}
14
15 {% block sidebar %}
16     {% include 'rooms/rooms_menu.html' %}
17 {% endblock %}
18
19 {% block content %}
20
21     <h1><img class="icon" src='{{ icon0 }}it_pc.png'>Christian</h1>
22
23     <div class="block">
24         <div class="set-2" data-role="collapsible-set" data-theme="c" data-content-theme="a"
25           data-mini="true">
```

```

25     <div data-role="collapsible" data-collapsed="false">
26
27         <h3>Light</h3>
28
29         <table width="100%">
30             <td align="left">
31
32                 {{ device.dimmer('Christian_CeilingLamp', 'Ceiling Lamp',
33 'FirstFloor.Christian.CeilingLamp.Switch', 'FirstFloor.Christian.CeilingLamp.Value') }}<br />
34                 {{ basic.switch('Christian_DeskLamp',
35 'FirstFloor.Christian.DeskLamp.Switch', icon1~'light_office.png', icon0~'light_office.png')
36 }} Desk Lamp<br />
37                 {{ basic.switch('Christian_RGBBed',
38 'FirstFloor.Christian.RGB_Bed.Switch', icon1~'light_led.png', icon0~'light_led.png') }} RGB
39 Bed
40
41             </td>
42         </table>
43
44     </div>
45 </div>
46 </div>
47
48 <div class="block">
49 <div class="set-2" data-role="collapsible-set" data-theme="c" data-content-theme="a"
50 data-mini="true">
51 <div data-role="collapsible" data-collapsed="false">
52
53     <h3>Multimedia</h3>
54
55     <table width="100%">
56         <td align="left">
57
58             {{ basic.switch('Christian_SocketStereo',
59 'FirstFloor.Christian.Socket_Stereo.Switch', icon1~'audio_sound.png',
60 icon0~'audio_sound.png') }} HiFi-System<br />
61             {{ basic.switch('Christian_SocketPhono',
62 'FirstFloor.Christian.Socket_Phono.Switch', icon1~'audio_headphone.png',
63 icon0~'audio_headphone.png') }} Phonograph<br />
64             {{ basic.switch('Christian_SocketMediacenter',
65 'FirstFloor.Christian.Socket_Mediacenter.Switch', icon1~'scene_gaming.png',
66 icon0~'scene_gaming.png') }} Mediacenter<br />
67             {{ basic.switch('Christian_SocketTV',
68 'FirstFloor.Christian.Socket_Television.Switch', icon1~'it_television.png',
69 icon0~'it_television.png') }} TV
70
71         </td>
72     </table>
73
74 </div>
75 </div>
76 </div>
77
78 <div class="block" style="width: 50%">
79 <div class="set-2" data-role="collapsible-set" data-theme="c" data-content-theme="a"
80 data-mini="true">
81 <div data-role="collapsible" data-collapsed="false">
82
83     <h3>Blinds</h3>

```

```

71         {{ shutter.one ('Christian_Blinds', 'Christian',
72         'FirstFloor.Christian.Blinds.Move', 'FirstFloor.Christian.Blinds.Stop',
73         'FirstFloor.Christian.Blinds.Position') }}
74     </div>
75 </div>
76
77 <div class="block" style="width: 50%">
78 <div class="set-2" data-role="collapsible-set" data-theme="c" data-content-theme="a"
79     data-mini="true">
80     <div data-role="collapsible" data-collapsed="false">
81         <h3>Comfort Chart</h3>
82
83         {{ plot.comfortchart('Christian_ComfortChart',
84         'FirstFloor.Christian.Temperature', 'FirstFloor.Christian.Humidity') }}
85
86         <table width=100%>
87             <tr>
88                 <td align=left style="padding-left: 10px;">
89                     Carbon Dioxide:
90                 </td>
91                 <td align=right>
92                     {{ basic.float('Christian_CarbonDioxide',
93         'FirstFloor.Christian.CarbonDioxide', 'ppm') }}
94                 </td>
95             </tr>
96         </table>
97     </div>
98 </div>
99 </div>
100 </div>
101 </div>
102
103 <div class="block">
104 <div class="set-2" data-role="collapsible-set" data-theme="c" data-content-theme="a"
105     data-mini="true">
106     <div data-role="collapsible" data-collapsed="false">
107         <h3>RGBW Lights Christian Bookshelf</h3>
108
109         <table width="100%">
110             <td align="left">
111
112                 {{ basic.dual('Christian_RGBWBookshelfSwitch',
113         'FirstFloor.Christian.RGBW_Bookshelf.Switch', icon1~'light_led.png', icon0~'light_led.png',
114         '', '', 'midi') }}
115                 {{ basic.colordisc('Christian_RGBWBookshelfColour',
116         'FirstFloor.Christian.RGBW_Bookshelf.Red', 'FirstFloor.Christian.RGBW_Bookshelf.Green',
117         'FirstFloor.Christian.RGBW_Bookshelf.Blue', 0, 255, 16, 20) }} RGBW Light Bookshelf<br /><br />
118
119         <div class="preblock">
120             <span data-role="controlgroup" data-type="horizontal">
121
122                 Scenes<br />
123
124                 {{ basic.button('Christian_RGBWBookshelfSequence1',
125         'FirstFloor.Christian.RGBW_Bookshelf.Sequence1_Colourful', '', icon0~"scene_scene.png", '1',

```

```

121     'midi') }}
122         {{ basic.button('Christian_RGBWBookshelfSequence4',
123         'FirstFloor.Christian.RGBW_Bookshelf.Sequence4_Sunrise', '', icon0~"weather_sunrise.png",
124         '1', 'midi') }}
125         {{ basic.button('Christian_RGBWBookshelfSequence5',
126         'FirstFloor.Christian.RGBW_Bookshelf.Sequence5_Lounge', '', icon0~"scene_party.png", '1',
127         'midi') }}
128     </span>
129 </div>
130
131     </td>
132 </table>
133
134 </div>
135
136 <div data-role="collapsible" data-collapsed="true">
137
138     <h3>RGBW Lights seperate</h3>
139
140     <label for="Christian_RGBW_Bookshelf_Red">Red<br /></label>
141
142     {{ basic.slider('Christian_RGBWBookshelfRed',
143     'FirstFloor.Christian.RGBW_Bookshelf.Red') }}
144
145     <br /><label for="Christian_RGBW_Bookshelf_Green">Green<br /></label>
146
147     {{ basic.slider('Christian_RGBWBookshelfGreen',
148     'FirstFloor.Christian.RGBW_Bookshelf.Green') }}
149
150     <br /><label for="Christian_RGBW_Bookshelf_Blue">Blue<br /></label>
151
152     {{ basic.slider('Christian_RGBWBookshelfBlue',
153     'FirstFloor.Christian.RGBW_Bookshelf.Blue') }}
154
155     <br /><label for="Christian_RGBW_Bookshelf_White">White<br /></label>
156
157     {{ basic.slider('Christian_RGBWBookshelfWhite',
158     'FirstFloor.Christian.RGBW_Bookshelf.White') }}
159
160 </div>
161 </div>
162 </div>
163 {% endblock %}

```

Listing 8.36: HTML-Code einer Seite

Ist alles fertig konfiguriert, rufen wir die Webseite auf.

```
1 https://<IP.des.Raspberry.Pi>/smartvisu
```

Listing 8.37: SmartVisu aufrufen

Nun sollte man sicherstellen, dass unter dem Menüpunkt »Config« unter »I/O Connection« der Punkt »Smarthome.py« ausgewählt ist. Desweiteren ist noch zu beachten, dass unter »Address« die IP-Adresse des Raspberry Pi eingestellt ist.

Unter »Interface« wählt man nun die neu erstellte Visualisierung aus. Danach wird »Pagecache« und »Animation« aktiviert. In Abbildung 8.1.2 sehen Sie die zu treffenden Einstellungen.

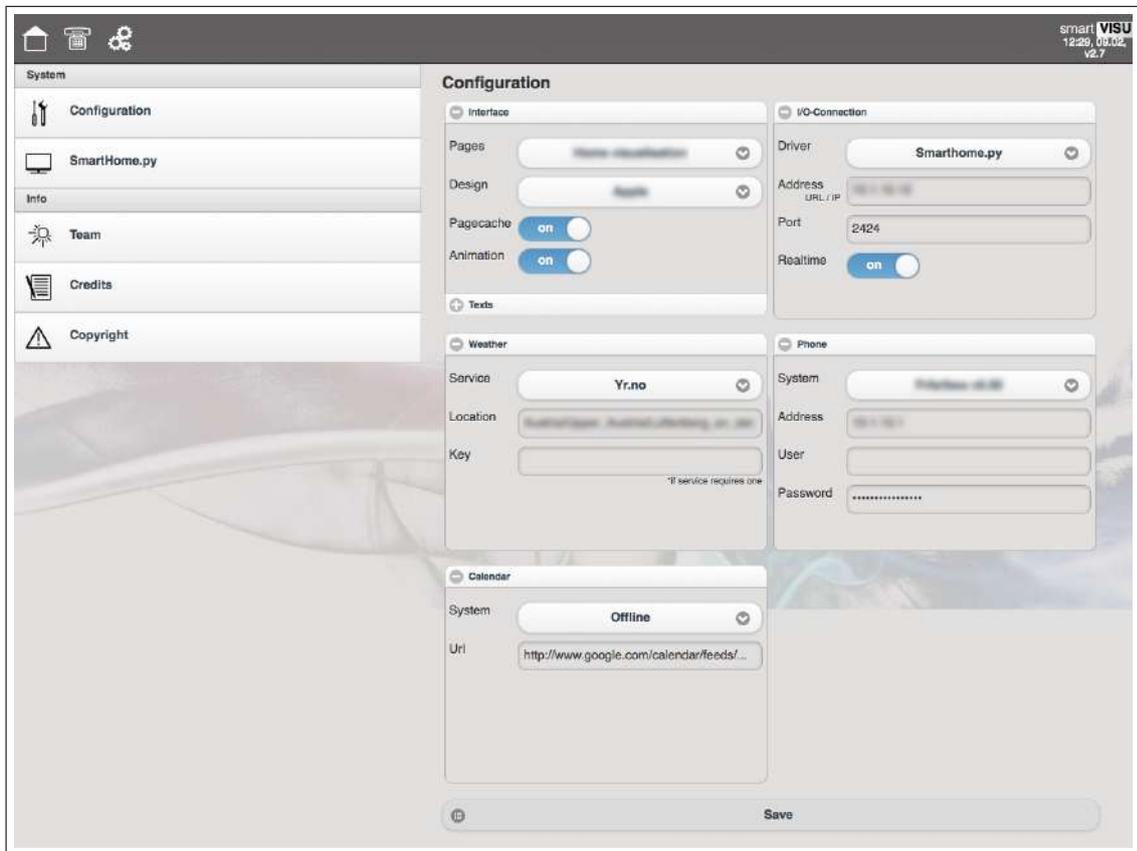


Abbildung 8.1.2: SmartVisu Configuration

Die fertige Visualisation unseres Projektes sehen Sie nun auf Abbildung 8.1.3.

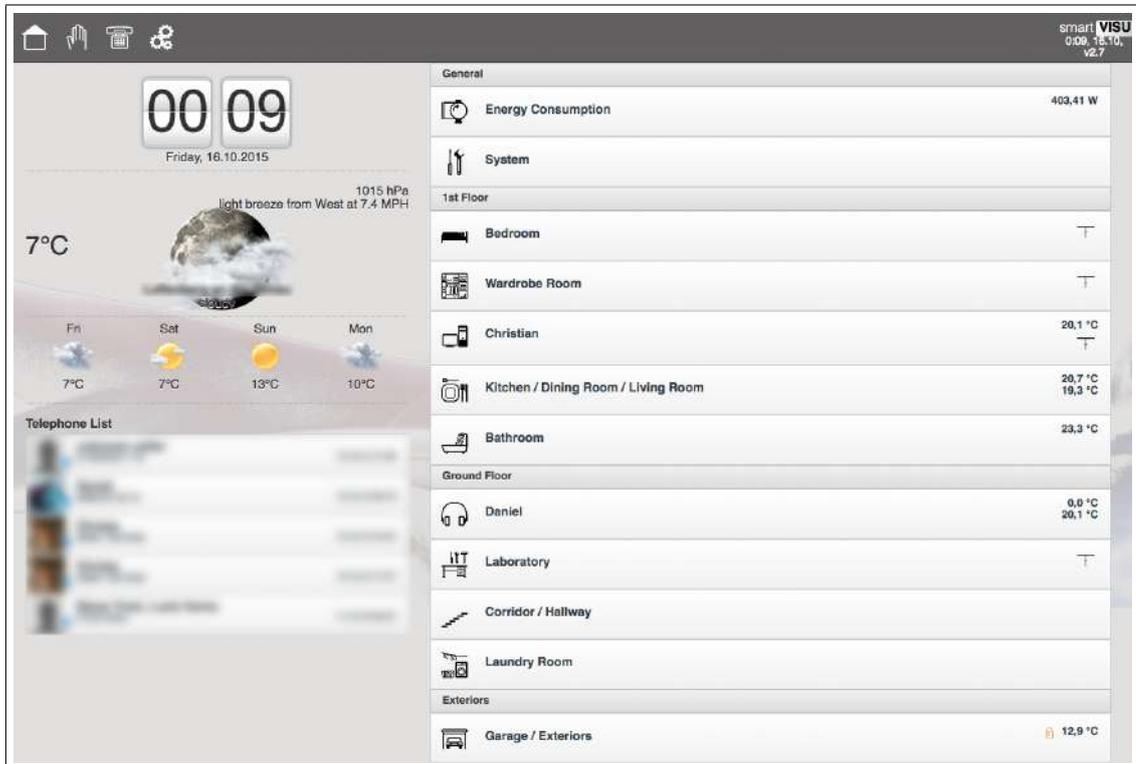


Abbildung 8.1.3: SmartVisu Front

8.2 CubeVision

Als zweite Visualisation steht uns die Software CubeVision der Firma BAB TECHNOLOGY zur Verfügung. Die Beschreibung auf der Homepage gibt einen Einblick in die Firmenphilosophie: »Früher waren anspruchsvolle Visualisierungen zeitaufwändig und dadurch kostenintensiv. Das ändert sich mit CUBEVISION von BAB TECHNOLOGIE. Der »Cube« generiert sich voll automatisch. Einfach nur die Gebäudestruktur anlegen, Gruppenadressen verknüpfen, das war's. Der Rest erledigt sich von selbst.« [25] Die CubeVision steht als App für iOS und Android kostenlos zum Download bereit. Abbildung 8.2.1 zeigt die BABtec CubeVision.

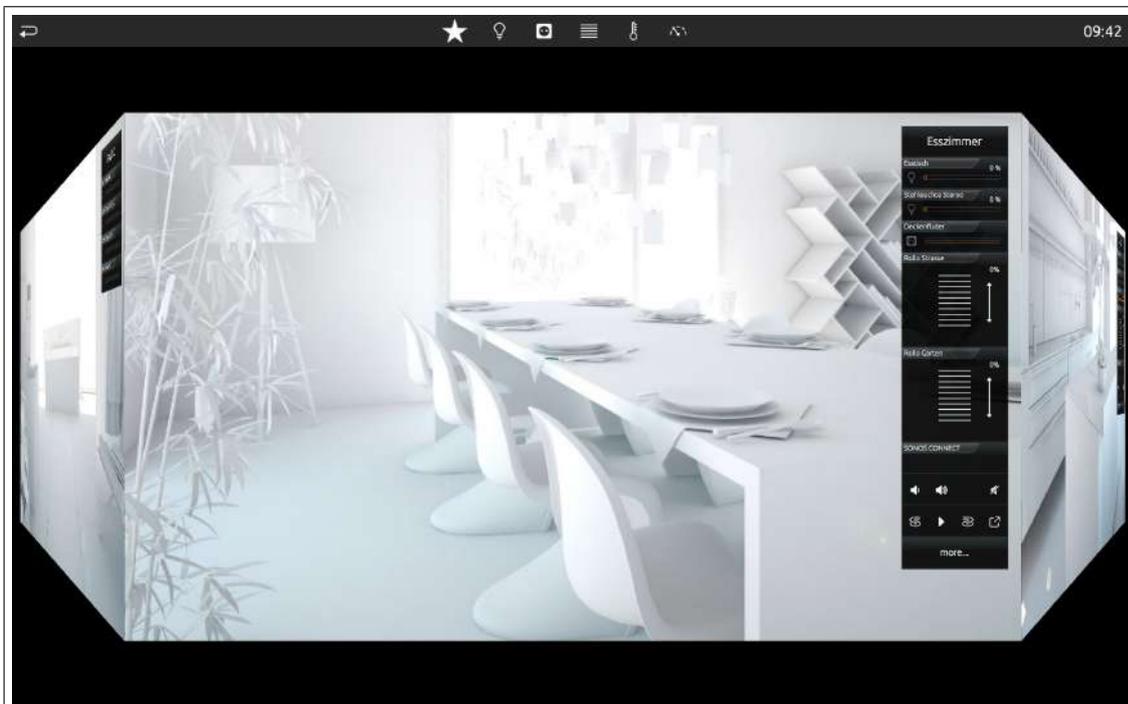


Abbildung 8.2.1: BABtec CubeVision

8.2.1 Konfiguration

Um in den CubeVision Editor, der aus dem Visualisierungsektor geöffnet wird, zu gelangen, muss die wie in Abbildung 8.2.2 rot markierte Schaltfläche ausgewählt werden.

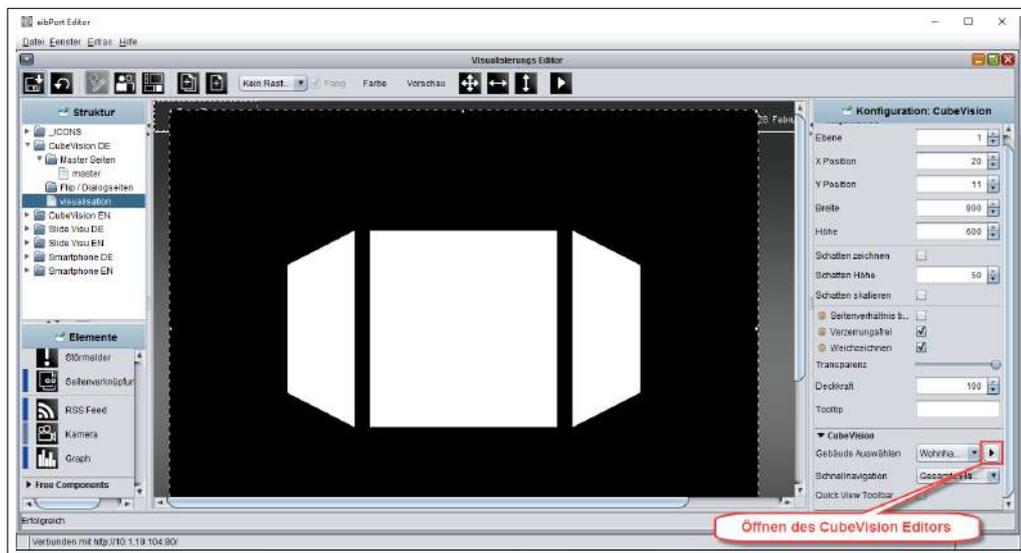


Abbildung 8.2.2: Öffnen des CubeVision Editors

8.2.2 Topologie

Anschließend folgt die Anbringung einer Topologie. Abbildung 8.2.3 zeigt die drei verschiedenen Strukturen der Topologie.

1. Ein Gebäude entspricht einer kompletten Struktur aus Geschossen (Ebenen), Räumen und Elementen. Es ist in der Topologie als Verzeichnisbaum hierarchisch aufgebaut und liegt dort also immer auf oberster Verzeichnisebene.
2. Ebenen stellen die Geschosse eines Gebäudes dar.
3. Die Räume sind als Objekte in den Ebenen-Verzeichnissen platziert. Elemente (Schalter/Dimmer, Jalousie, RTR, etc.) werden den Räumen jeweils über eine Art Liste im Parameterfenster hinzugefügt und verwaltet.

[26]



Abbildung 8.2.3: Anbringung der Topologie

8.2.3 Parametrierung der Werte

Das Element Werte/Diagramme bietet eine Vielzahl an Möglichkeiten, um Werte anzuzeigen, zu versenden, miteinander zu vergleichen oder deren Verläufe grafisch darzustellen.

1. **Beschreibung der Werte:** Wird in Standardansicht und Diagramm-Ansicht als Titel des Elements verwendet. Um das Element als Diagramm anzuzeigen wird das Häkchen bei »Als Diagramm anzeigen« gesetzt.
2. **Werte:** In diesem Feld werden die angelegten Werte bzw. ihre zugehörigen Gruppenadressen verwaltet.

[26]

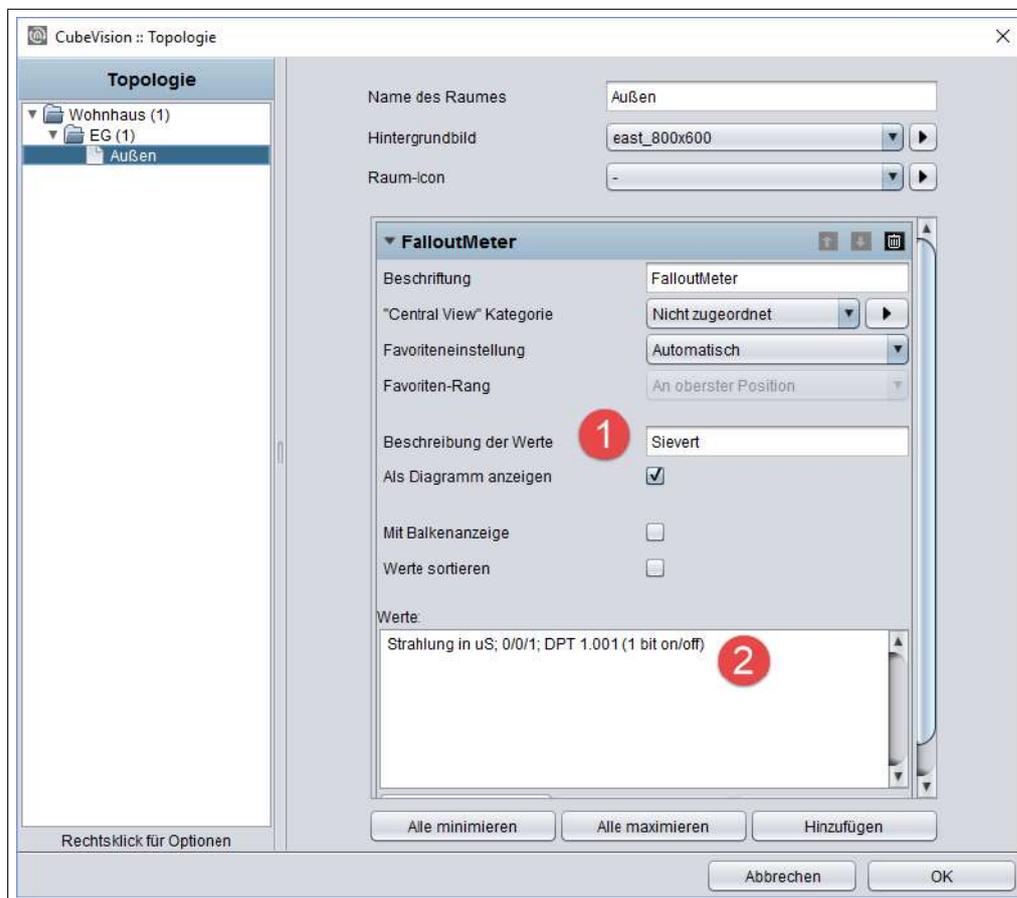


Abbildung 8.2.4: Wert/Diagramm Element

Abbildung 8.2.5 beschreibt die einzelnen Parameter.

1. **Name:** Erscheint sowohl in der Titelleiste des Bedienelements als auch im Diagramm (in Werteliste und auf der entsprechenden Schaltfläche).
2. **Adresse:** Gruppenadresse des darzustellenden Wertes. Die Gruppenadresse ist 0/0/1.
3. **Datentyp:** Der Datentyp, der den Wert beschreibt. Der Datentyp ist ein DPT 14 (früher auch EIS 9). Dieser gibt eine 16 Bit Gleitkommazahl an.
4. **Format:** Unter »Format« wird die Einheit der gewichteten Dosisleistung (Äquivalentdosisleistung) in $\frac{\mu\text{Sv}}{\text{h}}$ angegeben.

[26]



Abbildung 8.2.5: Parameterwerte einstellen

Abbildungsverzeichnis

3.2.1	Teviso RD2014 Blockdiagramm	6
4.2.1	TEC ohne Kondensatoren Ausgangsstufe	11
4.2.2	TEC ohne Kondensatoren Ausgangsdiagramm	12
4.2.3	TEC ohne Kondensatoren Rippelstrom	12
4.2.4	TEC mit Kondensatoren Ausgangsstufe	13
4.2.5	TEC mit Kondensatoren Ausgangsdiagramm	13
4.2.6	Teviso RD2014	15
4.3.1	FalloutMeter Schaltplan Seite 1	20
4.3.2	FalloutMeter Schaltplan Seite 2	21
4.3.3	FalloutMeter Schaltplan Seite 3	22
4.3.4	FalloutMeter Layout Top	23
4.3.5	FalloutMeter Layout Bot	24
4.3.6	FalloutMeter Lötstoppmaske Top	25
4.3.7	FalloutMeter Lötstoppmaske Bot	26
4.3.8	FalloutMeter Bestückungsplan Top	27
4.3.9	FalloutMeter Bestückungsplan Bot	28
4.3.10	FalloutMeter Bohrplan	29
4.3.11	FalloutMeter Sensor Schaltplan	32
4.3.12	FalloutMeter Sensor Layout Top	33
4.3.13	FalloutMeter Sensor Layout Bot	33
4.3.14	FalloutMeter Sensor Lötstoppmaske Top	34
4.3.15	FalloutMeter Sensor Lötstoppmaske Bot	34
4.3.16	FalloutMeter Sensor Bestückungsplan Top	35
4.3.17	FalloutMeter Sensor Bestückungsplan Bot	35
4.3.18	FalloutMeter Sensor Bohrplan	36
4.3.19	Platinen belichten	38
4.3.20	Platinen entwickeln	38
4.3.21	Platinen ätzen	39
4.3.22	Lötstoppmaske aufbringen	40
4.3.23	Bohrplan Legende	40
4.3.24	Platinen Fertigen Löten	41
5.1.1	3D-Modell Gehäuse	44
5.1.2	3D Modell der Gewindestangen für die Sensorplatine	45

5.2.1	Blechausschnitt	47
5.2.2	Gewindestangen für die Sensorplatine	48
5.2.3	Halterung ohne Platine	49
5.2.4	Befestigung des Kühlkörpers und des Peltier Elementes	50
5.2.5	Lüftungsschlitze	51
5.2.6	Rahmen mit Gewindebohrungen	52
5.2.7	verzinktes Gehäuseunterteil	53
5.2.8	Befestigung des Kühlkörpers	54
5.2.9	verzinktes Gehäuseoberteil	55
5.2.10	Befestigung des Sensors	56
5.2.11	Wandhalterung des Gehäuses	57
6.1.1	Peltier-Element Sprungantwort	61
7.1.1	KNX-Topologie	64
7.1.2	KNX TP Telegrammaufbau	65
7.1.3	Weinzierl BAOS-Modul 820	66
7.1.4	ETS5 Datenpunkte	68
7.1.5	ETS5 Parameter	69
7.2.1	GSM Funktionsweise	70
7.2.2	BABtec eibPort mit GSM	71
7.2.3	Job-Editor	72
7.2.4	SMS Sender Einstellungen	72
7.2.5	Werte senden	73
7.3.1	Push-Meldung am Lock-Screen	74
7.3.2	HTTP-Request Job für Push-Benachrichtigung 1	75
7.3.3	HTTP-Request Job für Push-Benachrichtigung 2	76
8.1.1	SmartControl	84
8.1.2	SmartVisu Configuration	93
8.1.3	SmartVisu Front	94
8.2.1	BABtec CubeVision	95
8.2.2	Öffnen des CubeVision Editors	96
8.2.3	Anbringung der Topologie	96
8.2.4	Wert/Diagramm Element	97
8.2.5	Parameterwerte einstellen	98

Tabellenverzeichnis

4.3.1	FalloutMeter Stückliste	17
4.3.1	FalloutMeter Stückliste	18
4.3.1	FalloutMeter Stückliste	19
4.3.2	FalloutMeter Sensor Stückliste	31
6.1.1	ATmega328P Fuses	59
7.1.1	verwendete Gruppenadressen	67
7.1.2	verwendete Parameterbytes	67
A.2.1	Kostenaufstellung FalloutMeter	XVII
A.2.1	Kostenaufstellung FalloutMeter	XVIII
A.2.1	Kostenaufstellung FalloutMeter	XIX
A.2.1	Kostenaufstellung FalloutMeter	XX
A.3.1	Kostenaufstellung Visualisierung	XX

Literaturverzeichnis

- [1] Strahlenfrühwarnsystem. <https://de.wikipedia.org/wiki/Strahlenfr%C3%BCwarnsystem>, Oktober 2015.
- [2] Messwerte des Strahlenfrühwarnsystems. https://www.bmlfuw.gv.at/umwelt/strahlen-atom/strahlenschutz/strahlen-warn-system/messwerte_aktuell.html, Dezember 2015.
- [3] Bundesamt für Strahlenschutz. Ionisierende Strahlung. http://www.bfs.de/DE/themen/ion/einfuehrung/einfuehrung_node.html, April 2015.
- [4] Auswirkungen der ionisierenden Strahlung. <http://www.netdokter.de/magazin/was-bewirkt-radioaktivitaet-im-koerper/>, Oktober 2015.
- [5] Texas Instruments LMR14050. <http://www.ti.com/lit/ds/symlink/lmr14050.pdf>, Dezember 2015.
- [6] Berechnung des Rippelstromes am TEC (Seite 10). <http://www.ti.com/lit/ds/symlink/drv592.pdf>, Dezember 2015.
- [7] TEC Handhabungshinweise (Seite 1). http://www.eppich.de/astro/Handhabungshinweise_german.pdf, Dezember 2015.
- [8] Peltier-Element Datenblatt. http://www.produktinfo.conrad.com/datenblaetter/175000-199999/182597-da-01-en-PELTIER_ELEMENT_QC_31_1_4_3_7M.pdf, März 2016.
- [9] Aosong AM2321 i2c temperatur- und luftfeuchtigkeitssensor. http://akizukidenshi.com/download/ds/aosong/AM2321_e.pdf, Dezember 2015.
- [10] Peltier-handhabungshinweise. http://eppich.de/astro/Handhabungshinweise_german.pdf, März 2016.
- [11] Teviso Sensor Technologies Ltd. Teviso Sensor Technologies Ltd. <http://www.teviso.com/>, März 2015.
- [12] RD2014 Sensor für Beta- Gamma- und Röntgenstrahlung Datenblatt. <http://www.teviso.com/file/pdf/rd2014-dataspecification.pdf>, März 2016.
- [13] Weinzierl KNX BAOS Modul 820 Software. http://www.weinzierl.de/download/development/82x/Download/KnxBAOS_Demo_Source.zip, März 2016.

- [14] Grundlagenwissen zum KNX Standard. <http://www.knxaustria.at/2016/index.php/downloads/category/12-anleitungen?download=43:knx-basics-de>, März 2016.
- [15] Weinzierl Engineering GmbH. Weinzierl KNX BAOS Modul 820. <http://www.weinzierl.de/index.php/de/alles-knx1/knx-module/knx-baos-modul-820>, März 2015.
- [16] Weinzierl Engineering GmbH Downloads. <http://www.weinzierl.de/index.php/en/download-baos>, März 2016.
- [17] Grundlagen Mobilfunk. <http://www.elektronik-kompodium.de/sites/kom/0406221.htm>, Februar 2016.
- [18] GSM Funktionsweise. <http://informationszentrum-mobilfunk.de/wie-funktioniert-der-gsm-mobilfunkstandard#header>, Februar 2016.
- [19] eibPort Editor. http://bab-tec.de/index.php/eibport_v3_de.html?file=tl../eibPort, März 2016.
- [20] GROWL The Ultimate Notification System for OS X. <http://growl.info/>, März 2016.
- [21] Prowl iOS App Store. <https://itunes.apple.com/us/app/prowl-easy-push-notifications/id320876271?mt=8>, März 2016.
- [22] BABtec Wiki Prowl Push-Benachrichtigung. http://wiki.bab-tec.de/wiki/Deutsch:Jobs#Push_Notification_an_iOS_APP_PROWL, März 2016.
- [23] Prowl Website. <https://www.prowlapp.com/>, März 2016.
- [24] Raspbian Download. https://downloads.raspberrypi.org/raspbian_latest, März 2016.
- [25] CubeVision. CubeVision. (<http://www.cubevision.info/>), März 2016.
- [26] Berker. Grundlagenwissen zur CubeVision. https://www.google.at/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiBnN2_v9DLAhUGYJoKHRbmArAQFggbMAA&url=http%3A%2F%2Fwww.berker.de%2Ffiles%2Fdownload%2F0%2F3944946_1%2F0%2F75710004-75710036__IP-CONTROL__CUBEVISION__DE__2014-04.PDF&usg=AFQjCNE8sigJoRK7RTQAJMoRrgDDB15Vkf&sig2=Ijq6faaM_GqFiDkHQLBR5g&bvm=bv.117218890,d.bGs&cad=rja, April 2014.

A Anhang

A.1 Pflichtenheft



PFLICHTENHEFT

Fallout Meter - stationäres Radioaktivitätsmessgerät

Autor:

Mihajlo KREČAR, Christian HAUN

Betreuungslehrer:

Prof. Dipl.-Ing. Dr. Christoph LACKNER

8. März 2016

Inhaltsverzeichnis

1	Vorbemerkungen	3
1.1	Zielsetzung	3
2	Messung der Strahlung	4
2.1	Halbleiterdetektor	4
2.2	Strahlungssensoren der Firma Teviso	4
3	KNX Busverbindung	6
4	GSM Anbindung	6
5	Visualisierung	7
6	Schlussbemerkungen	7
7	Meilensteine	8

1 Vorbemerkungen

Im Land Österreich gibt es Messstationen zum Messen von ionisierender Strahlung. [1]

Wenn sich im Atomkern eines Stoffes eine für die Protonenzahl geeignete Anzahl von Neutronen befindet, ist der Kern in seinem Grundzustand stabil. Weicht der Wert der Neutronen jedoch ab, so wird dieser Kern instabil und es setzt somit ein radioaktiver Zerfall ein.

Die wichtigsten Zerfallsarten sind hierbei: α -, β - und γ -Zerfall.

Beim α -Zerfall werden Heliumkerne ausgesendet, beim β -Zerfall Elektronen (selten auch aus Positronen) und beim γ -Zerfall Photonenstrahlung mit hoher Frequenz. [2]

Wenn der menschliche Körper ionisierende Strahlung absorbiert, entstehen dadurch im Körper aus neutralen Teilchen elektrisch geladene Ionen. Diese Ionen sind sehr reaktionsfreudig, wollen also eine stabile Verbindung eingehen. Die dabei entstehenden Verbindungen können für den Körper gefährlich sein.

Dies kann dazu führen, dass Enzyme ihre Funktion einbüßen, oder sogar ganze Zellen zerstört werden. Die Strahlung kann auch zur Beschädigung des Erbmaterials führen, was wiederum Krebs auslösen kann. [3]

1.1 Zielsetzung

Die Entwicklung und der Bau eines halbleiterbasierenden Strahlenmessgerätes mit KNX-Anbindung. Desweiteren sollen die benötigten KNX-Komponenten eingebaut werden und eine Alarmierung, z.B. mittels GSM, realisiert werden.

2 Messung der Strahlung

Die Messhardware besteht prinzipiell aus einem Halbleiterdetektor und einem Mikrocontroller, der diesen auswertet.

2.1 Halbleiterdetektor

Bei diesem Projekt soll ein Halbleiterdetektor, anstatt eines Geiger-Müller-Zählrohres verwendet werden.

2.2 Strahlungssensoren der Firma Teviso

Durch Internetrecherche sind wir auf die schweizer Firma Teviso und deren Sensoren gestoßen. Die Firmenbezeichnung klingt vielversprechend:

»Teviso Sensor Technologies Ltd. entwickelt Strahlungssensoren auf Halbleiter Basis, die für die Messung von Beta-, Gamma- und Röntgenstrahlung bestimmt sind. Diese Sensoren sind kostengünstig und dank ihrer Robustheit geeignet, in vielen Anwendungen das traditionelle Geiger-Müller Zählrohr abzulösen.

Sensoren für radioaktive Strahlung in Festkörper Technologie Ltd. werden in der Forschung bereits seit mehreren Jahren eingesetzt. Sie sind relativ groß und komplex und vor allem sehr teuer. Mit den neuen Produkten ist es Teviso gelungen, kleine und vor allem kostengünstige Sensoren zu entwickeln, die primär als Herzstück in batteriebetriebenen Strahlungsmessgeräten verwendet werden können.

Teviso hat sich zum Ziel gesetzt, innovative Produkte auf hohem Stand der Technik zu entwickeln und zu produzieren. Der Technologiestandort Schweiz bietet für TEVISO ideale Voraussetzungen, den hohen Ansprüchen an Präzision und Zuverlässigkeit gerecht zu werden.« [4]

Abbildung 1 zeigt den Strahlungssensor, auf Abbildung 2 sehen Sie das dazugehörige Blockdiagramm.



Abbildung 1: Abbildung des RD2014 [5]

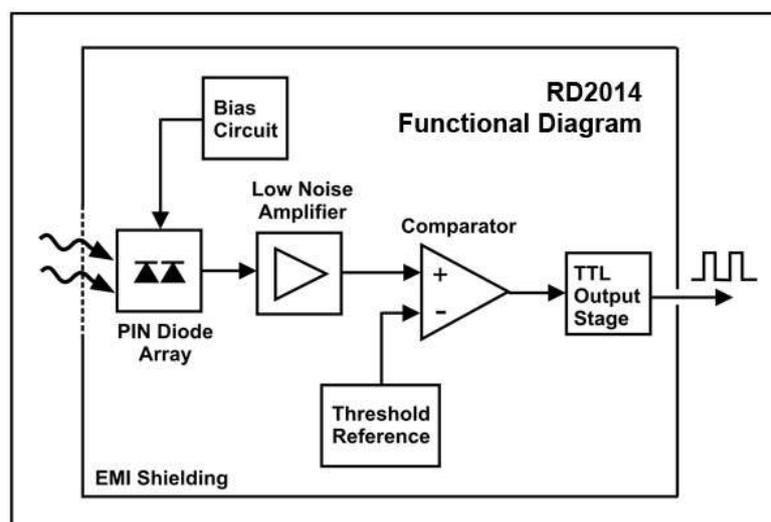


Abbildung 2: Blockdiagramm des RD2014 [5]

3 KNX Busverbindung

Um eine effiziente und zuverlässige Busverbindung zu gewährleisten, gibt es verschiedene Möglichkeiten zur KNX-Anbindung.

Die bevorzugte Methode stellt eine der vielen KNX-Schnittstellen der Firma Weinzierl dar. [6]

Weinzierl bietet eine breite Palette an verschiedensten KNX-Anbindungen, welche Neuentwicklung eines KNX-fähigen Gerätes zu erleichtern. Das KNX BAOS Modul 820 eignet sich für unsere Anwendung besonders, da es auf der Sekundärseite mittels UART (FT2.1) kommuniziert, ideal zur Anbindung an den Mikrocontroller. Die KNX-Schnittstelle sehen Sie auf Abbildung 3.

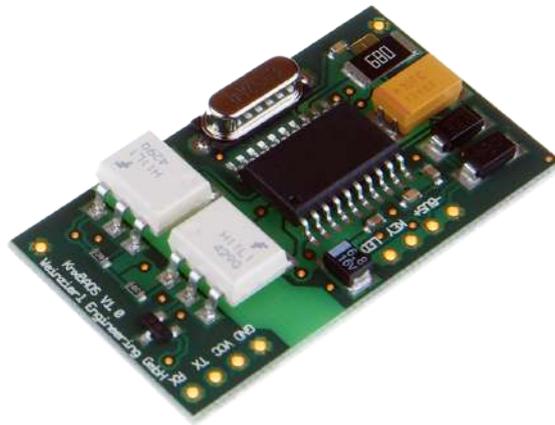


Abbildung 3: Weinzierl KNX BAOS Modul 820 [7]

4 GSM Anbindung

Um eine möglichst langlebige und erweiterbare Lösung zu erreichen, haben wir uns für ein KNX-GSM-Gateway anstatt einer speziellen GSM-Anbindung, welches nur für die Messstation ausgelegt wäre entschieden. Somit ist es auch möglich, weitere KNX-Komponenten mit einer GSM-Anbindung auszustatten.

5 Visualisierung

Die Visualisierung soll mit der Software SmartHome.py erfolgen, welche unter der GNU General Public License lizenziert ist. Sie bietet eine grafische Oberfläche, welche sich SmartVisu nennt. Diese sehen Sie auf Abbildung 4.

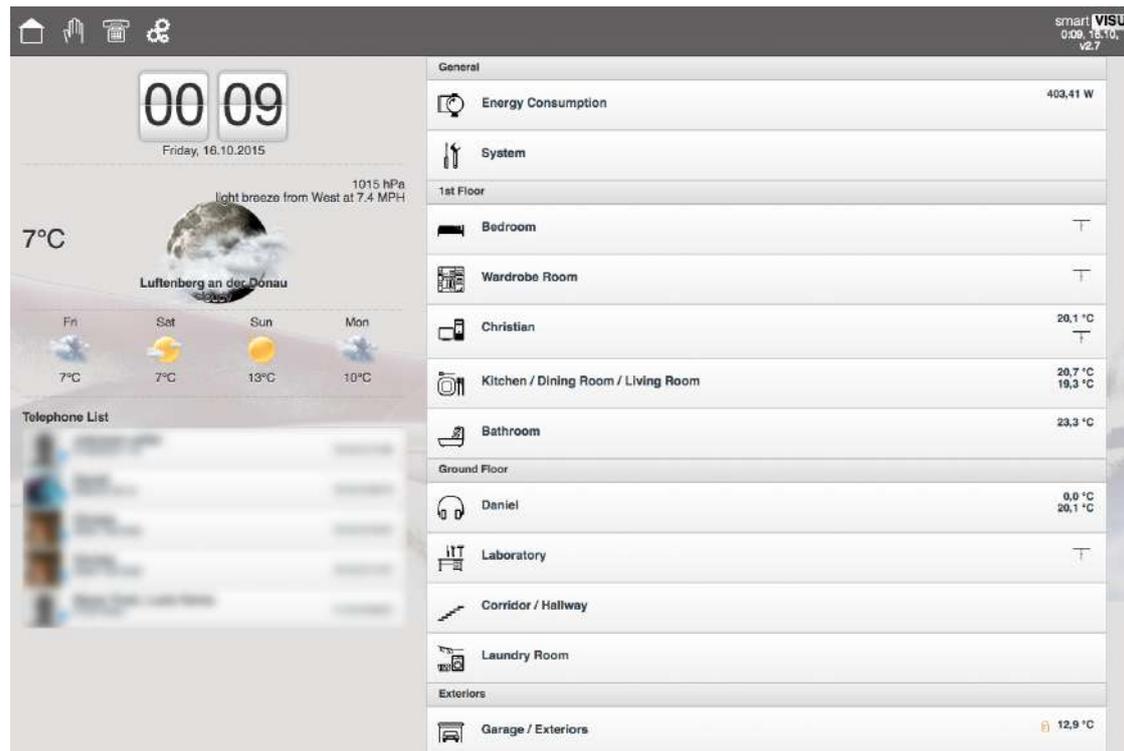


Abbildung 4: SmartVisu [8]

SmartHome.py kann auf verschiedenen Servern installiert werden. In unserem Fall kommt ein Raspberry Pi zum Einsatz, da dieser besonders kostengünstig und stromsparend ist. SmartHome.py kann auch verwendet werden, um verschiedene Logiken für die KNX-Anlage zu programmieren.

6 Schlussbemerkungen

Atomare Gefährdungen sind in der heutigen Zeit eine ernst zu nehmende Bedrohung. Gerade hier ist es notwendig, immer den Überblick über die Lage zu behalten - dies kann sehr gut durch eine kontinuierliche Messung erreicht werden.

7 Meilensteine

Nr.	Datum	Beschreibung	Bearbeiter
0	30.10.2015	Gesamtkonzept erstellen	Krečar, Haun
1	27.11.2015	Hardwareentwicklung (Schaltplan, tlw. Simulation sofern notwendig, Layout, Fertigung Prototypen und Produkt)	Krečar, Haun
2	01.01.2016	Programmierung	Haun
3	02.02.2016	Testen, Kalibrieren	Krečar, Haun
4	25.03.2016	Visualisierung und Inbetriebnahme	Krečar, Haun

Literatur

- [1] Strahlenfrühwarnsystem. <https://de.wikipedia.org/wiki/Strahlenfr%C3%BChwarnsystem>, Oktober 2015.
- [2] Bundesamt für Strahlenschutz. Ionisierende Strahlung. http://www.bfs.de/DE/themen/ion/einfuehrung/einfuehrung_node.html, April 2015.
- [3] Auswirkungen der ionisierenden Strahlung. <http://www.netdokter.de/magazin/was-bewirkt-radioaktivitaet-im-koerper/>, Oktober 2015.
- [4] Teviso Sensor Technologies Ltd. Teviso Sensor Technologies Ltd. <http://www.teviso.com/>, März 2015.
- [5] Teviso Sensor Technologies Ltd. RD2014 Sensor für Beta- Gamma- und Röntgenstrahlung. <http://www.teviso.com/de/produkte/strahlungssensor-rd2014.htm>, Juni 2015.
- [6] Weinzierl Engineering GmbH. Weinzierl Engineering GmbH. <http://www.weinzierl.de/index.php/de/>, März 2015.
- [7] Weinzierl Engineering GmbH. Weinzierl KNX BAOS Modul 820. <http://www.weinzierl.de/index.php/de/alles-knx1/knx-module/knx-baos-modul-820>, März 2015.
- [8] SmartHome.py. <http://mknx.github.io/smarthome/>, Oktober 2015.

A.2 Kostenaufstellung FalloutMeter

Tabelle A.2.1: Kostenaufstellung FalloutMeter

Beschreibung	Wert	Lieferant	Preis
Control-Board			
C1	100 nF	Mouser	0,457 €
C2	47 nF	Mouser	0,38 €
C3	100 nF	Mouser	0,457 €
C4	100 nF	Mouser	0,457 €
C5	100 nF	Mouser	0,457 €
C6	100 nF	Mouser	0,457 €
C7	100 nF	Mouser	0,457 €
C8	4.7 μ F	Mouser	0,567 €
C9	4.7 μ F	Mouser	0,567 €
C10	10 μ F	Mouser	0,624 €
C11	4.7 μ F	Mouser	0,567 €
C12	4.7 μ F	Mouser	0,567 €
C13	10 μ F	Mouser	0,624 €
C14	100 nF	Mouser	0,457 €
C15	100 nF	Mouser	0,457 €
C17	100 nF	Mouser	0,457 €
C20	100 nF	Mouser	0,457 €
C_BOOT	100 nF	Mouser	0,457 €
C_IN	4,7 μ F	Mouser	0,567 €
C_LOAD	15 pF	Mouser	0,567 €
C_OUT	180 μ F	Mouser	1,35 €
C_SS	22 nF	Mouser	0,265 €
D1	1N4148	Conrad	0,21 €
D2	LD117ADT-TR	Mouser	0,768 €
D3	BAS40	Mouser	0,344 €

Tabelle A.2.1: Kostenaufstellung FalloutMeter

Beschreibung	Wert	Lieferant	Preis
D4	BAS40	Mouser	0,344 €
F1	0,6 A/T	Mouser	0,877 €
IC1	ATmega328P	Mouser	3,23 €
IC2	LMR14050	Mouser	4,20 €
IC3	IR2104S	Mouser	1,24 €
IC4	IR2104S	Mouser	1,24 €
L1	8,2 μH	Mouser	1,45 €
L2	330 μH	Mouser	2,29 €
L3	330 μH	Mouser	2,29 €
LED1	KNX-LED	Mouser	0,338 €
Q1	N-HEXFET-2D_SOT232-	Mouser	0,86 €
Q2	N-HEXFET-2D_SOT232-	Mouser	0,86 €
Q3	N-HEXFET-2D_SOT232-	Mouser	0,86 €
Q4	N-HEXFET-2D_SOT232-	Mouser	0,86 €
QG1	7,3728 MHz	Mouser	1,97 €
R1	10 k Ω	Mouser	0,091 €
R2	4,7 k Ω	Mouser	0,317 €
R3	4,7 k Ω	Mouser	0,317 €
R4	Varistor 1206	Mouser	0,70 €
R5	22 Ω	Mouser	0,317 €
R6	22 Ω	Mouser	0,317 €
R7	100 k Ω	Mouser	0,317 €
R8	22 Ω	Mouser	0,317 €
R9	22 Ω	Mouser	0,317 €
R100	0 Ω	Mouser	0,317 €
R_FBB	17,8 k Ω	Mouser	0,443 €
R_FBT	100 k Ω	Mouser	0,317 €
R_LOAD	10 k Ω	Mouser	0,091 €
R_T	84,5 k Ω	Mouser	0,443 €

Tabelle A.2.1: Kostenaufstellung FalloutMeter

Beschreibung	Wert	Lieferant	Preis
S1	KNX-Button	Mouser	0,788 €
SV1	Pinhead	Mouser	1,20 €
U3	Aosong AM2321	Aliexpress	3,05 €
X1.1	MKDSN1,5/3-5,08	Conrad	0,53 €
X1.2	MSTB1,5/3-5,08	Conrad	0,53 €
X2.1	MKDSN1,5/2-5,08	Conrad	0,62 €
X2.2	MSTB1,5/2-5,08	Conrad	0,62 €
X3	243-211 KNX-Klemme	Conrad	0,10 €
X4.1	MKDSN1,5/2-5,08	Conrad	0,62 €
X4.2	MSTB1,5/2-5,08	Conrad	0,62 €
Peltier-Element	$U_N = 3,8 V$, $I_N = 3,7 A$	Conrad	28,99 €

Sensor-Board

C1	$100 \mu F$	Mouser	0,434 €
R1	150Ω	Mouser	0,443 €
U\$1	Teviso RD2014	Teviso	96,00 €
X1.1	MKDSN1,5/3-5,08	Mouser	0,53 €
X1.2	MSTB1,5/3-5,08	Mouser	0,53 €

Gehäuse

Baustahl		Eisen Rath	20 €
Kühlkörper		Conrad	9,39 €
M3 Schrauben		Bauhaus	3,00 €
M3 Gewindebolzen		Bauhaus	3,00 €

Sonstiges

Lötzinn	bleifrei	Conrad	13,99 €
Platinenbasismaterial	fotobeschichtet	Conrad	4,59 €
Lötstopplaminat		Octamex	2,278 €

Tabelle A.2.1: Kostenaufstellung FalloutMeter

Beschreibung	Wert	Lieferant	Preis
Verbrauchsmaterial		Conrad	5 €
Summe:			223,39 €

A.3 Kostenaufstellung Visualisierung

Tabelle A.3.1: Kostenaufstellung Visualisierung

Beschreibung	Wert	Lieferant	Preis
Visualisierung			
Raspberry Pi B		Conrad	36,00 €
4 GB SD-Karte SCL		Conrad	43,00 €
Summe:			79,00 €

B weitere Dokumente

B.1 Projektstagebücher

B.1.1 Projektstagebuch von Mihajlo Krečar

Tag	Datum	Beschreibung	Zeit
September 2015			
Sa	12.09.2015	Besprechung der Arbeit; Planung des Gehäuses begonnen	5 h
Mi	16.09.2015	Erstellen des Diplomarbeitsantrags	2 h
Oktober 2015			
November 2015			
Fr	13.11.2015	Erstellen des Diplomarbeitstext für die Maturazeitung	1 h
Mi	18.11.2015	Entwurf des Layoutdesigns; Gespräch mit Prof. Dr. Dipl. Ing. Christoph Lackner zwecks der Spannungsregelung	4 h
Fr	27.11.2015	Besprechung und Planung vom Gehäuse, Konstanttemperaturregelung	4 h
Dezember 2015			
Do	10.12.2015	Installation und Einführung von SketchUp für die Gehäusemodellierung	3 h
Mo	14.12.2015	Gespräch mit Prof. Dr. Dipl. Ing. Christoph Lackner zwecks Fortschritt	1 h
Mi	16.12.2015	Gehäusemodellierung mit Solid Edge ST4	6 h
So	20.12.2015	Installation und Kennenlernen des Programms Autodesk Inventor 2016	3 h

Tag	Datum	Beschreibung	Zeit
So	27.12.2015	Einführung von Autodesk Inventor 2016 mit Hilfe von YouTube Tutorials	4 h
Di	29.12.2015	Beginn der 3D Modellierung vom oberen Gehäuseteil	5 h
Mi	30.12.2015	3D Modellierung vom oberen Gehäuseteil	4 h
Januar 2016			
Di	12.01.2016	Erstellen des Diplomarbeitsplakates für den Tag der offenen Tür	1 h
Fr	15.01.2016	Änderungen der 3D Modellierung vom oberen Gehäuseteil	5 h
So	17.01.2016	Kennenlernen der Grundkenntnisse in LaTeX	3 h
Di	19.01.2016	Beginn mit der Dokumentation	3 h
Fr	22.01.2016	Dokumentieren	2 h
So	24.01.2016	Fertigstellung der 3D Modellierung vom oberen Gehäuseteil	3 h
Mi	27.01.2016	Zusammenschneiden und Feilen von Vorder- und Hinterteil vom oberen Gehäuseteil	3 h
Februar 2016			
Di	02.02.2016	Blechausschnitt für das Sensorfenster, Feilen	1 h
Fr	05.02.2016	Kauf der Gewindebolzen und Zurechtschneiden mittels Flex	3 h
So	07.02.2016	Hinzufügen von Gewindebolzen für die Verschraubung der Sensorplatine	3 h
Di	09.02.2016	Dokumentieren	3 h
Do	11.02.2016	Dokumentieren	2 h
Fr	12.02.2016	Änderungen und Anpassungen vom 3D Modell in Autodesk Inventor	6 h
Mo	15.02.2016	Dokumentieren	3 h
Di	16.02.2016	Skizzen und Überlegungen für den unteren Gehäuseteil	3 h
Mi	17.02.2016	Absprache der weiteren Schritte, Skizze/Planung für das untere Gehäuseteil	6 h

Tag	Datum	Beschreibung	Zeit
Do	18.02.2016	Dokumentieren	3 h
Fr	19.02.2016	Dokumentieren	4 h
So	21.02.2016	Besprechung zwecks Weiterbau vom Gehäuse, Planänderungen	2 h
Mo	22.02.2016	Änderung und Verbesserung der Dokumentation	3 h
Di	23.02.2016	Beginn und Überlegungen für das Video für den Bosch Preis	1 h
Do	25.02.2016	Einpressen der Lüftungsschlitze, Bohrung für die KNX Zuleitung, Einkauf diverser Materialien	5 h
Fr	26.02.2016	Dokumentieren	2 h
So	28.02.2016	Visualisierung und GSM-Alarmierung mit dem Bab-Tec eibPort	7 h
Mo	29.02.2016	Dokumentieren	3 h
März 2016			
Mo	07.03.2016	Änderung und Verbesserung der Dokumentation	3 h
Di	08.03.2016	Anpassung vom Gehäusemodell, Änderung der Dokumentation	6 h
Mi	09.03.2016	Aktualisieren des Projektstagebuches, Dokumentieren, Installation von Adobe Indesign für das Flyer	4 h
Do	10.03.2016	Dokumentieren	3 h
Sa	12.03.2016	Dokumentieren, Versuch zum Erstellen einer U3D-Datei in Autodesk Inventor 2016	6 h
Fr	18.03.2016	Dokumentieren	3 h
So	20.03.2016	Dokumentieren	3 h
Mo	21.03.2016	Dokumentieren	2 h
Mi	23.03.2016	Dokumentieren	4 h
Fr	25.03.2016	Korrekturlesen	3 h
Sa	26.03.2016	Korrekturlesen	2 h
So	27.03.2016	Korrekturlesen	3 h
Mo	28.03.2016	Beendigung der Dokumentation für Buchdruck	6 h

Tag	Datum	Beschreibung	Zeit
April 2015			
Mo	11.04.2016	Designen des Flyers	6 h
Do	14.04.2016	Designen des Flyers	5 h
Di	19.04.2016	Erstellen eines Kurzfilms der Diplomarbeit für den Technik fürs Leben-Preis	3 h
Fr	22.04.2016	Erstellen eines Kurzfilms der Diplomarbeit für den Technik fürs Leben-Preis	6 h
Summe:			190 h

B.1.2 Projektstagebuch von Christian Haun

Tag	Datum	Beschreibung	Zeit
September 2015			
Sa	05.09.2015	Brainstorming, Internetrecherche	4 h
So	06.09.2015	Vorbereitung der LaTeX-Diplomarbeitvorlage	6 h
Mo	07.09.2015	Anfang der Dokumentation	5 h
Di	08.09.2015	Internetrecherche	5 h
Mi	09.09.2015	Dokumentation	6 h
Sa	12.09.2015	Beschaffung der Materialien zum Platinen ätzen	2 h
So	13.09.2015	Erste Konzepte zur Schaltplanentwicklung realisieren	5 h
Fr	18.09.2015	Bestellung der Komponenten: Halbleiterdetektor, Mikrocontroller	6 h
Sa	19.09.2015	Aufsetzen des Raspberry Pi mit SmartHome.py, Eintragen der verwendeten Gruppenadressen, Dokumentieren	10 h
So	20.09.2015	Installieren von Smarthome.py sowie Konfiguration der SmartVisu, Test der Visualisierung	8 h
Mo	21.09.2015	Dokumentieren von Smarthome.py	6 h
Do	24.09.2015	Dokumentieren der SmartVisu	5 h
Fr	25.09.2015	Testaufbau der Sensorschaltung, Schreiben eines Testprogrammes in C	5 h
So	27.09.2015	Kontrolle der Testschaltung; Auswertung der Ergebnisse	4 h
Oktober 2015			
Fr	02.10.2015	Einlesen in die Produktbeschreibung des BAOS Modul 820 der Firma Weinzierl, Herunterladen des Demoprogrammes in Atmel Studio	4 h
Sa	03.10.2015	Test der KNX-Schnittstelle BAOS Modul 820 der Firma Weinzierl, schreiben einer eigenen Software	4 h
So	04.10.2015	Software für das BAOS Modul 820 der Firma Weinzierl verbessern; Tests durchführen	3 h

Tag	Datum	Beschreibung	Zeit
Di	06.10.2015	Erstellen der nötigen Eagle-Librarys	4 h
Do	08.10.2015	Vollenden der Eagle-Librarys, zeichnen des Schaltplans in Eagle	6 h
Fr	09.10.2015	Vollenden des Schaltplans, Gespräch mit Prof. Dr. Dipl-Ing. Christoph Lackner zwecks der Platine	5 h
So	11.10.2015	Beginn des Layoutdesigns	6 h
Sa	12.10.2015	Vollendung des des Layoutdesigns	6 h

November 2015

Mo	02.11.2016	Dokumentieren	5 h
Di	03.11.2016	Einlesen in das bestehende Programm für das Weinzierl Development-Board	5 h
Mi	04.11.2016	Dokumentieren	5 h
Sa	07.11.2016	Programmenwicklung	4 h
So	08.11.2016	Berechnungen	3 h
Mo	09.11.2016	Dokumentieren	4 h
Mi	11.11.2016	Dokumentieren	5 h
Fr	20.11.2016	Dokumentieren	3 h
So	28.11.2015	Änderung des Schaltplanes sowie des Layouts aufgrund der Anpassung der Kühlung, bzw. Heizung	3 h

Dezember 2015

Sa	05.12.2015	Vollendung des Layouts	5 h
Di	08.12.2015	Ätzen der Platine	8 h
Sa	12.12.2015	Dokumentieren	6 h
So	13.12.2015	Dokumentieren diverser Berechnungen, auch Berechnung der Zuleitungsquerschnitte	3 h
Di	15.12.2015	Programmänderungen	4 h
Mi	16.12.2015	Dokumentieren	5 h
Fr	19.12.2015	Kauf weiterer Komponenten	2 h
Sa	20.12.2015	Arbeiten am Raspberry Pi und an der SmartVisu	5 h

Tag	Datum	Beschreibung	Zeit
So	21.12.2015	Arbeiten am BabTec eibPort, Kauf einer geeigneten SIM-Karte zwecks GSM-Anbindung	6 h
Januar 2016			
Fr	1.01.2016	Dokumentieren	5 h
Sa	2.01.2016	Arbeiten am Programm	5 h
So	3.01.2016	Arbeiten am Programm	4 h
Sa	09.01.2016	Arbeiten am Programm, Funktionstests	6 h
So	10.01.2016	Arbeiten an der SmartVisu	3 h
Mo	11.01.2016	Testaufbauten mit Peltier-Element testen, Dokumentieren	6 h
Di	12.01.2016	Dokumentieren	5 h
Di	18.01.2016	Dokumentieren	3 h
Di	20.01.2016	Funktionstests	4 h
Fr	22.01.2016	Arbeiten am Eagle 3D-Modell	5 h
Sa	23.01.2016	Dokumentieren	4 h
So	23.01.2016	Dokumentieren	3 h
So	31.01.2016	Fertigstellen der Lötstopmmaske für die Sensor-Platine sowie die Control-Platine.	4 h
Februar 2016			
Di	02.02.2016	Arbeiten an der LaTeX-Beamer Präsentation	4 h
Mi	03.02.2016	Dokumentieren, Softwareentwicklung	6 h
Fr	05.02.2016	Dokumentieren, Funktionstest	5 h
So	07.02.2016	Löten der Platinen, Funktionstest	6 h
Di	09.02.2016	Dokumentieren	5 h
Fr	12.02.2016	Versuchsaufbauten der Messschaltung aufbauen und Testprogramme schreiben	6 h
Sa	13.02.2016	Halbleiterdetektor Messaufbau testen	4 h
Fr	19.02.2016	Dokumentieren	5 h
Sa	20.02.2016	SmartVisu Änderungen vornehmen	5 h
So	21.02.2016	Dokumentieren	5 h

Tag	Datum	Beschreibung	Zeit
Do	25.02.2016	Änderungen des Programmes der KNX-Anbindung vornehmen	4 h
Fr	26.02.2016	Weitere Änderungen des Programmes der KNX-Anbindung vornehmen	3 h
Sa	27.02.2016	Dokumentieren	6 h
So	28.02.2016	Dokumentieren, Softwareentwicklung	8 h
März 2016			
Fr	04.03.2016	Ändern der internen Versorgungsspannung sowie der Taktfrequenz	2 h
Sa	05.03.2016	Dokumentieren	4 h
So	06.03.2016	Programmieren und Testen der KNX-Anbindung, Parametrieren in der ETS, Dokumentieren	5 h
Mo	07.03.2016	Programmieren und Testen der KNX-Anbindung, Parametrieren in der ETS, Dokumentieren	3 h
Di	08.03.2016	Ändern des Programms, Temperaturregelung, KNX-Kommunikation	5 h
Sa	12.03.2016	U3D-Dateien erstellen und in Dokumentation einbringen, weiteres Dokumentieren	4 h
So	13.03.2016	Programmtest, Dokumentieren	5 h
Mo	14.03.2016	Dokumentieren	6 h
Fr	18.03.2016	Dokumentieren	6 h
Sa	19.03.2016	Aufnehmen der Sprungantwort des Gehäuses, Dokumentieren	5 h
So	20.03.2016	Programmieren des digitalen Reglers, Dokumentieren	4 h
Mo	21.03.2016	Dokumentieren	6 h
Do	24.03.2016	Korrekturlesen	3 h
Fr	25.03.2016	Prowl Push-Benachrichtigung einrichten, Dokumentieren	5 h
Sa	26.03.2016	Dokumentieren	6 h
Sa	27.03.2016	Erstellung letzter nötiger Abbildungen, weiterer Funktionstest, Dokumentieren	7 h

B.1 Projektstagebücher

Tag	Datum	Beschreibung	Zeit
Mo	28.03.2016	Dokumentieren, Korrekturlesen	6 h
Summe:			407,00 h