

DIPLOMARBEIT

Gesamtprojekt

KLIMAKAMMER

**Teammitglieder:**

Simon Koller 5AHEL
Niko Leimegger 5AHEL
Joel Rupp 5AHEL

Betreuer:

Dipl.Ing. Lauritsch Franz

Rankweil, am 05.04.2024

Abgabevermerk:

DA original, am 28.05.2024

.....

Dipl.Ing. Lauritsch Franz

DA digital, am 05.04.2024

.....

AV Dipl.-Ing Moosbrugger
Leopold

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Rankweil, am 05.04.2024

.....
Simon Koller

.....
Niko Leimegger

.....
Joel Rupp

Abkürzungen & Hinweise

DA	Diplomarbeit
WPF	Windows Presentation Foundation
CAD	Computer-aided design
Regex	Regular expression
GPL	General Public License
SI	Internationales Einheitensystem
COM	Component Object Model
GUI	Graphical User Interface
UI	User Interface
MUX	Multiplexer
API	Application Programming Interface
JS	JavaScript
CSS	Cascading Style Sheet
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocoll
IC	Integrated Circuit
DB	Datenblatt
PWM	Pulsweitenmodulation
DC	Gleichstrom
ADC	Analog-Digital-Converter
DAC	Digital-Analog-Converter
GPIO	General Purpose Input/Output
MSB	Most significant bit

Inhaltsverzeichnis

Inhaltsverzeichnis	III
DIPLOMARBEIT DOKUMENTATION	7
DIPLOMA THESIS DOCUMENTATION	9
ZUSAMMENFASSUNG	12
Abstract (English)	13
1 Pflichtenheft	14
1.1 Kategorien und Einteilung der Aufgaben	14
1.2 Anforderungen	15
1.2.1 Gehäuse	15
1.2.2 Spannungsversorgung	15
1.2.3 Wassersystem	16
1.3 Meilensteine	16
2 Mechanik	18
2.1 Gehäuse	18
2.1.1 Grundgerüst	18
2.1.2 Verkleidung	20
2.1.3 Innenkammer	22
2.2 Wassersystem	25
2.2.1 Ziel	25
2.2.2 Regner-Modul	26
2.2.3 Druckaufbau und Filtrierung	29

2.2.4 Transportmedium	31
2.2.5 Steuerung	32
2.2.6 Adapter	33
2.2.7 Verkalkung	35
2.2.8 Aufbau	35
3 Hardware	37
3.1 Spannungsversorgung	37
3.1.1 Server-PSU	37
3.1.2 Platine zur Verteilung der Spannung	43
3.1.3 Befestigung im Gehäuse	47
3.2 Zentrale Steuerung	48
3.2.1 Raspberry Pi	49
3.2.2 I ² C-Extender	50
3.2.3 Platine	51
3.3 Steuerplatine für alle Module	55
3.4 Wassersteuerung	58
3.4.1 Steuerung der Pumpe	59
3.4.2 Steuerung des Ventils	60
3.4.3 Lesen des Drucksensors	62
3.4.4 Kommunikation mit der Steuerplatine	62
3.4.5 Platine	63
3.5 Sonnensystem	79
3.5.1 Theorie	79
3.5.2 Ansatz	79
3.5.3 Lampen	80

3.5.4	Steuerung	80
3.6	Kameraerfassung	84
3.7	Sensoren	85
3.7.1	BME280	86
3.7.2	LTR-390	86
3.8	Windsteuerung	87
4	Software	89
4.1	Website	89
4.1.1	Anforderungen	89
4.1.2	Framework und Libraries	90
4.1.3	Features	93
4.1.4	Funktionsweise	101
4.1.5	Hinzufügen eines neuen Features	104
4.2	API	110
4.2.1	Sprachsteuerung	119
5	Firmware	128
5.1	Vorlage	129
5.1.1	Verwendung von zwei I2C-Bussen	132
5.2	Wassersystem	132
5.2.1	Setup der Pins	132
5.2.2	Steuerung der Module	134
5.3	Sensoren	137
5.3.1	BME280	138
5.3.2	LTR-390	139

5.4	Sonnensteuerung	140
5.5	Firmware für die Lüftersteuerung	141
6	Projektmanagement	143
6.1	Verwendete Software	143
6.2	Tracking der Stunden	144
6.3	Dokumentation der Zwischenergebnisse	145
7	Ausblick	147
8	Abbildungsverzeichnis	148
9	Tabellenverzeichnis	153
10	Codeverzeichnis	154
11	Quellenverzeichnis	155

DIPLOMARBEIT DOKUMENTATION

Name der Verfasser innen	Simon Koller, Niko Leimegger, Joel Rupp
Jahrgang Schuljahr	5AHEL 23SJ 24SJ
THEMA der Diplomarbeit	Klimakammer
Kooperationspartner	WolfVision GmbH
Aufgabenstellung	Ziel ist es, eine Klimakammer zu entwickeln, welche Parameter wie Sonne, Niederschlag und Wind berücksichtigt. Diese soll über eine Website gesteuert werden können, und über eine Wetter-API verschiedene Orte der Welt abbilden können.

Individuelle Themenstellung im Rahmen des Gesamtprojektes

Simon Koller	Entwicklung eines Wassersystems, das in der Lage ist Nieselregen bis Starkregen innerhalb der Kammer zu realisieren und die Luftfeuchtigkeit innen, im Zusammenspiel mit Wind, verändern kann. Programmierung eines Algorithmus, welcher aus Sprachbefehlen, die von einer Künstlichen Intelligenz erkannt werden, die nötigen Parameter filtert, um den gegebenen Befehl auszuführen.
Niko Leimegger	Entwicklung einer Website zur Steuerung der Klimakammer, die alle wichtigen Messparameter visualisiert und es ermöglicht, den Kamerastream aus der Kammer abzurufen. Einbindung einer Sprachsteuerung auf der Website sowie Aufbau und Konfiguration eines AWS-Servers zur Verarbeitung der Sprachbefehle.
Joel Rupp	Projektplanung. Entwickeln des Gehäuses. Bereitstellen der Spannungsversorgung. Entwickeln einer API für Aktuatoren. Entwickeln einer Steuerplattform für die Aktuatoren. Steuerung des Windes. Auslesen der Sensoren
Realisierung	Eigene Klimakammer zur Realisierung gefertigt und mit unseren Aktoren und Sensoren ausgestattet.
Ergebnisse	In der Klimakammer können Wind, Sonne und Niederschlag eingestellt werden. Außerdem kann ein Ablauf erstellt werden, oder dieser automatisch mit einer Wetter-API erstellt werden.
Einsichtnahmen**)	Archiv der HTL Rankweil, www.diplomarbeiten.berufsbildendeschulen.at

DIPLOMA THESIS DOCUMENTATION

Author(s)	Simon Koller, Niko Leimegger, Joel Rupp
Form Academic year	5AHEL 23SJ 24SJ
Diploma Thesis Topic	Klimakammer
Cooperation Partners	WolfVision GmbH
Assignment of Tasks	The objective is to develop a climate chamber that accounts for parameters such as sunlight, precipitation, and wind. It should be controllable via a website and capable of replicating various locations around the world through a weather API.

Individual Tasks within the overall project

Simon Koller	Development of a water system, which is able to perform drizzle up to heavy rain inside the chamber and changing the humidity inside in cooperation with wind. Programming an algorithm, which filters the important parameters from voice commands, interpreted by an artificial intelligence, to perform said commands.
Niko Leimegger	Development of a website for controlling the climate chamber, which visualizes all important measurement parameters and allows accessing the camera stream from the chamber. Integration of voice control on the website as well as setup and configuration of an AWS server for processing voice commands.
Joel Rupp	Project planning. Housing development. Provision of power supply. API development for actuators. Development of a control platform for actuators. Wind control. Sensor data retrieval.
Realisation	Custom-built climate chamber realized and equipped with our actuators and sensors.
Result	In the climate chamber, wind, sunlight, and precipitation can be adjusted. Additionally, a sequence can be created manually, or it can be generated automatically using a weather API.
Publication**)	Archive of HTL Rankweil, www.diplomarbeiten.berufsbildendeschulen.at

Vorwort und Danksagung

Das Team der Diplomarbeit will sich aufrichtig bei allen Unterstützern bedanken. Voran bei unserem Betreuungslehrer Franz Lauritsch der uns bei jeglichen Theorie und Praxisfragen zur Seite stand.

Wir bedanken uns bei unseren Familien, welche in dieser teils stressigen Zeit immer ein offenes Ohr hatten, und uns mit Rat und Tat zur Seite standen.

Einen besonderen Dank möchten wir an unsere Partnerfirma und unseren Ansprechpartner Dieter Mathis richten. Ohne die zur Verfügung gestellten Ressourcen wäre diese Diplomarbeit nie im Ansatz möglich gewesen, und das vermittelte Wissen und Know-How hat uns in der Realisierung weitgehend unterstützt.

Pascal Berchthold von der Firma Sanlight hat uns wichtige Theorie zur Beleuchtung der Kammer vermittelt, auf die wir nie selbst gekommen wären. Dafür, und für die großzügige Unterstützung in der Beschaffung der Leuchten, wollen wir uns bedanken.

Bei der Ing. R. Longhi GmbH & Co KG bedanken wir uns für die Fertigung des Grundgerüsts unserer Klimakammer und die unkomplizierte Bereitstellung weiterer benötigter Komponenten.

Nicht zuletzt wollen wir uns bei unseren Mitschülern und allen Lehrern bedanken, die uns in kleinster Sache unterstützt und mit uns mitgegrübelt haben. Nochmals herzlichen Dank an alle, die uns in diesem Projekt begleitet und unterstützt haben.

Zusammenfassung

Die Motivation und Ausgangslage für die Entwicklung einer Klimakammer basiert auf dem Bedarf, klimatische Umweltsimulationen in Laboren weltweit durchführen zu können. Diese Klimakammer, ausgestattet mit Modulen für Sonnenlicht, Regen und Wind, soll durch einen Raspberry Pi über eine Website gesteuert werden. Sie ermöglicht es, dynamische klimatische Bedingungen manuell oder automatisch mittels einer Live Wetter API einzustellen.

Der Hauptzweck dieser Klimakammer liegt in der Produktverifizierung. Zum Beispiel sind Klimakammern essentiell für statische Klimaprüfungen, Klimawechseltests und insbesondere für den Nachweis der Funktionsfähigkeit von Produkten unter spezifischen klimatischen Bedingungen. Solche Tests sind üblicherweise nicht Teil der Standardlaborprüfungen, sondern erfolgen am Ende der Produktentwicklung unter realen Bedingungen. Nicht bestandene Verifikationstests können zu einem Neustart der Produktentwicklung führen. Daher soll die Entwicklung dieser Klimakammer dazu beitragen, solche Risiken zu minimieren, indem praxisnahe Umweltprüfungen frühzeitig im Labor durchgeführt werden können. Solche Umweltprüfungen bzw. Standards wären z.B. IP¹ oder MIL-STD-810², bei denen z.B. auch die Widerstandsfähigkeit gegen Regen und Sonnenlicht getestet wird. Auch die Lagerung und der Transport der Produkte setzt sie Umwelteinflüssen aus. Diese Parameter können in handelsüblichen Klimakammern nicht simuliert werden.

Zur Erleichterung der Wartung und Erweiterung der Klimakammer soll diese weitgehend modular aufgebaut werden. Das heißt, dass die einzelnen Kernthemen (Sonne, Regen, usw.) eigenständige Module bilden, die über einen Bus mit einer zentralen Steuerung verbunden werden können.

¹ ISO 20653 - 2023-08 - Beuth.de. URL: <https://www.beuth.de/de/de/norm/iso-20653/372581287> (besucht am 08.03.2024).

² MIL-STD-810H. URL: <https://www.iest.org/Standards-RPs/MIL-STD-810H> (besucht am 28.01.2024).

Abstract (English)

The motivation and initial situation for the development of a climate chamber are based on the need to perform climatic environmental simulations in laboratories worldwide. This climate chamber, equipped with modules for sunlight, rain, and wind, is intended to be controlled via a website using a Raspberry Pi. It allows for the manual or automatic adjustment of dynamic climatic conditions through a live weather API.

The primary purpose of this climate chamber is for product verification. For example, climate chambers are essential for static climate testing, climate change tests, and especially for demonstrating the functionality of products under specific climatic conditions. Such tests are typically not part of standard laboratory testing but are conducted at the end of product development under real conditions. Failing verification tests can lead to a restart of product development. Therefore, the development of this climate chamber aims to minimize such risks by enabling practical environmental testing early in the laboratory. Examples of such environmental tests or standards include IP (Ingress Protection)³ or MIL-STD-810⁴ (Military Standard), which, among other things, also test resistance to rain and sunlight. Additionally, the storage and transport of products expose them to environmental influences. These parameters cannot be simulated in commercial climate chambers.

To facilitate maintenance and expansion, the climate chamber is intended to be largely modular. This means that the individual core topics (sun, rain, etc.) form independent modules that can be connected to a central control unit via a bus system.

³ ISO 20653 - 2023-08 - Beuth.de.

⁴ MIL-STD-810H.

1 Pflichtenheft

1.1 Kategorien und Einteilung der Aufgaben

Am Anfang der Projektplanung wurde ein Pflichtenheft und eine Grobplanung der Arbeitspakete erstellt. Diese sind:

- Core-Features

Dies sind Features, die vorhanden sein müssen, um die anderen, untergeordneten Features überhaupt zu ermöglichen, haben also allerhöchste Priorität.

Dazu zählen:

- Gehäuse, S. 18 ff.
- Spannungsversorgung, S. 37 ff.
- Zentrale Steuerung, S. 48 ff.

- Key-Features

Dies sind Features, welche der Klimakammer Funktion geben, und unbedingt erfüllt werden müssen. Dazu zählen:

- Regen, S. 25 ff. und S. 58
- Sonne, S. 79 ff.
- Kameraerfassung, S. 84 ff.

- Optionale Features

Features welche der Kammer Funktionen geben, aber nicht zwingend notwendig sind. Sie haben die niedrigste Priorität, werden also erst bearbeitet wenn alle anderen Features fertig sind

In Abbildung 1.1.1 ist ein Blockschaltbild gegeben, welches die Funktionsweise der Klimakammer visualisiert. Dabei ist die zentrale Steuerung das „Gateway“ zwischen Website und den einzelnen Modulen. Jedes Modul ist mit einem Raspberry Pico ausgestattet, und wird von der zentralen Steuerung über I2C angesprochen. Falls die zentrale Steuerung Daten vom Modul anfordert, werden diese ebenfalls über I2C zurückgesendet.

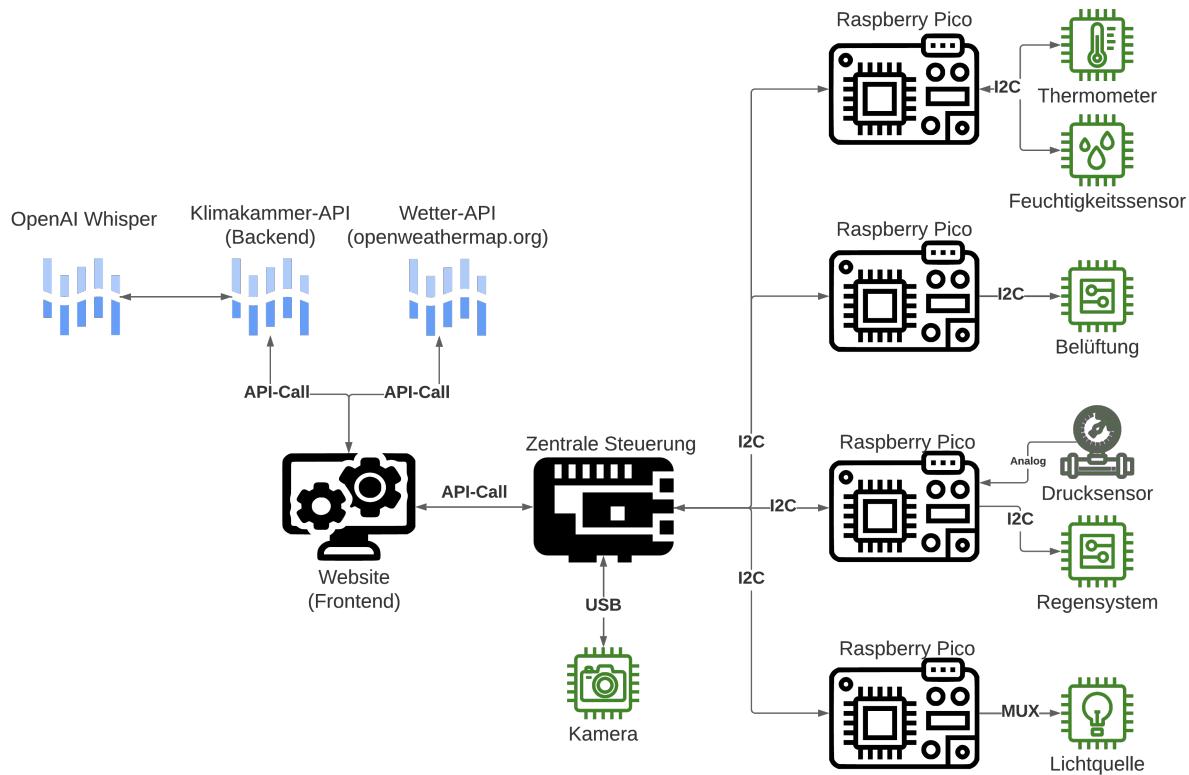


Abbildung 1.1.1: Blockschaltbild des Gesamtsystems

1.2 Anforderungen

1.2.1 Gehäuse

- **Robustheit:** Das Grundgerüst muss robust sein, und neben seinem eigenen Gewicht, auch das Gewicht der Verkleidung, der Innenkammer, unserer Hardware und des Prüflings tragen können.
- **Machbarkeit:** Es soll leicht machbar sein, also am Besten aus am Markt leicht verfügbaren Teilen bestehen.
- **Modifizierbarkeit:** Da das gesamte Projekt darauf ausgelegt ist, leicht erweitert werden zu können, soll auch das Grundgerüst so entworfen werden, dass es leicht angepasst werden kann, und z.B. Bohrlöcher für Kabel überall platziert werden können, ohne die strukturelle Integrität zu gefährden.

1.2.2 Spannungsversorgung

- **Skalierbarkeit**
Die Leistung der Spannungsversorgung soll gut skaliert werden können
- **Plug-and-Play**

Da die Klimakammer ein eigenständiges Produkt bilden soll, soll die Anbindung an das Stromnetz am besten direkt erfolgen, sie soll einfach an der Steckdose eingesteckt werden können

- **Sicherheit**

Durch die direkte Anbindung an das Stromnetz muss eine gewisse Sicherheit vor der Netzspannung gegeben werden, wie sie z.B. durch CE-Kennzeichnung gewährleistet wird. Auch wird Wert auf die mechanische Sicherheit gelegt, das Powersupply soll also robust sein, und ein Berühren der Netzspannung verhindern.

- **Spannung**

Die Ausgangsspannung sollte sowohl hoch genug sein, um einiges an Leistung ohne riesige Ströme zu ermöglichen, aber gleichzeitig auch niedrig genug um von der Elektronik relativ leicht verwendet werden zu können

1.2.3 Wassersystem

- **Niederschlag:** Wir wollen jedes mögliche Klima der Welt nachbilden, deswegen muss das Modul im Stande sein alles zwischen Niesel- und Starkregen nachzubilden.
- **Platzbedarf:** Da die Maße der Innenkammer sehr klein spezifiziert worden sind, soll das Modul möglichst keinen Platz im Inneren der Kammer beanspruchen.
- **Automatisierung:** Wie jedes Modul, muss auch das Wassersystem streng automatisiert gesteuert werden können.
- **Isolation:** Um jegliche Elektronik oder andere wichtige Bestandteile des Projektes zu schützen muss das komplette Wassersystem nach außen isoliert sein.

1.3 Meilensteine

Alle Meilensteine mit geplanten Enddatum und tatsächlichem Enddatum werden in der Tabelle 1.3.1 dargestellt. Die benötigte Funktion für die Komplettierung des Meilensteins kann im jeweiligen Kapitel nachgelesen werden

Meilenstein	Zuständig	Geplantes Enddatum	Tats. Enddatum
Zentrale Steuerung Hardware	Joel Rupp	29.10.2023	11.12.2023
Gehäuse Grundgerüst	Joel Rupp	26.11.2023	09.11.2023
Spannungsversorgung Hardware	Joel Rupp	26.11.2023	11.12.2023
Gehäuse Verkleidung	Joel Rupp	10.12.2023	04.12.2023
Niederschlag Modul-Planung	Simon Koller	11.12.2023	18.12.2024
Grundgerüst Website	Niko Leimegger	10.12.2023	13.11.2023
Auslesen und Darstellen von Daten	Niko Leimegger	31.12.2023	20.11.2023
Wetter API einbauen	Niko Leimegger	31.12.2023	20.12.2023
Zentrale Steuerung Software	Joel Rupp	31.12.2023	07.02.2024
Wassersystem Hardware	Simon Koller	15.01.2024	20.01.2024
Gehäuse Innenkammer	Joel Rupp	28.01.2024	26.02.2024
Spannungsversorgung Firmware	Joel Rupp	28.01.2024	02.01.2024
Kameraerfassung	Joel Rupp	28.01.2024	04.02.2024
Ablauf an Raspberry Pi senden	Niko Leimegger	28.1.2024	1.1.2024
Sonnenmodul Steuerplatine	Niko Leimegger	30.1.2024	22.1.2024
Wassersystem Software	Simon Koller	01.02.2024	26.02.2024
Steuerplatine testen	Niko Leimegger	8.2.2024	10.03.2024
Sonne in Gehäuse einbauen	Niko Leimegger	29.2.2024	11.03.2024
Wassersystem Fertigstellung	Simon Koller	29.2.2024	11.03.2024
Sprachsteuerung	Simon Koller	20.03.2024	18.03.2024

Tabelle 1.3.1: Meilensteine

2 Mechanik

2.1 Gehäuse

2.1.1 Grundgerüst

An das Grundgerüst der Klimakammer wurden folgende Anforderungen gestellt:

- **Robustheit:** Das Grundgerüst muss robust sein, und neben seinem eigenen Gewicht, auch das Gewicht der Verkleidung, der Innenkammer, unserer Hardware und des Prüflings tragen können.
- **Machbarkeit:** Es soll leicht machbar sein, also am Besten aus am Markt leicht verfügbaren Teilen bestehen.
- **Modifizierbarkeit:** Da das gesamte Projekt darauf ausgelegt ist, leicht erweitert werden zu können, soll auch das Grundgerüst so entworfen werden, dass es leicht angepasst werden kann, und z.B. Bohrlöcher für Kabel überall platziert werden können, ohne die strukturelle Integrität zu gefährden.

Im Anbetracht der oberen Punkte wurde das Grundgerüst aus Aluprofilen designed. Da das Grundgerüst von der „Ing. R. Longhi GmbH & Co KG“ gesponsert wurde, wurde es in MTpro64⁵ entworfen. MTPro64 ist eine CAD-Software, welche von Bosch Rexroth primär für den Entwurf von Montagestraßen entwickelt wurde. Daher sind im Program bereits alle von Bosch Rexroth verfügbaren Teile vorhanden. Das fertige Grundgerüst ist in Abbildung 2.1.1 zu sehen. Ein Vorteil von MTpro64 ist, dass es vorgefertigte Modelle für Menschen gibt. Damit war es leichter, die Kammer auf die richtige Größe zu designen. In Abbildung 2.1.2 sieht man einen Mann, mit der mittleren Größe eines europäischen Mannes, der neben dem Grundgerüst steht. Dabei sieht man, dass die Öffnung der Kammer, auf der richtigen Höhe liegt. Das gesamte Projekt wurde dann an „Ing R. Longhi GmbH & Co KG“ gesendet, und könnte bereits kurz später fertig abgeholt werden. Dabei bekamen wir noch einige Schrauben und Nutensteinen, und z.B. die Verkleidung befestigen zu können.

⁵ MTpro Planungssoftware. Bosch Rexroth Deutschland. URL: <https://www.boschrexroth.com/de/de/produkte/produktgruppen/montagetechnik/themen/mtpro-planungssoftware/> (besucht am 09.03.2024).

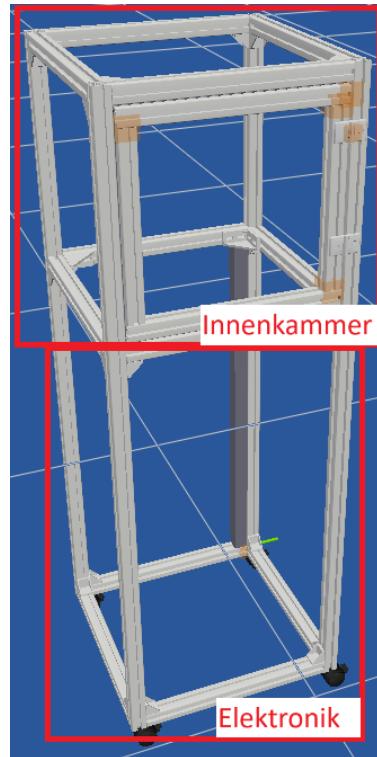


Abbildung 2.1.1: Grundgerüst in MTpro64

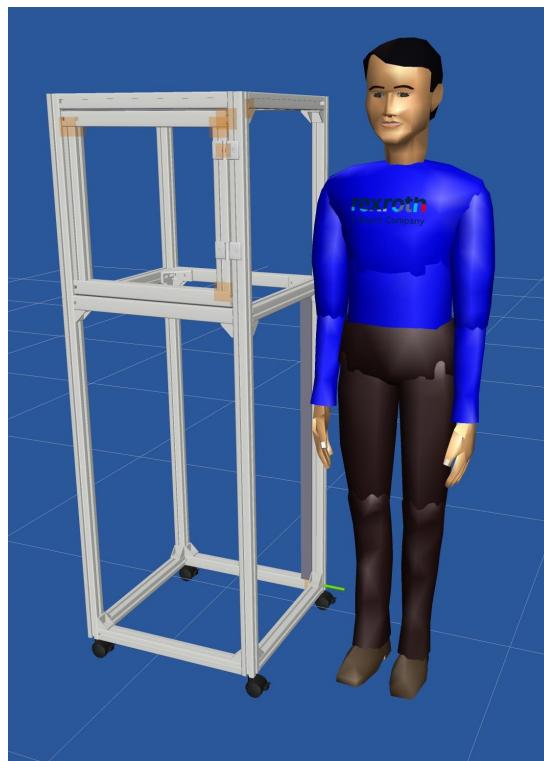


Abbildung 2.1.2: Grundgerüst mit Mann mittlerer Größe

2.1.2 Verkleidung

Material Zur Verkleidung der Kammer wurde zuerst in der Schule nachgefragt, welche Materialen in dieser Größe verfügbar sind, da das Grundgerüst eine Höhe von ca. 170cm hat, und wir jede Seite der Verkleidung aus einem Stück haben wollten. Dabei wurde uns empfohlen, 6mm dicke Acrylglasplatten zu verwenden, da diese zur Genüge vorhanden waren.

Planung Für die Planung der Verkleidung wurde das Grundgerüst aus MTpro64 als STL exportiert, und in Solidworks importiert. Da dabei aber Probleme mit der Performance von Solidworks auftraten, wurde es in Solidworks nachgebaut. Damit konnten die Maße direkt am Modell genommen werden, welches keinerlei Abweichungen zum fertigen Grundgerüst hatte. Das fertige Model ist in Abbildung 2.1.3 zu sehen. Dabei fehlen noch die Löcher für die Lüfter und die Lichter, da diese während des Fertigungsprozesses abgezeichnet wurden.

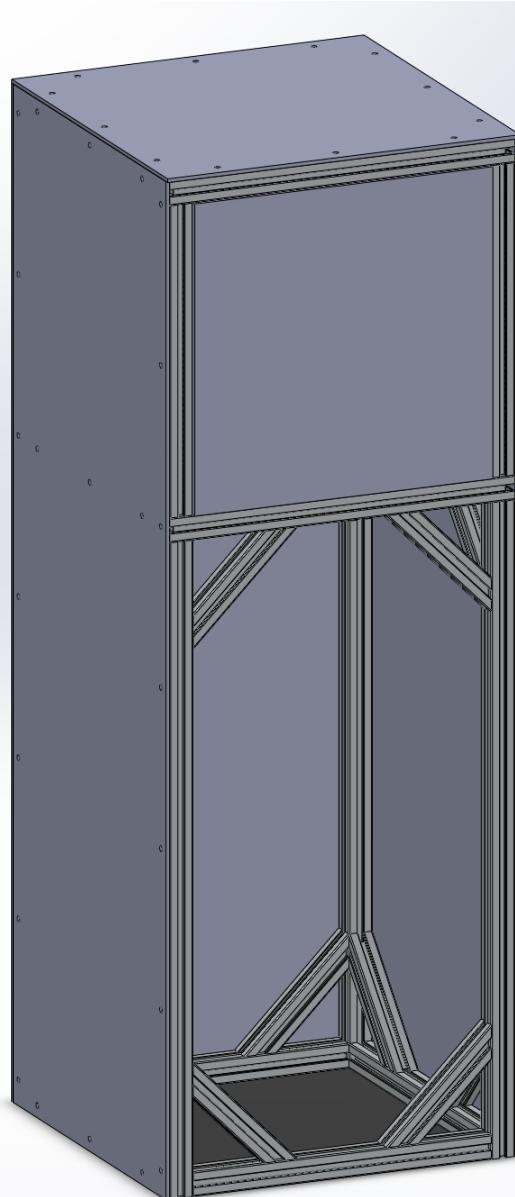


Abbildung 2.1.3: Fertiges Gehäuse in Solidworks

Herstellung Bei der Herstellung traten einige Probleme auf. So gestaltete sich bereits das Zusägen der Platten schwierig, da diese nicht ganz auf die Tischkreissäge passten, und somit selber nachgeschoben werden mussten. Sie wurden aber trotzdem mit der Tischkreissäge zugesägt, da damit die passenden Maße garantiert werden konnten.

Beim Bohren der Löcher taten sich dann aber größere Schwierigkeiten auf, da mit den in der Schule verfügbaren HSS-Bohrern das Acrylglass sehr stark aufriss. Daher wurden ein Holzbohrer verwendet, welche im Internet dafür empfohlen wurden⁶, weil Acrylglass ähnliche Eigenschaften wie Holz habe. Mit diesem Bohrer in Kombination mit extrem niedriger Drehzahl und viel Kühlmittel gelangen Löcher ohne Ausreißen

Beim Aussägen der Aussparung für die Tür traten wieder einige bekannte Probleme auf. Dabei riss die Stichsäge das Acrylglass zuerst stark auf, was aber durch eine niedrigere Drehzahl und mehr Kühlung gelöst werden konnte. Am Ende entstand durch die Vibration der Säge ein Riss durch die gesamte Oberseite. Da die Verkleidung nur kosmetisch ist, und der Riss nur beim genauen Hinsehen zu erkennen ist, wurde das Teil trotzdem verbaut.

2.1.3 Innenkammer

An die Innenkammer wurden einige Anforderungen gestellt:

- **Größe:** Die Innenkammer muss in die Aussparung im Grundgerüst passen, also ca. 50cm lang, breit und hoch sein
- **Dichtheit:** Die Innenkammer muss wasserdicht sein, so dass kein Wasser auf die Elektronik tropft
- **Abfluss des Wassers:** Da es im Inneren der Kammer regnen soll, muss dieses Wasser auch wieder abtransportiert werden. Dafür wurde der Boden der Innenkammer mit einer Schräge von ca. 2% geplant, der gleiche Wert wie in einer offenen Dusche. Zum Abfluss wurde eine Gardena Schlauchkupplung an den tiefsten Punkt der Kammer montiert.
- **Locher zur Lüftermontage:** Um Luftzirkulation innerhalb der Kammer zu gewährleisten, müssen Löcher eingeplant werden, an welche Lüfter montiert werden können.
- **Integrierung der Beleuchtung:** Für die Beleuchtung müssen an der Decke

⁶ noauthor_toon.

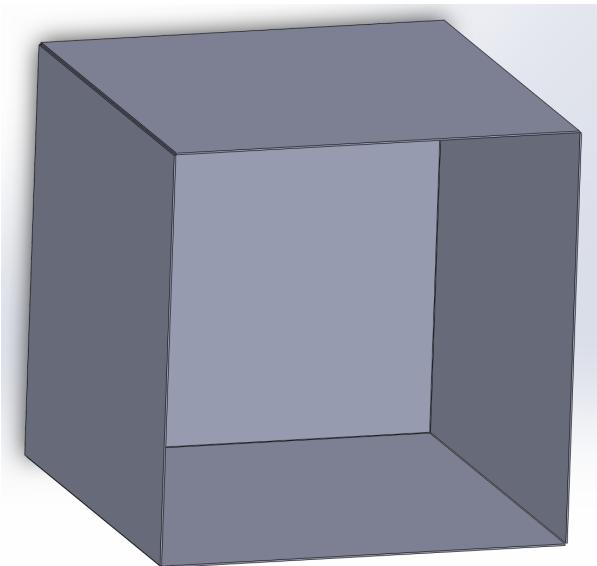


Abbildung 2.1.4: Innenkammer in Solidworks

der Kammer ebenfalls rechteckige Löcher ausgesägt werden, in welche die Lichter eingelassen werden können.

Material Als Material wurde von uns Alublech gewählt, da dieses gegen Wind und Wetter, wie es eben simuliert werden soll, widerstandsfähig ist. Außerdem kann es gebogen werden, und somit aus einem Teil hergestellt werden und es muss nicht jede Kante abgedichtet werden.

Planung Zur Planung wurde Solidworks⁷ verwendet, da es in Solidworks bereits Funktionen zur Erstellung von Biegeteilen gibt. Solidworks ist ein CAD-Programm, in welchem 3D-Konstruktionen erstellt werden können. Dabei werden verschiedene Faktoren wie z.B. der Biegeradius automatisch mitberechnet, und es ist sicher gestellt dass die gebogenen Teile später auch passen. Dabei kann am Anfang eine Basisfläche erstellt werden, an welche dann immer wieder weitere Flächen hinzugefügt werden. In Abbildung 2.1.4 ist die Innenkammer in Solidworks zu sehen, jedoch fehlen hier einige Löcher, da diese während der Herstellung abgezeichnet wurden

⁷ 3D-CAD-Konstruktionssoftware | SOLIDWORKS. URL: <https://www.solidworks.com/de/home-page-2021> (besucht am 17.03.2024).

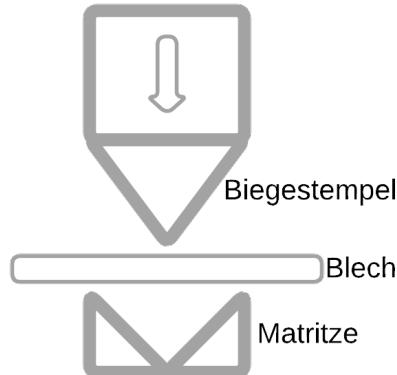


Abbildung 2.1.5: Funktionsweise Gesenkbiegen

Herstellung Beim Biegen traten einige Probleme auf, weil die Innenkammer etwas zu groß für die Gesenkbiegepresse⁸ der Schule war. Wie in Abbildung 2.1.5 zu sehen, wird beim Blech bearbeiten mit einer Gesenkbiegepresse ein Biegestempel hinunter bewegt, welcher das Blech in die Matritze drückt. Das Problem dabei ist, dass das Blech nach der zweiten Biegung quasi herausgedreht werden muss. Dies war aber nicht möglich, da die Presse nach oben mehr Platz brauchte, als die Kammer hoch war, wie in Abbildung 2.1.6 zu sehen. Als Lösung wurden dann einige Biegestempel ausgebaut, um die Kammer nach dem Biegen zur Seite schieben, und der Gesenkbiegemaschine entnehmen zu können.

Bei der letzten Kante der Innenkammer konnte die Gesenkbiegemaschine nicht mehr verwendet werden, da die beiden seitlichen Kanten der Kammer nun bereits beim Einlegen in die Maschine mit ihr kollidierten. Nun wurde eine andere Methode zum Blechbiegen verwendet. Anstatt das Material mit einem Stempel von oben in eine Matritze zu pressen, wurde wie beim Schwenkbiegen⁹ das Material festgehalten, und bis auf den gewünschten Winkel gebogen. Da die Kammer für die Schwenkbiegemaschine in der Schule zu groß ist, wurde die Kammer mit Muskelkraft gebogen. Die führte zu einer großen Ungenauigkeit. Dies Ungenauigkeit wurde später mit dem Hammer ausgeglichen.

⁸ *Gesenkbiegepresse*. In: Wikipedia. Page Version ID: 228053128. 17. Nov. 2022. URL: <https://de.wikipedia.org/w/index.php?title=Gesenkbiegepresse&oldid=228053128> (besucht am 09.03.2024).

⁹ *Schwenkbiegen*. In: Wikipedia. Page Version ID: 238327141. 20. Okt. 2023. URL: <https://de.wikipedia.org/w/index.php?title=Schwenkbiegen&oldid=238327141> (besucht am 09.03.2024).

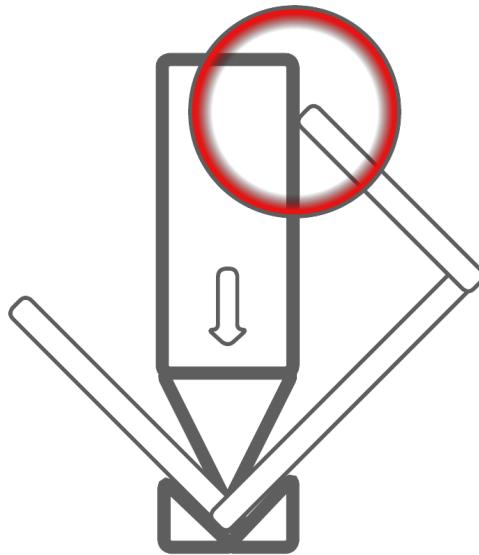


Abbildung 2.1.6: Kollision des Werkstückes mit der Gesenkbiegenmaschine

Montieren im Grundgerüst Die Kammer wurde mit dem Grundgerüst mit Silikon verbunden. Dies wurde Schraubverbindungen vorgezogen, da das Silikon als Schwingungsdämpfer arbeiten können, und somit der Transport der Kammer beinahe geräuschlos wurde.

Abdichten mit Silikon Um die Dichtheit der Kammer zu gewährleisten, wurden alle offenen Kanten mit Silikon abgedichtet.

2.2 Wassersystem

2.2.1 Ziel

Die Zielsetzung des Wassersystems, oder Regen-und-Feuchte-Moduls, ist dahingehend formuliert, dass es möglich sein soll in der Kammer zu regnen und das möglichst naturgetreu. Das hebt diese Kammer von allen anderen auf dem Markt erhältlichen Klimaschränken ab, da diese nur Feuchte erzeugen können und keinen messbaren Regen. Dieses Modul soll gegen Ende ein USP darstellen. Das bedeutet, dieses Modul soll für die Verwendung der Klimakammer sprechen und im Gegensatz zu Kontrahenten auf dem Markt einzigartig sein.

Um dieses Vorhaben zu realisieren muss ein Objekt vorliegen, welches in der Lage ist aus Wasser Sprühregen oder Tröpfchen zu erzeugen. Des Weiteren muss das Wasser zu diesem Objekt transportiert werden. Damit sich nicht das ganze nieder-

geschlagene Wasser am Boden der Kammer sammelt, sollte es auch wieder abtransportiert werden. Wie das Ganze erreicht wird und welche weiteren wichtigen Bestandteile unverzichtbar sind, wird in den jeweiligen Sub-Themen beschrieben.

2.2.2 Regner-Modul

Das Objekt, welches letztendlich in der Kammer den Regen erzeugen soll, hat viele Vorgaben zu erfüllen:

- **Naturnähe**

Wie bereits erwähnt soll der Niederschlag so nah wie möglich an der Natur sein. In den Punkten Dauer, Intensität und impulsartigen Änderungen.

- **Platzbedarf**

Da das Objekt einen Platz an der Decke der Kammer finden muss, ist der Platzbedarf von großer Wichtigkeit, da von diesem, bei einer 50cm mal 50cm großen Fläche, nur wenig zu Verfügung steht. Ebenfalls muss sich die Decke mit der Simulation der Sonne geteilt werden. Aus diesen Gründen wäre ein Objekt mit geringem Platzbedarf zu bevorzugen.

- **Automatisierbarkeit**

Um verschiedene Niederschläge und Änderungen zu realisieren, muss das Objekt automatisiert über einen Parameter z.B. durch die Wassermenge oder den Wasserdruck steuerbar sein.

- **Skalierbarkeit**

Damit die im Punkt Naturnähe genannten Änderungen erzielt werden können, muss das Objekt eine gewisse Skalierbarkeit aufweisen. Es soll sich also der Niederschlag bei einer Änderung des Eingangsparameters verändern.

Letztendlich fiel die Entscheidung den Schlauch-Regner von Gardena (Artikel-Nr. 995-20)¹⁰, als jenes Objekt zu verwenden. Diese Entscheidung wurde maßgeblich durch eine Expertenmeinung beeinflusst. Diese wurde am 02.11.2023 im Dehner Garten-Center in Hohenems eingeholt. Laut jener Einschätzung wird durch die Verwendung des Schlauch-Regners das Ergebnis mit der höchsten Naturnähe erzielt. Zusätzlich handelt es sich dabei um die Lösung mit dem geringsten Platzbedarf.

Aus Abbildung 2.2.1¹¹ lässt sich herauslesen, dass der Schlauch-Regner je nach Platzierung das Wasser sowohl nach oben als auch nach unten abstrahlen kann. Für

¹⁰ Schlauch-Regner | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/regner/schlauch-regner/900883801.html> (besucht am 04.04.2024).

¹¹ Schlauch-Regner | GARDENA.

die Realisierung eines Niederschlags ist allerdings nur letzteres von Bedeutung, da er an der Decke platziert wird. In Abbildung 2.2.2¹² ist ersichtlich, dass auch Niederschläge mit hoher Intensität/Kraft realisierbar sind.

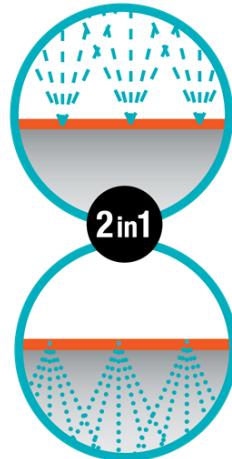


Abbildung 2.2.1: Anwendungen des Schlauch-Regners



Abbildung 2.2.2: Schlauch-Regner in Aktion

Der Schlauch-Regner ist mit feinen Löchern versehen durch welche das zugeführte Wasser in Form von Sprühnebel und/oder Tröpfchen nach außen gelangt.

¹² Schlauch-Regner | GARDENA.

Laut Hersteller sorgen die feinen Löcher in dem aus PVC bestehenden Schlauch-Regner, welche im Abstand von jeweils 16cm angeordnet sind, für eine maximale Sprengbreite von 3 Metern. Mehr als ausreichend also für die geplante Maße mit einer 0,25m² großen Grundfläche. Allerdings darf ein Maximaldruck von 6 bar nicht überschritten werden.

Weitere Ansätze waren etwa ein Perlschlauch wie man es aus Gärten kennen könnte. Im Grunde handelt es sich dabei um dasselbe Prinzip wie bei dem präferierten Schlauch-Regner. Der einzige signifikante Unterschied ist dabei, dass Perlschlüche wie klassische Schlüche rund sind, nicht flach wie der Schlauch-Regner, und somit das Wasser zu allen Seiten ausgestoßen wird. In Abbildung 2.2.3 ist ein klassischer Perlschlauch¹³ abgebildet.



Abbildung 2.2.3: Perlschlauch

Diese Idee wurde verworfen, da wie erwähnt, laut Expertenmeinung, der Schlauch-Regner das Ergebnis, welches der Natur ähnlicher ist erzielt.

Für kurze Zeit wurde erwägt einen eigenen Duschkopf zu bauen. Um sich das besser vorzustellen: Die Idee war zwei Aluminium-Platten übereinander zu legen. Wo bei die obere wie eine Art Schale geformt sein soll damit sich das Wasser dort ansammeln kann und auf ihrer unteren Seite mit Löchern versehen, damit das Wasser niederschlägt. Während die untere Platte die Intensität steuern soll. Sie wird mit denselben Löchern ausgestattet wie die obere, so dass wenn die beiden Platten exakt übereinander liegen, kein Unterschied bemerkbar ist. Nun soll die untere Platte

¹³ Perl-Regner | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/regner/perl-regner/900969401.html> (besucht am 04.04.2024).

aber mit einem Motor verschoben werden, die Löcher liegen dann nicht mehr exakt übereinander und es kann weniger Wasser nach unten regnen. Diese Idee wurde verworfen, weil als Team inklusive Dieter Mathis und Franz Lauritsch entschieden wurde, dass das Bauen eines solchen Duschkopfes viel zu zeitintensiv ist und der Platzbedarf im Verhältnis zu den anderen Ideen immens größer ist, die „Sonne“ würde keinen Platz mehr haben.

2.2.3 Druckaufbau und Filtrierung

Nun muss das Wasser zum Schlauch-Regner transportiert werden. Das gelingt mit einer Wasserpumpe. Diese Pumpe muss genügend Druck erzeugen, damit verschiedene Intensitäten des Niederschlags eingestellt werden können. Außerdem muss sie in der Lage sein das Wasser weit genug (auch nach oben) zu transportieren und einen ausreichenden Wasserfluss zu erzeugen.

Eine billige Hochdruckpumpe von Amazon¹⁴ ist für diese Anwendung zureichend. Die Pumpe, für die wir uns entschieden haben ist in Abbildung 2.2.4 dargestellt.

¹⁴ Wasserpumpe Wohnmobil 12V, Druckwasserpumpe Membranpumpe Selbstansaugende mit Druckschalter Automatik Hochdruckreinigerpumpen, für Camping Boot Autoreinigung : Amazon.de: Auto & Motorrad. URL: https://www.amazon.de/dp/B09KZFVMPN?ref_=cm_sw_r_cp_ud_dp_9MYF29QS3533C3Z05ZR8 (besucht am 04.04.2024).



Abbildung 2.2.4: Hochdruck-Wasserpumpe der Marke Ejoyous

Diese DC-Pumpe hat folgende Maximal-Werte:

• **Wasserdruck**

Die Pumpe erreicht einen maximalen Druck von 0,8MPa also 116psi oder 8bar, welcher allerdings aufgrund der Beschränkung durch den Schlauch-Regner nicht zur Gänze genutzt werden kann.

• **Reichweite**

Die maximal erreichbare Reichweite des Wassers liegt bei 7m, sollte also mehr als genug sein aufgrund der kleinen Dimensionen der Klimakammer.

• **Wasserfluss**

Die Pumpe erreicht einen maximalen Wasserfluss von 5l/min. Bei der kleinen Fläche der Klimakammer ist das mehr als ausreichend. Bei der Fläche von 50cm mal 50cm ergeben 5l/min einen Niederschlag von 300l/h auf 0,25m². Zum Vergleich: Bei einem Niederschlag ab 40l/m² in einer Stunde spricht man von einem extremen Unwetter(quelle). Wobei aber zu beachten ist, dass durch die feinen Löcher im Schlauch-Regner nie die kompletten 5l/min ausgeschöpft

werden können.

Der Nachteil dieser Pumpe liegt im hohen Strombedarf von bis zu 5A. Das bedeutet, dass ein Treiber benötigt wird, der einem solchen Strom standhalten kann.

Ebenfalls wird ein Filter eingebaut, welcher dafür sorgt, dass es innerhalb des Wasserkreises zu keinen Verschmutzungen oder gar Verstopfungen kommt. In Abbildung 2.2.5 ist ein Filter mit Doppelschicht zu sehen, welcher für Schläuche mit bis zu 1/2“ oder 13mm Durchmesser geeignet ist.¹⁵



Abbildung 2.2.5: Filter der Marke LATRAT

Dieser Filter kann leicht mit Schläuchen verbunden werden. Einzig eine Schlauchschelle wäre noch von Vorteil um die Verbindung zu verstärken.

2.2.4 Transportmedium

Natürlich kann das Wasser nicht über die Luft von der Pumpe zum Schlauch-Regner transportiert werden. Deswegen wird ein Transportmedium benötigt, welches dem Maximaldruck der Pumpe von 8 bar standhält. Denn obwohl nur 6 bar für Anwendungen mit dem Schlauch-Regner zulässig sind, wären die Auswirkungen, würde es dennoch zu höheren Drucken kommen, katastrophal wenn der Schlauch explodiert und das Wasser zu der Ganzen Elektronik gelangt. Da der Schlauch-Regner von Gardena stammt wäre es sinnlos den Rest des Wasserkreises von anderen Herstellen

¹⁵ LATRAT Hochdruckreiniger Wasserpumpe Saugfilter für Waschmaschine, 1/2SSaugschlauch Filter mit Doppelschicht Verschlüsselungssieb für Sprühgeräte Ölpumpen Industrieausrüstung : Amazon.de: Garten. URL: https://www.amazon.de/dp/B097ZWX981?ref_=cm_sw_r_cp_ud_dp_VCTSFEVRNS6TWY8WEB (besucht am 04.04.2024).

zu nehmen, die eventuell andere Verbinder und Adapter benötigen. Also wurde der Comfort Flex Schlauch mit einem 13mm großen Durchmesser von Gardena (Artikel-Nr. 18030-20)¹⁶ hergenommen. Dieser ist für einen Druck von bis zu 25 bar geeignet. Er reicht also leicht für diese Anwendung. Dazu ist sein Durchmesser perfekt für die Verbindung mit dem bereits genannten Filter kompatibel. In Abbildung 2.2.6 ist der Schlauch ersichtlich.



Abbildung 2.2.6: 10m Comfort Flex Schlauch 1/2"(13mm)

Darüber hinaus ist mit seiner Länge von 10 Metern für alle Fälle ausgesorgt und auch die Möglichkeit, den Schlauch einfach an einer beliebigen Stelle zu durchtrennen, spricht für den Comfort Flex.

2.2.5 Steuerung

Um den Wasserfluss in den Schlauch-Regner zu steuern eignet sich eine Schlauchkupplung mit Regulierventil, ebenfalls von Gardena (Artikel-Nr. 18266-20)¹⁷. Diese ist in Abbildung 2.2.7 ersichtlich.

¹⁶ Comfort Flex Schlauch 13 mm (1/2"), 10 m | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/schlaeuche/comfort-flex-schlauch-13-mm-12-10-m/967243601.html> (besucht am 04.04.2024).

¹⁷ Kupplung mit Regulierventil | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/ogs/kupplung-mit-regulierventil/970625801.html> (besucht am 04.04.2024).



Abbildung 2.2.7: *Gardena Schlauchkupplung mit Regulierventil*

Das Ventil lässt sich um 90° drehen und somit den Wasserfluss regulieren. Allerdings ist das eingebaute Kugelventil bereits bei einem Drehwinkel von circa 45° schon komplett geschlossen und sperrt den Wasserfluss. Die Auflösung der Einstellmöglichkeiten hält sich also in Grenzen. Des Weiteren muss das Ventil manuell bedient werden, es ist somit nicht automatisierbar, deswegen wird ein Zwischenstück wie z.B. ein Motor benötigt um es automatisiert drehbar zu machen.

Zusätzlich dazu lässt sich der Druck den die DC-Pumpe aufbaut durch einfache Pulsweitenmodulation steuern.

2.2.6 Adapter

Damit nun alle diese Herzstücke miteinander verbunden werden können, sind gewisse Adapter von Nöten. Dabei handelt es sich um folgende Adapter:

Schlauchverbinder Der Schlauchverbinder (Artikel-Nr. 18215-20)¹⁸ wird insgesamt 5-mal gebraucht. Mithilfe dieses Adapters, der in Abbildung 2.2.8 abgebildet ist, kann man den Schlauch-Regner mit dem Transport-Schlauch verbinden, die Schlauchkupplung einbauen und die beiden Anschlüsse der Pumpe mit einen Schlauch verbinden. Aufpassen muss man beim Kauf dieses Bauteils aber, da sich je nach Durchmesser des Schlauches die Dimensionen des Adapters ändern, er aber dennoch gleich aussieht. Deswegen muss man vorsichtig sein, keinen Fehlkauf zu tätigen.

¹⁸ *Schlauchverbinder 13 mm (1/2") - 15 mm (5/8") | GARDENA*. URL: <https://www.gardena.com/at/produkte/bewaesserung/ogs/schlauchverbinder-13-mm-12---15-mm-58/967314001.html> (besucht am 04.04.2024).



Abbildung 2.2.8: Gardena Schlauchverbinder für 1/2"(13mm) Schläuche

Hahnverbinder Da der Outlet und der Inlet der Pumpe (Abbildung: 2.2.4) mit Gewinden versehen sind müssen als Zwischenstück für die Schlauchverbinder noch zwei Hahnverbinder (ebenfalls Gardena Artikel-Nr. 18220-20)¹⁹ eingebaut werden.



Abbildung 2.2.9: Gardena Hahnverbinder für G1/2"Gewinde

Da es sich laut Hersteller der Pumpe um ein G1/2"Gewinde handeln soll, wurden dementsprechende Hahnverbinder bestellt. Allerdings stellte sich heraus, dass die Gewinde an den Anschlüssen der Pumpe keine G1/2"sind. Nach einer Messung ergab sich, dass es schlussendlich M18*1.5 Gewinde sind. Der einfachste Weg diese Gewinde nun auf G1/2"zu transformieren ist mittels 3D-Druck. Also wird ein Adapterstück gedruckt, damit der Hahnverbinder doch an die Pumpe montiert werden kann. Nachteil der Ganzen Sache ist, dass der 3D-Druck etwas undicht ist. Das bedeutet, dass der gedruckte Adapter im Weiteren noch isoliert werden muss.

¹⁹ Hahnverbinder | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/ogs-hahnverbinder/970521301.html> (besucht am 04.04.2024).

2.2.7 Verkalkung

Nach all diesen Spezifikationen kommt die Frage auf, welche Art von Wasser verwendet wird, ob Verkalkung eine Thema ist. Einige auf dem Markt verfügbare Prüfschränke erfordern destilliertes Wasser. Da aber für den Wasserkreislauf dieser Klimakammer keinerlei Bauteile aus Metall, mit Ausnahme der Innenkammer, verwendet werden und das zirkulierende Wasser nicht erhitzt wird, was einer der Hauptgründe für starke Verkalkung ist²⁰, wurde die Annahme getätigt, das sauberes Leitungswasser genauso gut für die Anwendung eingesetzt werden kann. Als Vorsichtsmaßnahme wird empfohlen, den Wasserkreis hin und wieder mit Reinigungs- und Entkalkungsmitteln durchlaufen zu lassen und die Innenkammer zu reinigen, um Ablagerungen zu vermeiden.

2.2.8 Aufbau

Das Einzige, das jetzt noch fehlt ist ein Behälter in dem das Wasser aufbewahrt werden kann, sollte gerade ein Klima ohne Regen realisiert werden. Dieser Behälter wird ganz unten im Gehäuse platziert, um den Schwerpunkt des Gerüsts unten zu halten und somit gegen ein Umkippen der Klimakammer zu steuern. Außerdem soll er gänzlich isoliert und abgeschottet von der Elektronik sein. Das hat natürlich Sicherheitsgründe um die Elektronik zu schützen. Damit das komplette Paket noch einmal zusammengefasst wird und sich so besser vorstellen lässt wird in Abbildung 2.2.10 der Aufbau aller Bauteile verdeutlicht.

²⁰ 020910. URL: <https://www.ikz.de/ikz-praxis-archiv/p0209/020910.php> (besucht am 04.04.2024).

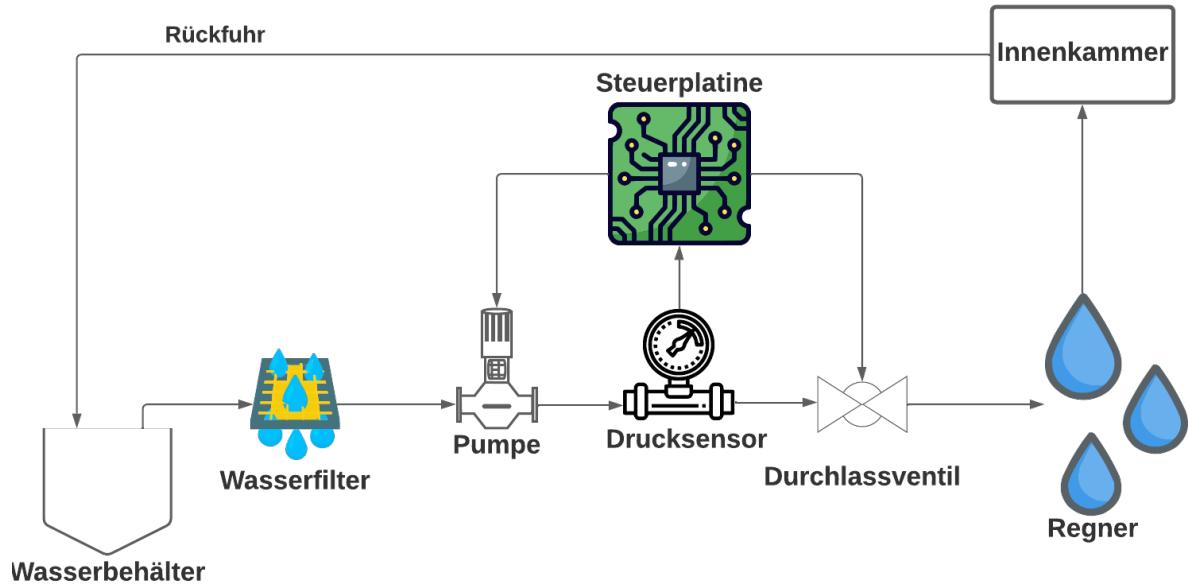


Abbildung 2.2.10: Blockschaltbild des Wasserkreises

Zwei der Bauteile aus Abbildung 2.2.10 wurden noch nicht erwähnt. Das ist einerseits der Drucksensor, welcher dazu dient den Wasserdruck im Schlauch zu messen, bevor er durch das Ventil manipuliert wird. Dadurch wird ein besserer Einblick in die Steuerung verschaffen und es kann rechtzeitig erkannt und verhindert werden, dass es zu Überdrucken kommt, welche den gesamten Kreis und im schlimmsten Fall noch mehr zerstören könnten. Andererseits gibt es noch die Steuerplatine, welche die Bauteile zu Leben erweckt und den Niederschlag steuert, indem sie die Pumpe steuert, den Wasserdruck misst und den Ventilstand kontrolliert. Die genauen Details wie alles gesteuert wird ist in Kapitel 3.4 beschrieben.

3 Hardware

3.1 Spannungsversorgung

Da die Klimakammer im allgemeinen so geplant wurde, dass sie leicht und ohne großen Hardwareaufwand mit weiteren Funktionen erweitert werden kann, wurde auch bei der Spannungsversorgung dieser Ansatz verfolgt. Dabei waren folgende Punkte wichtig:

- **Skalierbarkeit**

Die Leistung der Spannungsversorgung soll gut skaliert werden können

- **Plug-and-Play**

Da die Klimakammer ein eigenständiges Produkt bilden soll, soll die Anbindung an das Stromnetz am besten direkt erfolgen, sie soll einfach an der Steckdose eingesteckt werden können

- **Sicherheit**

Durch die direkte Anbindung an das Stromnetz muss eine gewisse Sicherheit vor der Netzspannung gegeben werden, wie sie z.B. durch CE-Kennzeichnung gewährleistet wird. Auch wird Wert auf die mechanische Sicherheit gelegt, das Powersupply soll also robust sein, und ein Berühren der Netzspannung verhindern.

- **Spannung**

Die Ausgangsspannung sollte sowohl hoch genug sein, um einiges an Leistung ohne riesige Ströme zu ermöglichen, aber gleichzeitig auch niedrig genug um von der Elektronik relativ leicht verwendet werden zu können

Durch frühere Recherchen zu Spannungsversorgungen mit ähnlichen Anforderungen (Hi-Fi System) waren bereits einige mögliche Lösungen bekannt. Dabei erschienen gebrauchte Server-Powersupplies von HP als kostengünstige, aber sehr effektive Lösung. Diese liefern eine Ausgangsspannung von 12V und erreichen unter Netzspannung ($220V \pm 10\%$) eine Effizienz zwischen 91.3% und 92.8%, wie in Abbildung 3.1.1 zu sehen ist.

3.1.1 Server-PSU

Beim „HSTNS-PL29“ handelt es sich um ein Server-Powersupply, welches von HP hergestellt wird, und als „Common Slot Powersupply“ vertrieben wird. Dabei gibt

es mehrere Varianten²¹, welche 460W bis 1500W Ausgangsleistung liefern. Damit wäre der oben genannte Punkt „**Skalierbarkeit**“ befriedigt, denn alle dieser Powersupplies sind miteinander pinkompatibel. Da sie von HP hergestellt und vertrieben werden, und alle mit einem CE-Prüfzeichen versehen sind, ist auch die **Sicherheit** auf der Netzspannungsseite gewährleistet. Die Spannung liegt mit 12V auch im Rahmen, und da die Powersupplies mit einem Kaltgerätestecker ausgestattet sind, ist auch der Punkt „**Plug-and-Play**“ befriedigt.

In der Basisversion der Spannungsversorgung wurden zwei 750W Powersupplies (HPE Generic Part Number: 643932-001) verwendet, welche parallel geschaltet wurden, um bei 12V bis zu 125A (1500W) zu ermöglichen. Die Powersupplies sind so aufgebaut, dass die Parallelschaltung eine N+1 Redundanz ermöglicht. N+1 bedeutet, dass ein Powersupply mehr vorhanden ist, als benötigt wird.

Input Voltage Range (V rms)	100 - 240						
Frequency Range (Nominal) (Hz)	50 - 60						
Nominal Input Voltage (Vrms)	100	120	200	208	220	230	240
Maximum Rated Output Wattage Rating	750	750	750	750	750	750	750
Nominal Input Current (A rms)	8.5	7.0	4.1	3.9	3.7	3.6	3.5
Maximum Rated Input Wattage Rating (Watts)	843	831	811	811	809	808	821
Maximum Rated VA (Volt-Amp)	855	842	822	821	820	819	832
Efficiency (%)	88.9	90.3	92.4	92.5	92.7	92.8	91.3

Abbildung 3.1.1: Auszug aus „HPE Common Slot Power Supplies Quickspecs“

Von genau diesen Powersupplies gibt es von HP keine Datenblätter bzw. andere Informationen. Jedoch gibt es von „Murata Power Solutions“ Powersupplies, welche denen von HP entsprechen, und zu diesen gibt es auch Datenblätter. Das Datenblatt²² stimmt zwar nicht perfekt mit unserem Powersupply überein, aber von der Leistung her ist es nah dran (800W von Murata, 750W von HPE), jedoch ist es die **T**-Revision des Powersupplies. Im Internet wird immer auf das Datenblatt²³ der **P**-Revision verwiesen. Diese haben im Pinout einige Unterschiede. Im folgenden wird das Datenblatt der **P**-Revision verwendet. Es ist also nicht 100% sicher, ob diese Powersupplies identisch sind, die wichtigen Punkte wie z.B. das Pinout stimmen aber

²¹ Hewlett Packard Enterprise. *HPE Common Slot Power Supplies*. URL: https://support.hpe.com/hpsc/public/docDisplay?docLocale=en_US&docId=c04111541 (besucht am 04.01.2024).

²² Murata Power Solutions. *d1u86t-w-800-12-hxxc.pdf*. 2018. URL: <https://www.murata.com/products/productdata/8807027245086/d1u86t-w-800-12-hxxc.pdf?1583754811000> (besucht am 03.01.2024).

²³ Murata Power Solutions. *d1u86p-w-1600-12-hbxd.pdf*. URL: <https://www.murata.com/products/productdata/8807027081246/d1u86p-w-1600-12-hbxd.pdf?1583754811000> (besucht am 04.01.2024).

miteinander überein.

OUTPUT & SIGNAL INTERFACE; PIN ASSIGNMENT; CARD EDGE GOLD FINGERS				PCB BOTTOM (SOLDER) SIDE			
PIN#	Length	NAME/FUNCTION	DESCRIPTION	PIN#	Length	NAME/FUNCTION	DESCRIPTION
53-64	Standard	V1 (MAIN) 12Vdc	12Vdc Main Output	1-12	Standard	V1 (MAIN) 12Vdc	12Vdc Main Output
41-52	Long	PWR RTN	12V Main & 12VSB Output RTN/GND	13-24	Long	PWR RTN	12V Main & 12VSB Output RTN/GND
40	Standard	+VSENSE	12VMain Output +VE Sense	25	Standard	CR_BUS#	Cold Redundant Status Signal
39	Standard	12VSB	Standby Output	26	Standard	-VSENSE	12Vdc Main Output -VE Sense
38	Standard	PS_AO	I2C Address LSB	27	Standard	VIN_GOOD	Incoming Source Status Signal
37	Standard	PWOK	Power OK Signal	28	Standard	I_SHARE	Analogue current share line/bus
36	Long	Signal Return	Standby/Signal Return	29	Standard	PSON#	Power Module Enable/Disable Signal
35	Short	SCL	I2C Serial Clock Line	30	Short	PSKILL	Rapid Enable/Disable for hotswap
34	Short	PRESENT	Power Module Present Signal	31	Short	Not Used	No User Connection
33	Short	SDA	I2C Serial Data Line	32	Short	SMBAALERT#	SMBus_Alert signal

Abbildung 3.1.2: Pinout der **T**-Revision

OUTPUT CONNECTOR AND SIGNAL SPECIFICATION			
Pin#	Function	Pin Type	Description
14-26, 39- 51	+12V_GND/RTN	Power Ground	Power and Standby Return
1-13, 52-64	+12V	Power	12V Output
37	+12VSB	Power	12V Standby Output
38	PSINTERRUPT	Output	Active low, interrupt line for power supply fault & warning detection as per PMBus™ spec
36	PRESENT#	Input	Power Supply Present Signal (shortest pin)
			Combination of three power supply output indicator signals: 1. AC input OK 2. Power Good 3. Power Supply Fault
35	PSOK	Analog output	
34	IMONITOR	Analog I/O	main output current signal typical analog voltage shall be 60.15mV/Amp of main output current.
33	PSON#	Input	Power Supply on/off control signal
32	SCL	Input	SMBus/PMBus Clock
31	SDA	I/O	SMBus/PMBus Data
30	+12V_GND/RTN	Analog I/O	Power Supply Signal Ground
29	N/A	N/A	Reserved; no User connection
28	N/A	N/A	Reserved; no User connection
27	ADDR	Analog input	PMBus Address

Abbildung 3.1.3: Pinout der **P**-Revision

Inbetriebnahme Da in den von HPE gegebenen „Quickspecs“ zwar einige Angaben zum Powersupply gemacht werden, aber es keine genauen Hinweise darauf gibt, wie es verwendet werden kann, wurde dafür das Datenblatt²⁴ des fast baugleichen Powersupplies von Murata verwendet. Darin werden alle Pins erklärt und es wird auch bereits ein fertiges Anschlussdiagramm (Abbildung 3.1.4) gegeben, welches auch zeigt, dass mehrere Powersupplies einfach zusammen gehängt werden können.

Im Anschlussdiagramm sieht man, dass zum Anschalten des Powersupplies „PSON#“ mit „RTN“ (Ground) verbunden werden muss, und „PRESENT#“ mit 12V verbunden werden muss. Als Quelle der 12V bieten bietet sich die Standby-Spannung an, da

²⁴ Murata Power Solutions, d1u86t-w-800-12-hxxc.pdf.

diese vorhanden ist, sobald das Powersupply angeschlossen wird.

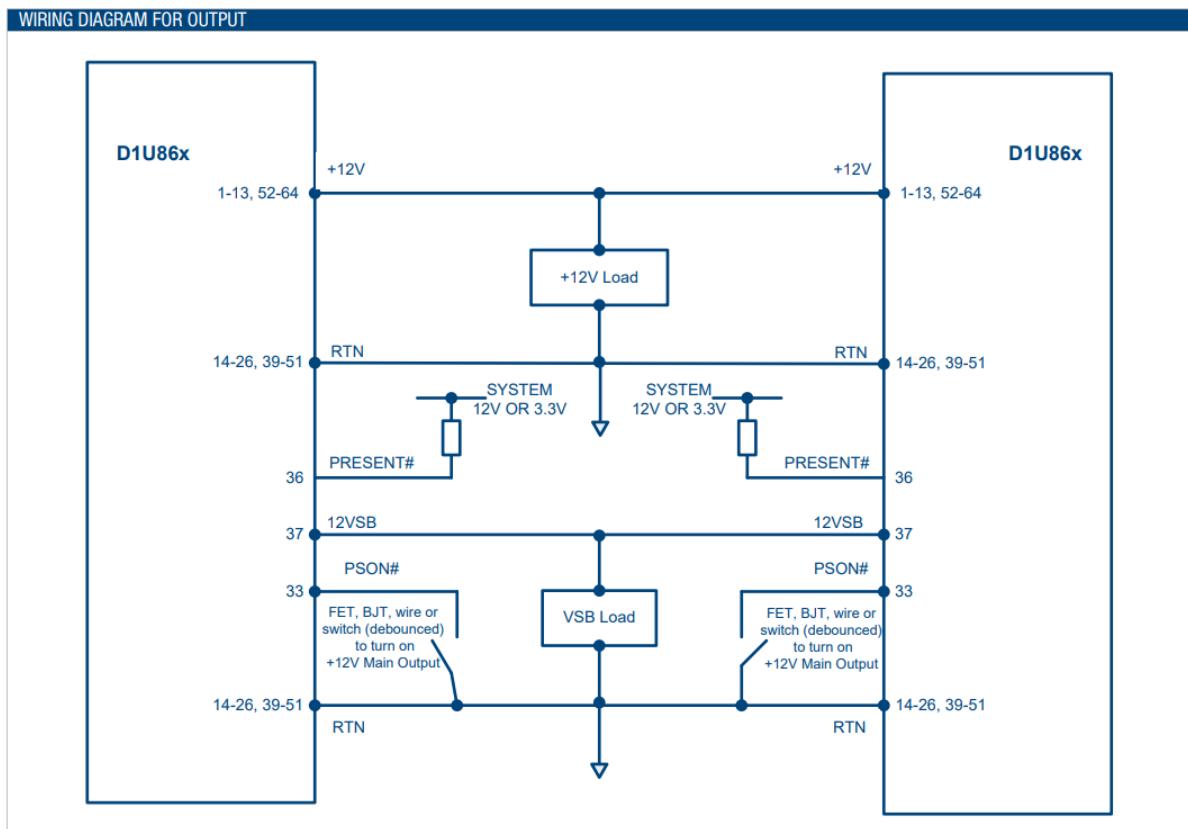


Abbildung 3.1.4: Anschlussdiagramm des Powersupply

Ansteuerung Um das Powersupply zu verwenden muss es nicht angesteuert werden, sondern nur einige Pins verbunden werden. Laut den Datenblättern von Murata²⁵ kann mit dem Powersupply über „PMBus“ Version 1.2 kommuniziert werden, und dadurch Informationen ausgelesen werden, und Parameter wie die Lüftergeschwindigkeit gesetzt werden.. PMBus²⁶ (Power-Management-Bus) ist eine Variante des SMBus (System-Management-Bus), welcher auf dem I²C-Standard basiert. Im Gegensatz zu I²C und SMBus werden beim PMBus bereits einige Commands definiert, sodass generische Treiber möglich sind.

Über Pins kann die Adresse des Powersupplies festgelegt werden. Entweder Analog mit einem Widerstand, oder digital über das High- oder Low-ziehen des „ADDR“-

²⁵ Murata Power Solutions, d1u86p-w-1600-12-hbxdc.pdf; Murata Power Solutions, d1u86t-w-800-12-hxxc.pdf.

²⁶ PMBus. Specification Archives | PMBus. 9. Juni 2010. URL: <https://pmbus.org/specification-archives/> (besucht am 04.01.2024).

Pins.

Im Datenblatt wurde nicht erwähnt, ob bei der Kommunikation mit dem Powersupply noch externe Pullups nötig sind. Jedoch wurde ein Blogpost gefunden, welcher darauf hinweist, dass diese bereits vorhanden sind²⁷. Später stellte sich bei uns heraus, dass externe Pullups nötig waren, da im Powersupply keine vorhanden waren.

Es wurde versucht, mit den Powersupplies von HPE über PMBus zu kommunizieren, aber erfolglos, da diese immer nur „0x55 0xAA“ zurückgaben. Dadurch wurde darauf geschlossen, dass der Fehler beim „PEC“ (Package Error Check) liegt. Dabei handelt es sich um eine 1 Byte große Checksum, welche am Ende der Übertragung vor der „Stop-Condition“ mitgesendet wird. Nach langer Recherche wurde dann auf GitHub ein Repository²⁸ gefunden, in welchem genau auf dieses Problem eingegangen wurde. Dabei wurde der Code welcher auf dem internen Microcontroller gespeichert war ausgelesen, und analysiert. Dabei wurde bemerkt, dass HPE bei diesen Powersupplies wohl auf ein proprietäres Protokoll setzt, denn weder die Commands, noch die Checksum entsprachen der PMBus Spezifikation. In diesem Repository wurde auch ein Beispielcode in Python angeführt, welcher mit dem Powersupply kommuniziert, und einige Werte ausliest und setzt.

In einem anderen Repository²⁹, welches auf dem Erkenntnissen des ersten Repositories basiert, wurde der Python-Code in Arduino kompatibles C++ umgeschrieben. Auf diesem Code basiert die Firmware des Moduls der Spannungsversorgung.

Verteilung der Spannung Die Spannung wird zu den einzelnen Modulen mit einem 12VHPWR-Stecker³⁰ (Abbildung 3.1.5) transportiert. Dieser Stecker wurde entwickelt, um den immer größer werdenden Leistungsaufnahmen von Grafikkarten gerecht zu werden. Der Stecker verfügt über 12 Pins zur Spannungsübertragung, und 4 weitere zur Kommunikation. Diese 4 Pins werden von uns zur internen Kommunikation der Module verwendet.

²⁷ Colintd. *Hacking HP Common Slot Power supplies - 12V@100A*. 14. Okt. 2016. URL: <http://colintd.blogspot.com/2016/10/hacking-hp-common-slot-power-supplies.html> (besucht am 04.01.2024).

²⁸ Richard Aplin. *raplin/DPS-1200FB*. original-date: 2017-03-27T21:25:20Z. 13. Dez. 2023. URL: <https://github.com/raplin/DPS-1200FB> (besucht am 03.01.2024).

²⁹ Simon Lundell. *slundell/dps_charger*. original-date: 2019-08-12T09:06:03Z. 7. Nov. 2023. URL: https://github.com/slundell/dps_charger (besucht am 23.12.2023).

³⁰ Minitek® Pwr Hybrid 3.0 Header. URL: <https://www.amphenol-cs.com/product/101609201240100lf.html> (besucht am 09.03.2024).



Abbildung 3.1.5: 12VHPWR-Stecker von Amphenol

Es hätte auch noch andere Optionen bei den Steckern gegeben, wie z.B. den XT60-Stecker³¹ (Abbildung 3.1.6) aus dem Modellbaubereich. Dieser war dem 12VHPRW aber in drei wichtigen Punkten unterlegen:

- **Maximaler Strom:** Der maximale Strom, den der Stecker dauerhaft aushält, ist mit 30A spezifiziert. Beim 12VHPWR sind es 9.5A pro Kontakt. Damit ergeben sich maximal 57A, also fast doppelt so viel.
- **Benötigt einen weiteren Stecker:** Da nur jeweils ein Kabel für DC+ und DC- vorhanden ist, wird ein weiterer Stecker mit Kabel für die Kommunikation benötigt. Beim 12VHPWR sind vier weitere Kabel für die Kommunikation reserviert
- **Dokumentation unzureichend:** Die Dokumentation des XT60-Steckers ist in vielen Teilen unzureichend, bzw. ist es schwer überhaupt eine Dokumentation zu finden. So variiert der maximale Strom je nach Quelle zwischen 30A bis 120A, und die bei diesem Strom auftretende Temperaturerhöhung wird in keiner Quelle genannt.

³¹ XT60. URL: <https://docs.rs-online.com/4610/A700000008956683.pdf> (besucht am 09.03.2024).



Abbildung 3.1.6: XT60-Stecker

3.1.2 Platine zur Verteilung der Spannung

Die Platine musste folgende Aufgaben erfüllen:

- **Anschalten der Powersupplies**

Beide Powersupplies sollen angeschaltet werden, sobald sie eingesteckt werden

- **Auswählen der Adressen der Powersupplies**

Die Adressen der Powersupplies sollen so ausgewählt werden, dass sie nicht miteinander kollidieren

- **Verbinden der Steuerplatine mit den Powersupplies**

Die Steuerplatine soll so mit den Powersupplies verbunden werden, dass sie möglichst viele nützliche Parameter auslesen kann (Spannung, Strom, etc.)

- **Schnelles Auswechseln der Powersupplies**

Die Powersupplies sollen im Fehlerfall schnell getauscht werden können. Darauf werden Stecker verwendet, anstatt sie anzulöten.

- **Status-LED**

Es soll eine Status-LED implementiert werden, welche anzeigt, ob die 12V-Rail aktiv ist

- **Temperatur der Platine**

Da sehr hohe Ströme transportiert werden, müssen entsprechend dicke Leiterbahnen verwendet werden, um zu gewährleisten, dass die Platine über längere Zeit nicht überlastet und beschädigt wird.

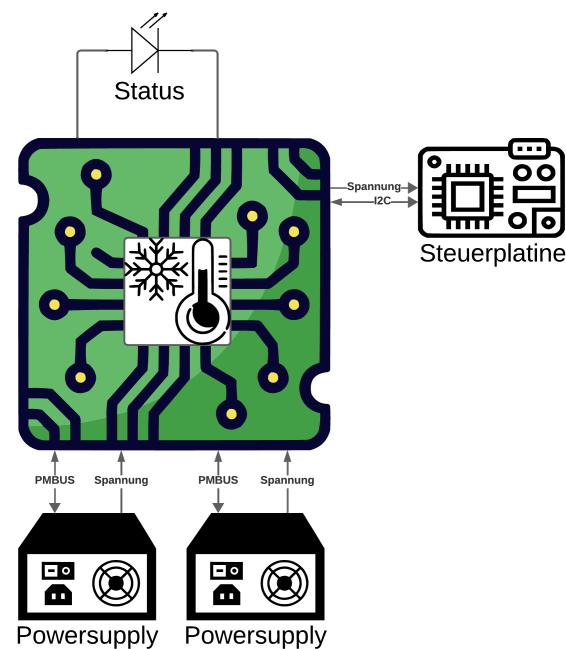


Abbildung 3.1.7: Aufgaben der Platine

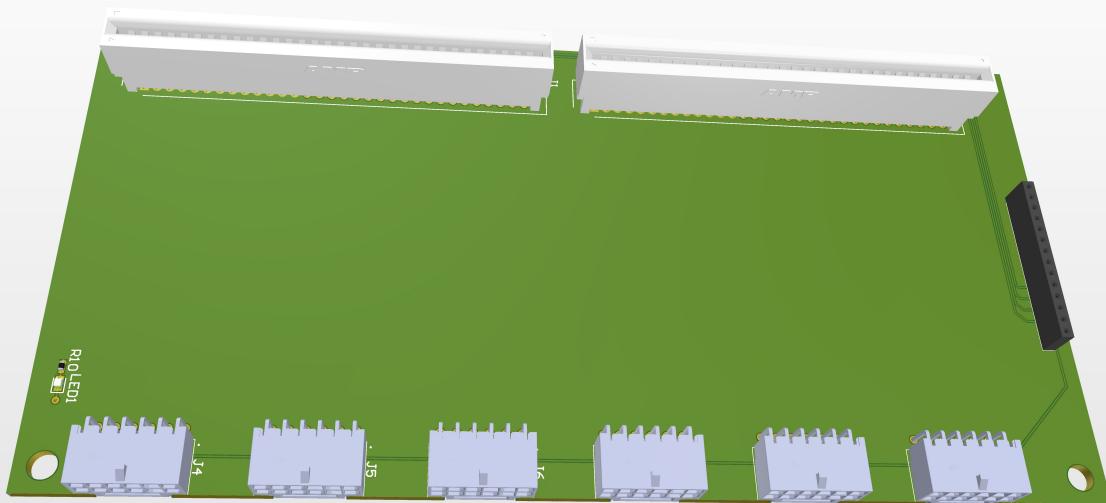


Abbildung 3.1.8: Fertige Platine im Altium 3D-Viewer

Da die Platine sehr hohen Strömen standhalten muss, wurde berechnet wie breit die Traces mindestens sein müssen, um den Strom bei einer akzeptablen Temperaturerhöhung aushalten zu können. Für solche Berechnungen gibt es die IPC2221-Leitlinie, bzw. die neuere IPC2152-Leitlinie mit entsprechenden Formeln entworfen. Da der Einblick in diese Richtlinien aber 675\$ pro Jahr kostet³², verließen wir uns bei der Berechnung auf einen Berechnungsweg, welcher aus den Erkenntnissen der IPC2152-Richtlinie entwickelt wurde³³. Dabei wird zuerst der benötigte Querschnitt der Trace berechnet, bezeichnet als „Ac“:

$$A_c(T, I) := (117.555 * T^{-0.913} + 1.15) * I^{0.84*T^{-0.108}+1.159}$$

Wenn dieser berechnet wurde, kann die Breite der Leiterbahn mit folgender Formel berechnet werden, wobei „Weight“ die Dicke der Leiterplatte in $\frac{Oz}{foot^2}$ ist:

$$Width = \frac{A_c}{\frac{Weight}{1.3}}$$

Mit diesen Formeln wurde dann in Mathcad die benötigte Breite der Traces für den maximalen Strom von 125A und einer maximalen Temperaturerhöhung von 50°C für die Dicke des Kupfers von 35µm, 70µm und 140µm berechnet. Die Ergebnisse betrugen:

$$Width(35\mu m) = 565mm$$

$$Width(70\mu m) = 282mm$$

$$Width(140\mu m) = 141mm$$

Da, um beide Powersupplies auf einer Platine verbinden zu können, sowieso eine Breite der Platine von ca. 200mm benötigt wurde, wurde die Platine mit 70µm dicken Kupfer bestellt. Die Kosten der Platine mit 140µm dicken Kupfer, hätten ca. 200\$ betragen, während die 70µm dicke nur etwa 50\$ kostete.

Bei der fertigen Platine wurde in Altium noch der „Keysight Power Analyzer“ verwendet, um das Design zu verifizieren. Dabei wird eingegeben welche Bauteile Spannung zur Verfügung stellen, und welche dann einen Strom aufnehmen. Da-

³² *Membership Pricing*. IPC International, Inc. 12. Mai 2022. URL: <https://www.ipc.org/ipc-membership-pricing> (besucht am 09.03.2024).

³³ *PCB Trace Width Calculator and Equations*. URL: <https://www.smpe.us/pcb-calculator.html> (besucht am 09.03.2024).

bei wurde in der Simulation die Betriebsspannung mit 12V aus beiden Powersupplies definiert. Es wurde eingestellt dass jeder Stecker 20A benötigt. Insgesamt wird also eine Leistung von 1440W geliefert. Dann berechnet der Analyzer die Stromdichte und Spannungsabfälle auf der Platine. Die Ergebnisse können dann direkt auf der Platine angeschaut werden, und als HTML-File exportiert werden. Standardmäßig kommen im Analyzer Warnungen ab einer Stromdichte von 100A/mm^2 , laut unserer Berechnung sollte dieser Wert für 50°C PCB-Erwärmung bei ca 6.3A/mm^2 liegen. Dieser Wert wird auf der Platine nur an wenigen Stellen überschritten, was aber nicht weiter schlimm ist, da sich die Platine zumindest stellenweise auch etwas mehr als 50°C erwärmen darf, und keine Dauerleistung von 1440W vorgesehen ist. Falls dieser doch benötigt wird, kann die Platine mit einem Lüfter gekühlt werden, was die thermische Situation stark verbessern sollte.

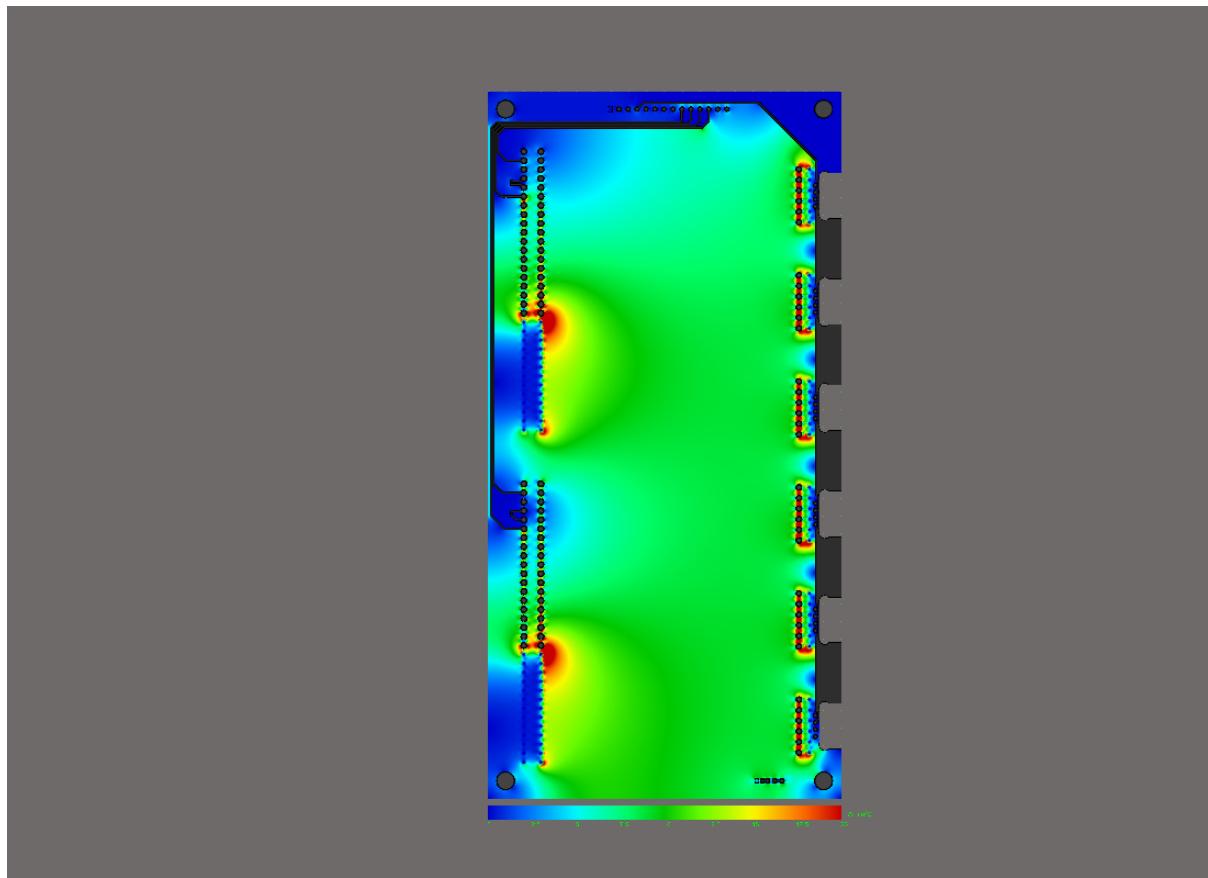


Abbildung 3.1.9: Stromdichte der Platine laut Keysight-PDA (Legende im Bild)

Es können auch „Probes“ platziert werden, welche entweder an einer Stelle messen, oder die Differenz zwischen zwei Stellen.

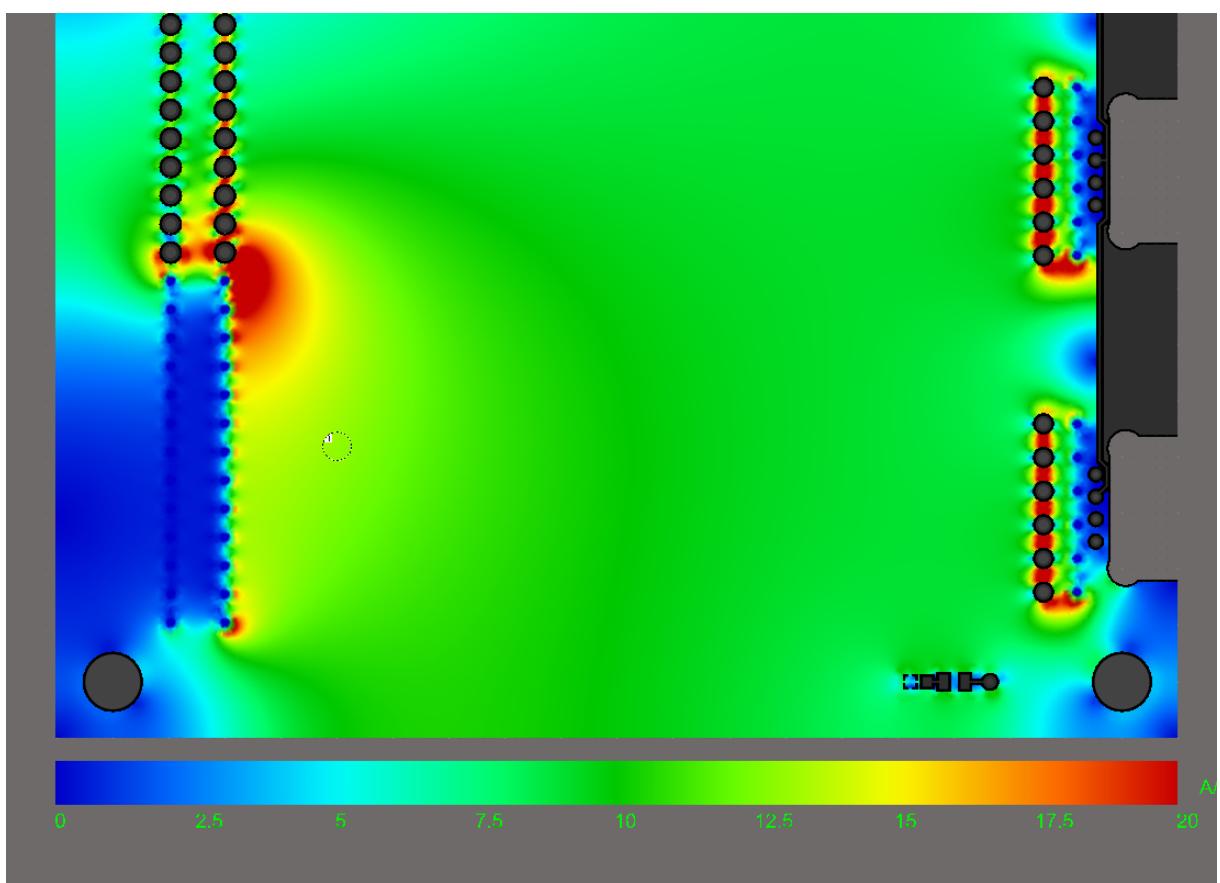


Abbildung 3.1.10: Stromdichte der Platine an einer Stelle

3.1.3 Befestigung im Gehäuse

Zur Befestigung im Gehäuse wurde eine Halterung in Solidworks designed und 3D-gedruckt. Diese ist in Abbildung 3.1.11 zu sehen.

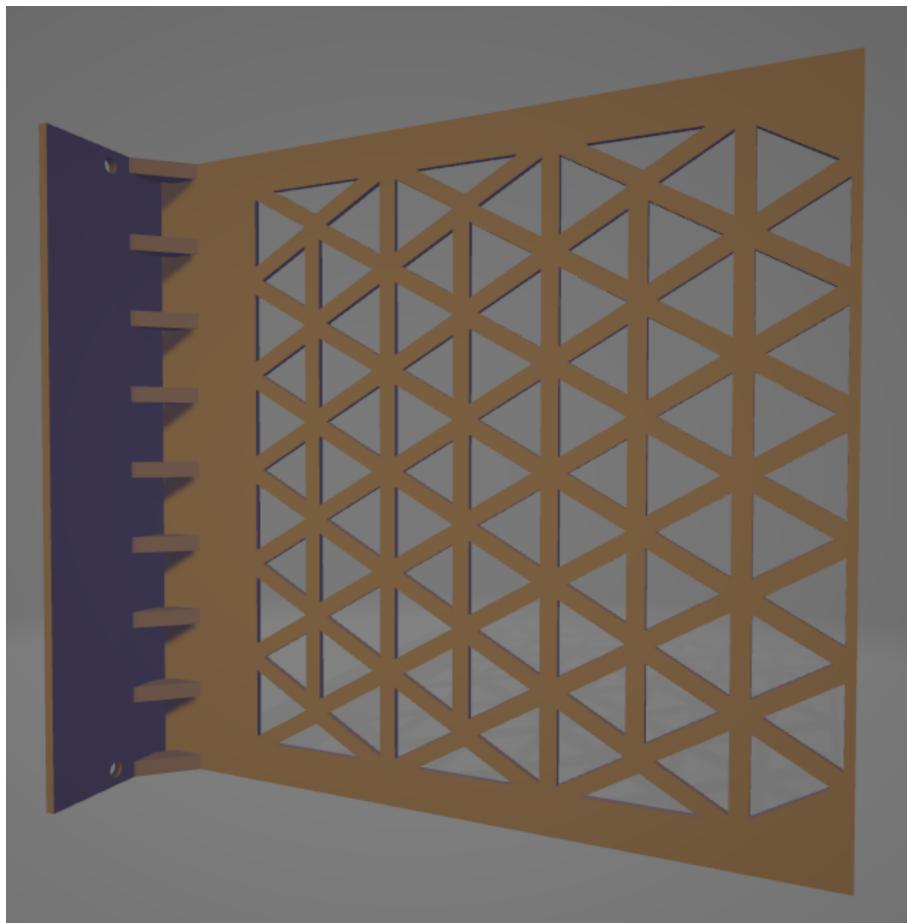


Abbildung 3.1.11: Halterung der Spannungsversorgung

3.2 Zentrale Steuerung

Da das Gesamtsystem modular aufgebaut ist, wird eine zentrale Steuerung benötigt, welche die Arbeit der Module überwacht und koordiniert. Außerdem hostet die zentrale Steuerung eine API, welche als Art „Abstraction-Layer“ zwischen der Steuerung und der Website fungiert, denn damit muss sich die Website nicht mit I²C-Adressen und -Commands usw. beschäftigen, sondern muss nur API-Calls senden. Wie in Abbildung 3.2.1 zu sehen, sendet die zentrale Steuerung Befehle an die Module welche diese dann verarbeiten. Wenn die Website zum Beispiel den aktuellen Wert der Temperatur wissen möchte, sendet sie einen API-Call an die zentrale Steuerung. Diese fordert diesen Wert dann vom Sensormodul an. Das Sensormodul liest den Thermometer aus, und sendet den Wert an die zentrale Steuerung zurück. Diese gibt dann eine Antwort auf den API-Call, in welcher der aktuelle Wert der Temperatur steht.

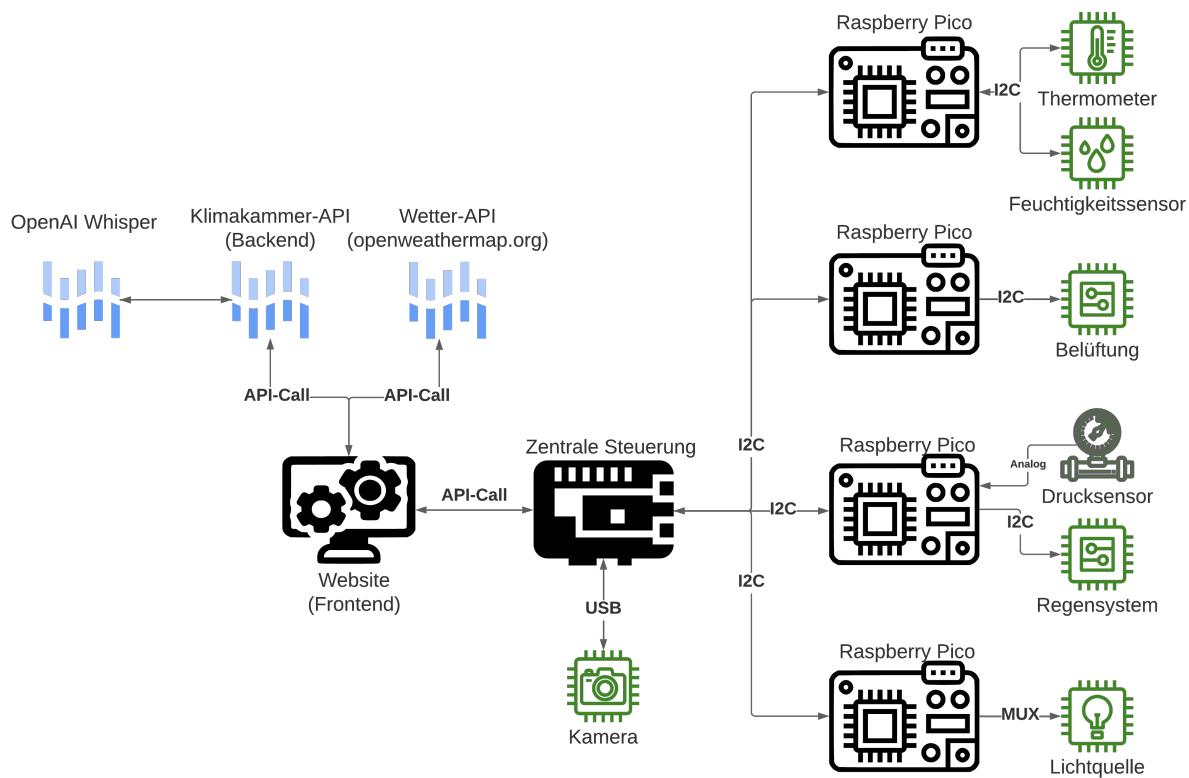


Abbildung 3.2.1: Blockdiagramm des Systems

3.2.1 Raspberry Pi

Für die Durchführung der zentralen Steuerung wurde ein „Raspberry Pi 4 Model B“³⁴ aus Abbildung [3.2.2] mit 4GB Ram verwendet. Zu dieser Entscheidung führten mehrere Gründe:

- **Einfache Verfügbarkeit:** Verglichen mit anderen Single-Board-Computern wie z.B. dem Banana Pi 4³⁵ und dem Orange Pi 5 Pro³⁶, hat der Raspberry Pi eine sehr gute Verfügbarkeit, und kann direkt bei lokalen Händlern bestellt werden.
- **Erfahrung:** Mit dem Raspberry Pi 4 wurden schon Erfahrungen gesammelt, insbesondere beim Entwickeln einer passenden Erweiterungsplatine und dem

³⁴ Raspberry Pi Ltd. *Buy a Raspberry Pi 4 Model B*. Raspberry Pi. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (besucht am 06.01.2024).

³⁵ Banana Pi BPI-M4 Berry with Allwinner H618, 2G RAM, 8G eMMC onboard -Banana Pi open source hardware community, Single board computer, Router, IoT, STEM education. URL: <https://www.banana-pi.org/en/banana-pi-sbcs/167.html> (besucht am 09.03.2024).

³⁶ Orange Pi - OrangePi. URL: <http://www.orangepi.org/html/hardware/computerAndMicrocontrollers/details/Orange-Pi-5-Pro.html> (besucht am 09.03.2024).

Benutzen des I²Cs.

- **Verbreitunggrad:** Der Raspberry Pi 4 ist sehr verbreitet, und hat auch eine sehr aktive Community, welche Software und Hardware für ihn entwickelt. Außerdem gibt es vom Hersteller ein funktionierendes Linux basierendes Betriebssystem, mit speziell auf die Hardware angepassten Anwendungen mit z.B. „Raspi-config“.

Es wurde eine Platine als HAT (Hardware Attached on Top) entwickelt, genauereres dazu im Kapitel „3.2.3 Platine“. Dabei wird das Betriebssystem „Raspberry Pi OS Lite“³⁷ verwendet, welches eine auf den Raspberry Pi angepasste Version von „Debian“³⁸ ist. Dabei wurde die Lite Version (ohne Desktop) über der normalen gewählt, da wir keinen Desktop benötigen, und dieser somit nur unnötige Ressourcen benötigen wurde.

Später wurde im Rahmen des Wahlpflichtfaches „Künstliche Intelligenz“ eine künstliche Intelligenz erarbeitet, welche die Klimakammer per Spracheingabe steuern soll. Wäre dies früher bekannt gewesen, wäre anstatt dem Raspberry Pi ein leistungsfähigeres Gerät, wie z.B. der „NVIDIA Jetson Nano“³⁹ gewählt worden, welcher durch seine 128 „CUDA“-Kerne sicherlich besser performt hätte.

3.2.2 I²C-Extender

Um die einzelnen Module miteinander zu verbinden, wird I²C verwendet. Da dabei für I²C relativ lange Strecken zurückgelegt werden, könnten die parasitären Kapazitäten zu hoch werden. Eine einfache Lösung wäre dann, die Taktgeschwindigkeit runterzustellen, was aber keine zufriedenstellende Lösung ist, da dann die Übertragungen unnötig lange gehen würden. Eine weitere, bessere Lösung ist es, einen „I²C Bus Extender“ zu verwenden. Wir haben einen „P82B715“⁴⁰ von Texas Instruments verwendet. Dieser erlaubt Buskapazitäten von bis zu 3000pF, da auf der gebufferten Seite kleinere Pullupwiderstände verwendet werden können, damit höhere Ströme fließen können und die parasitäre Buskapazität schneller aufladen kön-

³⁷ Raspberry Pi Ltd. *Operating system images*. Raspberry Pi. URL: <https://www.raspberrypi.com/software/operating-systems/> (besucht am 06.01.2024).

³⁸ Debian Org. *Debian – The Universal Operating System*. URL: <https://www.debian.org/> (besucht am 06.01.2024).

³⁹ NVIDIA Corporation. *NVIDIA Jetson Nano für KI-Anwendungen in der Peripherie und Bildung*. NVIDIA. URL: <https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-nano/> (besucht am 06.01.2024).

⁴⁰ Texas Instruments. *p82b715.pdf*. Dez. 2007. URL: https://www.ti.com/lit/ds/symlink/p82b715.pdf?ts=1704579921411&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fde-de%252FP82B715%252Fpart-details%252FP82B715DR (besucht am 06.01.2024).

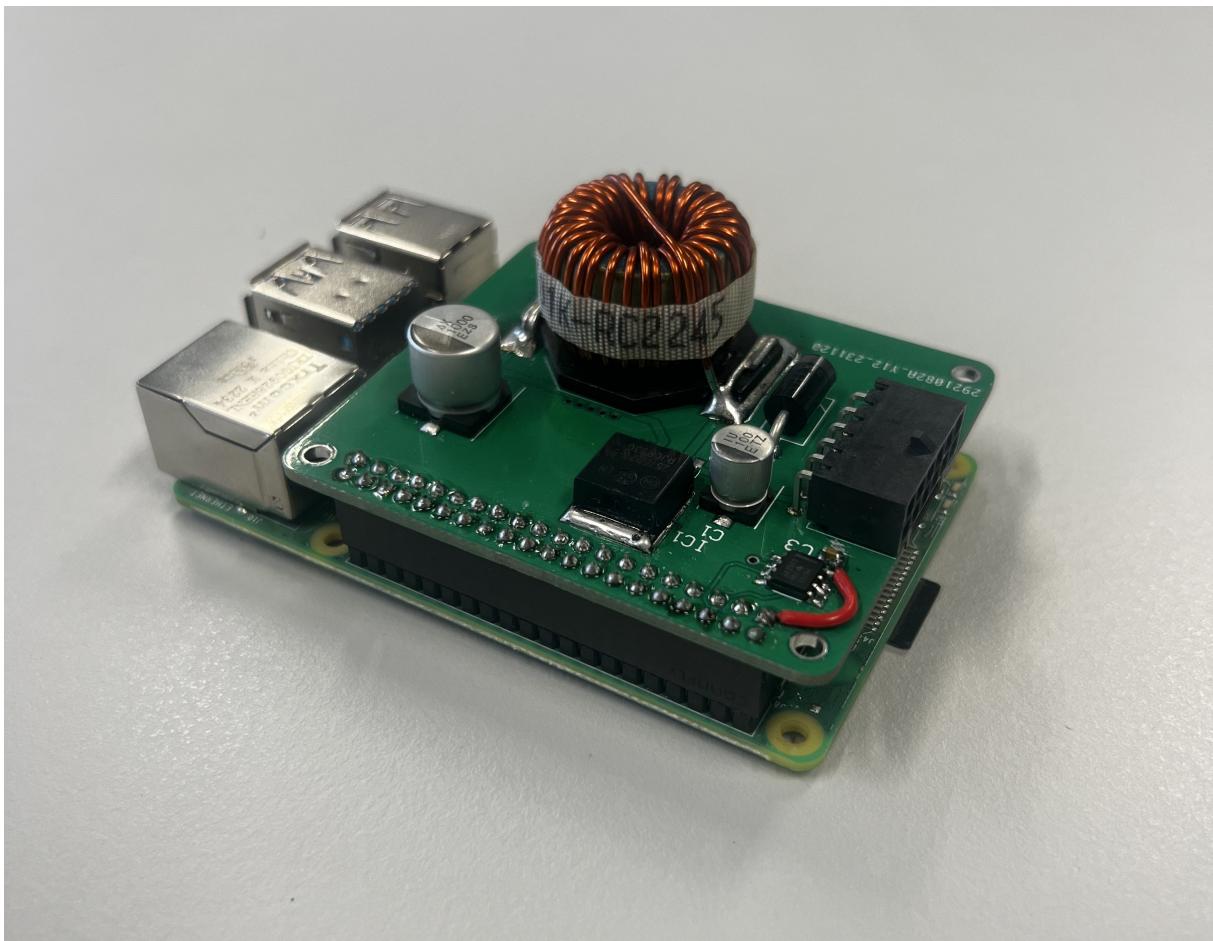


Abbildung 3.2.2: Raspberry Pi 4 Model B inkl. Powersupply aufsatz

nen. Dabei wurde falschlicherweise am Anfang angenommen, dass SDA/SCL und LDA/LCL unterschiedliche Logikpegel verwendet werden können.

3.2.3 Platine

Die Platine wurde als HAT (Hardware Attached on Top) designed, und kann über Pinheader aufgesteckt werden, und über passende Bohrlöcher gegebenenfalls auch noch fest mit dem Raspberry Pi verschraubt werden.

Buck-Converter Da die Systemspannung 12V beträgt und der Raspberry Pi aber nur mit maximal 5V direkt versorgt werden kann, wurde ein Buck-Converter designed, welche die Spannung effizient runterregeln soll. Dabei wurde ein „LM2576“⁴¹ von „onsemi“ verwendet. Dieser wurde verwendet da er mit 2.4€ relativ günstig ist,

⁴¹ LM2576. URL: <https://www.onsemi.com/products/power-management/dc-dc-power-conversion/converters/lm2576> (besucht am 09.03.2024).



Abbildung 3.2.3: NVIDIA Jetson Nano DevKit

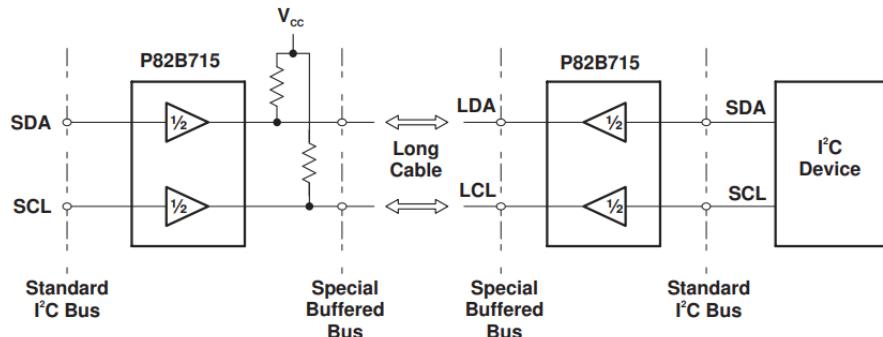


Abbildung 3.2.4: „Figure 4. Minimum Subsystem Diagram“ aus dem Datenblatt

und nur 4 externe Bauteile benötigt. Außerdem funktioniert er mit einer niedrigen Schaltfrequenz von 52kHz, was zwar eine größere Spule benötigt, dafür aber das Platinenlayout nicht so kritisch macht. Bei einer höheren Frequenz in Kombination mit einem Fehler beim Platinenlayout, könnten die elektromagnetischen Emissionen der Platine die Funktionsfähigkeit des Raspberry Pis beeinträchtigen.

Befestigung im Gehäuse Zur Befestigung des Raspberry Pis im Gesamtsystem wurde ein Gehäuse entworfen und 3D-gedruckt. Da der Raspberry Pi bereits ohne Gehäuse relativ warm wurde, wurde ein Gehäuse entworfen, mit welchem eine aktive Kühlung des Raspberry Pis möglich ist. Durch die auf den Raspberry Pi aufgesetzte Platine, konnte auf keiner Seite ein Lüfter platziert werden, welcher direkt auf die CPU gelüftet hätte. Es musste also ein Gehäuse entwickelt werden, welches die Luft über einen Kanal zur CPU hinführt.

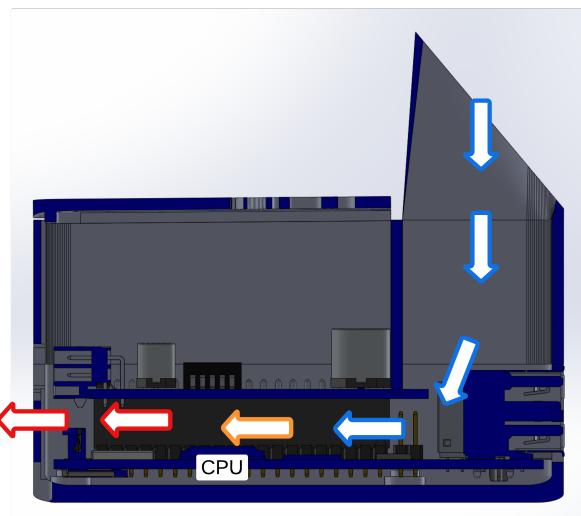


Abbildung 3.2.5: Luftzirkulation im Gehäuse

Dafür wurde von aus dem Internet ein fertiges Gehäuse⁴² runtergeladen und auf unsere Anforderungen modifiziert. Nach der Modifikation wird die Luft mit einem Luftleitblech am vorderen IO-Bereich vorbeigeleitet, über die CPU geführt, und dann am hinteren Bereich des Raspberry Pis wieder ausgeleitet. Bei diesem Design ist es essentiell, dass das Gehäuse überall dort dicht ist, wo keine Luft austreten soll, und ein Lüfter verwendet wird, welcher einen relativ hohen statischen Druck aufbauen kann, weil in diesem Design ein hoher Luftwiderstand besteht. Der statische Druck gibt an, wie gut ein Lüfter Luft gegen einen Widerstand hindurch bewegen kann. Im Querschnitt 3.2.5 ist ersichtlich, wie die Luft sich bewegt.

Anschließend wurden noch einige Test durchgeführt, um das fertige Design zu testen. Dabei wurde der Raspberry jeweils einmal ohne Lüfter, einmal mit konstantem 100% Lüfter und einmal mit 100% Lüfter ab 60°C Kerntemperatur auf allen 4 Kernen voll ausgelastet. Die Ergebnisse sind in Abbildung 3.2.6 zu sehen. Dabei ist zu sehen dass der Raspberry Pi ohne aktive Kühlung nach ca. 5 Minuten 85°C erreicht, die Temperatur bei der er beginnt seinen CPU-Takt zu verringern. Mit ständiger aktiver Kühlung und Kühlung ab 60°C hält er die Temperatur konstant bei ca. 55°C, bei welcher er noch seine volle Leistung liefern kann. Aufgrund dieser Erkenntnisse wurde der Lüfter bei uns so konfiguriert, dass er ab 60°C einschaltet.

⁴² Thingiverse.com. *Raspberry Pi 4 Case* by *anjanathisal*. Thingiverse. URL: <https://www.thingiverse.com/thing:3712499> (besucht am 11.03.2024).

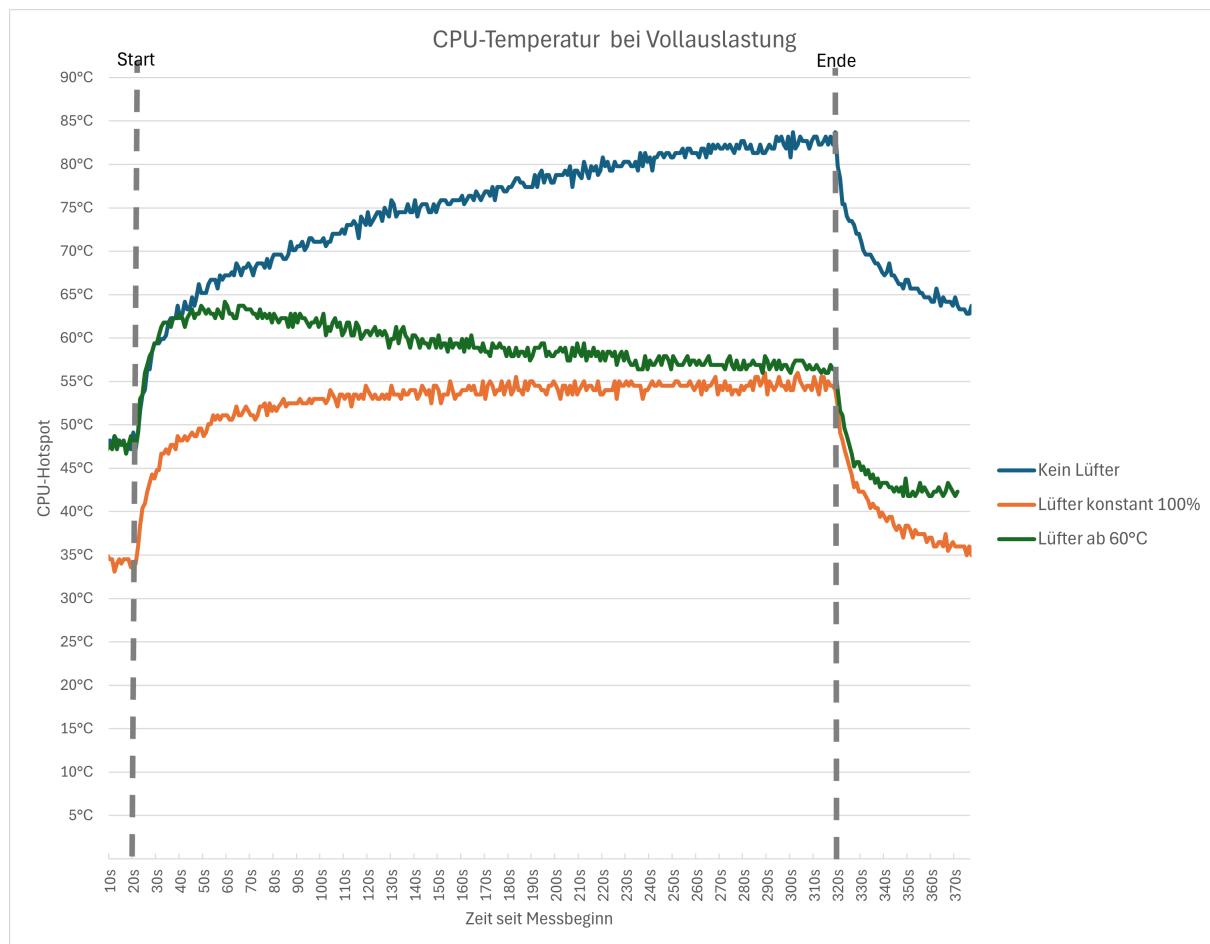


Abbildung 3.2.6: CPU-Temperatur bei Vollauslastung bei unterschiedlichen Lüfterkonfigurationen

3.3 Steuerplatine für alle Module

Da jedes Modul eine eigene Steuerung benötigt, und die Ansprüche an diese Steuerung für jedes Modul sehr ähnlich sind, wurde eine Steuerplatine entwickelt, welche auf die Module aufgesteckt werden kann. Die Anforderungen an diese Platine waren:

- **Kompatibilität mit 12V:** Da die Spannungsversorgung des Gesamtsystems auf 12V basiert, muss auch die Steuerplatine damit kompatibel sein. Jede Steuerplatine muss diese Spannung also auf ihre eigene Betriebsspannung runter regeln.
- **Ausgänge:** Es sollen PWM-fähige Ausgänge zur Verfügung gestellt werden, so dass jeder Ausgang nahezu analog ist.
- **Eingänge:** Einige der Eingänge sollen mittels ADC auch als analoge Eingänge funktionieren, so dass Spannungen gemessen und weiterverarbeitet werden können.
- **Zwei getrennte I2C-Busse:** Da der Raspberry Pi mittels I2C mit den Modulen kommuniziert, aber auch an die Steuerplatine I2C-fähige Bausteine angeschlossen werden sollen können, muss sie zwei getrennte I2C-Busse zur Verfügung stellen. Dabei wird einer im Slave-Modus verwendet, empfängt also die Daten vom Raspberry Pi, und der andere im gegebenenfalls im Slave-Modus, steuert als verschiedene Bausteine über I2C.

Anhand dieser Anforderungen standen zwei verschiedene Mikrocontroller zur Auswahl:

- **ESP32-Familie:** Bei der ESP32-Familie⁴³ handelt es sich um eine leistungsstarke Wireless-SOC-Familie, welche verschiedene Wireless-Standards (Bluetooth, WiFi, etc.) unterstützen. Dabei werden ein bis zwei Tensilica Xtensa LX6 Kerne mit bis zu 240MHz Systemtakt, oder ein RISC-V Kern mit bis zu 160Mhz Systemtakt verwendet. Obwohl diese Mikrocontroller oft viel zu viel Leistung für die verwendete Anwendung haben, werden sie aufgrund ihres Preises verwendet. Dieser ist mit ab 0.93€ sehr günstig, wenn man die sehr umfangreiche Peripherie betrachtet.
- **RP2040:** Der RP2040⁴⁴ ist ein Mikrocontroller welcher von Raspberry Pi ent-

⁴³ *ESP32 Wi-Fi & Bluetooth SoC | Espressif Systems.* URL: <https://www.espressif.com/en/products/socs/esp32> (besucht am 19.03.2024).

⁴⁴ *Raspberry Pi Ltd. RP2040. Raspberry Pi.* URL: <https://www.raspberrypi.com/products/>

worfen wurde, mit dem Ziel einen günstigen, aber leistungsstarken Mikrocontroller auf den Markt zu bringen. Dabei werden zwei Arm Cortex M0+ Kerne mit bis zu 133MHz verwendet. Eine Besonderheit dieses Mikrocontroller sind die PIOs (Programmable Input/Output) Statemachines. In diese können verschiedenste Funktionen programmiert werden, wie z.B. ein neues Interface.

In unserem Fall wurde der RP2040 dem ESP32 vorgezogen, da er günstiger war, und in früheren Versuchen den ESP32 als I2C-Slave zu verwenden, diese immer scheiterten. Außerdem ist der RP2040 bereits als fertiges Modul unter dem Namen Raspberry Pico⁴⁵(Abbildung 3.3.1) verfügbar, welches bereits die gesamte benötigte Peripherie für den Chip beinhaltet (Quarzkristall, Speicher, etc.). Dieses kann über „Castellated Holes“ direkt mit der Platine verlötet werden.

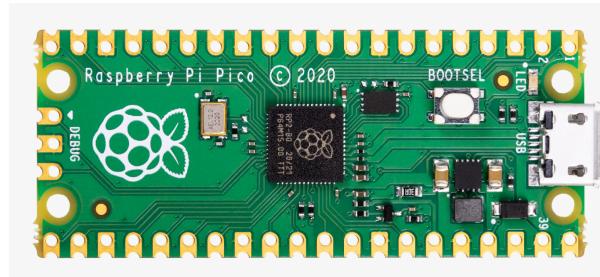


Abbildung 3.3.1: *Raspberry Pico*

Von uns wurde dann eine Platine entwickelt, welche die 12V Systemspannung auf 3.3V runterregelt und dem Raspberry Pico zuführt. Außerdem wurde wie in Kapitel 3.2.2 beschrieben, wieder ein I2C-Extender verbaut, welcher die gebuffernten Signale wieder in I2C-Signale mit „normalen“ Stromstärken umwandelt. Der Schaltplan des Board ist dem Anhang zu entnehmen, und ein Blockschaltbild der Platine ist der Abbildung 3.3.2 zu entnehmen. Darin sind auch bereits beide I2C-Busse eingezeichnet, auf welche später im Kapitel 5.1.1 eingegangen wird.

⁴⁵ rp2040/ (besucht am 19.03.2024).

⁴⁵ Raspberry Pi Ltd. *Pico*. Raspberry Pi. URL: <https://www.raspberrypi.com/products/raspberry-pi-pico/> (besucht am 19.03.2024).

Steuerplatine

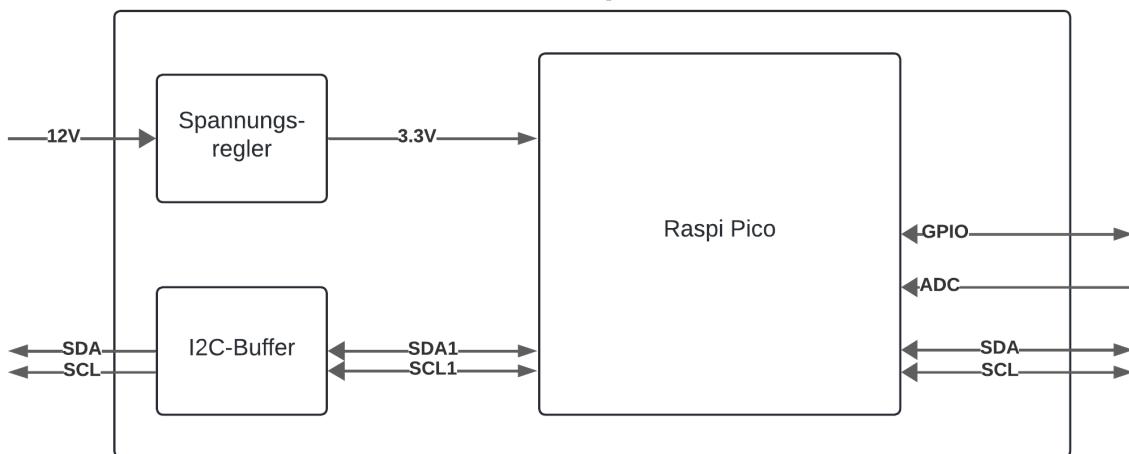


Abbildung 3.3.2: Blockschaltbild der Steuerplatine

Es wurden nicht alle GPIOs des Pico ausgeführt, sondern nur GPIO16, GPIO4 (I₂C0_SDA), GPIO5 (I₂C0_SCL), GPIO17, GPIO18, GPIO19, GPIO21, GPIO26 (ADC0), GPIO27 (ADC1) und GPIO28 (ADC2). Von diesen Pins ist jeder Pin PWM-fähig. Wie in Abbildung 3.3.3 zu sehen, können diese Pins dann über 90° Pinheader mit den Modulen verbunden werden.

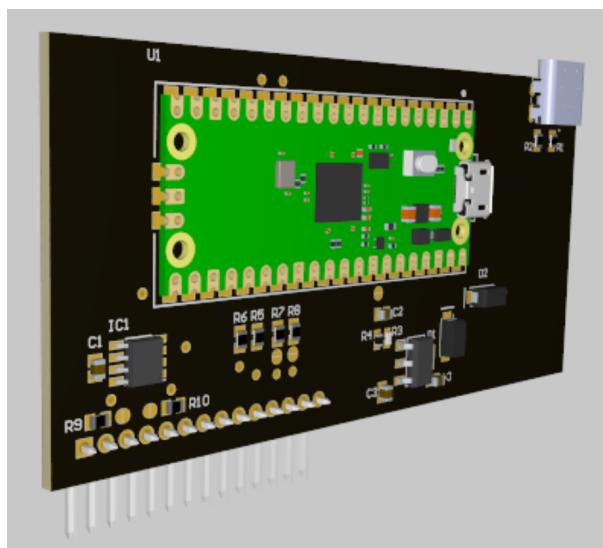


Abbildung 3.3.3: Fertige Steuerplatine mit Raspberry Pico

3.4 Wassersteuerung

Die Steuerung des Wasserkreises kann allgemein in 4 wichtige Punkte zusammengefasst werden:

- **Ansteuerung der Pumpe:** Der DC-Motor der Wasserpumpe muss angesteuert werden. Dabei müssen die Rahmenbedingungen der Pumpe eingehalten werden, da diese eine Nennspannung von 12 Volt hat und Ströme von bis zu 5 Ampere ziehen kann.
- **Ansteuerung des Durchlassventils:** Da das Durchlassventil als jenes nur gesteuert werden kann indem man es per Hand dreht, ist eine automatisierbare Lösung von Nöten, damit das Kriterium, dass die Klimakammer vollautomatisiert arbeiten soll, erfüllt wird.
- **Auslesen des Drucksensors:** Der gewählte Drucksensor (Abbildung 3.4.1)⁴⁶ liefert eine analoge Spannung von 0 bis 5 Volt, welche direkt proportional zum Bereich des Wasserdrucks 0 bis 150 PSI ist. Umgerechnet sind das 0 bis rund 10 bar. Diese Spannung muss umgerechnet werden, und damit kann realisiert werden, dass der Wasserkreis bei Überdruck sofort abgeschaltet wird.
- **Kommunikation mit der Steuerplatine:** Damit die einzelnen Steuerungen wissen, was sie einzustellen haben, muss mit der Steuerplatine (Kapitel: 3.3) kommuniziert werden.



Abbildung 3.4.1: Drucksensor von Amazon

⁴⁶ Edelstahl Drucksensor, G1/4"5V Ausgang 0,5-4,5V/0-5V Druckmessumformer Sensoreingang für Öl Brennstoff Gas Luft Wasser(0-150PSI) : Amazon.de: Auto & Motorrad. URL: https://www.amazon.de/dp/B07YKZ773Z?_encoding=UTF8&ref_=cm_sw_r_cp_ud_dp_AXZY3ANT7NM832QEC63E&th=1 (besucht am 05.04.2024).

3.4.1 Steuerung der Pumpe

Wie bereits erwähnt läuft die Pumpe mit einem DC-Motor und zieht bis zu 60 Watt, bei 12 Volt und 5 Ampere.

$$P = U * I$$

Es wäre möglich die Pumpe über einen High-Power FET zu steuern. Das würde aber bedeuten, dass nur die Zustände Ein und Aus realisiert werden können. Damit die Wassermenge die gepumpt wird skalierbar wird, muss auch die Geschwindigkeit des DC-Motors in der Pumpe skaliert steuerbar sein. Die wohl häufigste und bekannteste Methode dafür ist das Steuern über ein Pulsbreitenmoduliertes (PWM) Signal mithilfe einer H-Brücke. Unter einer H-Brücke versteht man zwei separate serielle Zusammenschaltung zweier Schalter (wahlweise Bipolar-Transistoren, FETs oder manuelle Knöpfe). Zwischen diese beiden Serienschaltung wird ein DC-Motor gehängt. Das Ganze ist in Abbildung 3.4.2 grafisch dargestellt.⁴⁷

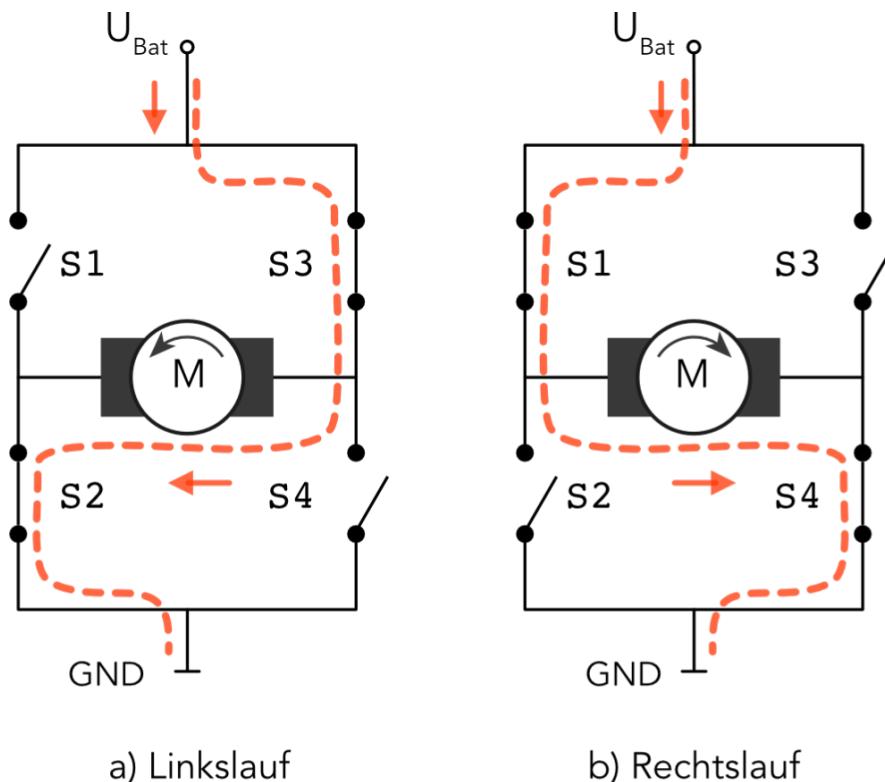


Abbildung 3.4.2: „Schaltung einer H-Brücke“

Dabei sind die Schalter S1 und S4, sowie S2 und S3, miteinander verknüpft. Sie ha-

⁴⁷ Einen Gleichstrommotor mit einer H-Brücke steuern | Rule The Bots. URL: <https://spacehal.github.io/docs/motor/hBridge> (besucht am 04.04.2024).

ben also immer den selben Status (leitend oder nicht-leitend). So kann je nach dem, welches Schalterpaar gerade leitend ist. die Laufrichtung des DC-Motors eingestellt werden wie in Abbildung 3.4.2 schön ersichtlich ist. Das setzt aber voraus, dass wenn ein Schalterpaar gerade leitend ist, das andere auf jeden Fall sperren muss. Sonst hat der Strom keinen designierten Pfad, den er fließen kann. Das könnte im schlimmsten Fall den Motor zerstören.

3.4.2 Steuerung des Ventils

Um das Durchlassventil automatisiert zu steuern, muss genügend Drehmoment entstehen damit das Ventil aus seiner Ruheposition herauskommt. Wie viel Drehmoment benötigt wird, kann experimentell näherungsweise gemessen werden: Es wird ein Hebel mit bekannter Länge gebaut. Das eine Ende des Hebels wird unter das drehbare Teil des Ventils gelegt und auf das andere wird eine Masse mit bekanntem Gewicht gelegt. Nun wird die Masse so lange verändert, oder die Länge des Hebels, bis der Wendepunkt, an dem sich das Ventil gerade noch bewegt, erreicht wird. Nun berechnet man das sogenannte Haltemoment des Ventils: $M = l * m$. Diese Berechnung beruht sich auf der physikalischen Grundlage des Drehmoments: $M = F * r$. Durch die experimentelle Messung ergab sich für das Ventil ein Haltemoment von etwa 0,25Nm. Wir haben uns entschieden einen Schrittmotor zu verwenden, da der Ventilstand so leicht über die Schritte berechnet und eingestellt werden kann. Als problematisch erwies sich dann, dass Hersteller von Schrittmotoren sehr selten ein Nenndrehmoment in ihren Datenblättern bereitstellen sondern wiederum nur das Nennhaltemoment. Deswegen wurde Prof. Christoph Stüttler konsultiert, da sein Wissen über Motoren in der Schule hoch angesehen ist. Er empfiehlt mindestens das 5-fache des benötigten Drehmoments an Haltemoment und um auf der sicheren Seite zu sein das 10-fache. Da aber das ermessene Mindestdrehmoment eher wesentlich größer ist als das tatsächliche, sollte auch das 5-fache kein Problem sein.

Schrittmotor: Zufälligerweise lag noch ein Schrittmotor, welcher eigentlich als Ersatz für einen 3D-Drucker gedacht war, herum. Dieser kommt von ShenZhen Ge-tech Technology und ist in Abbildung 3.4.3 und 3.4.4 ersichtlich.

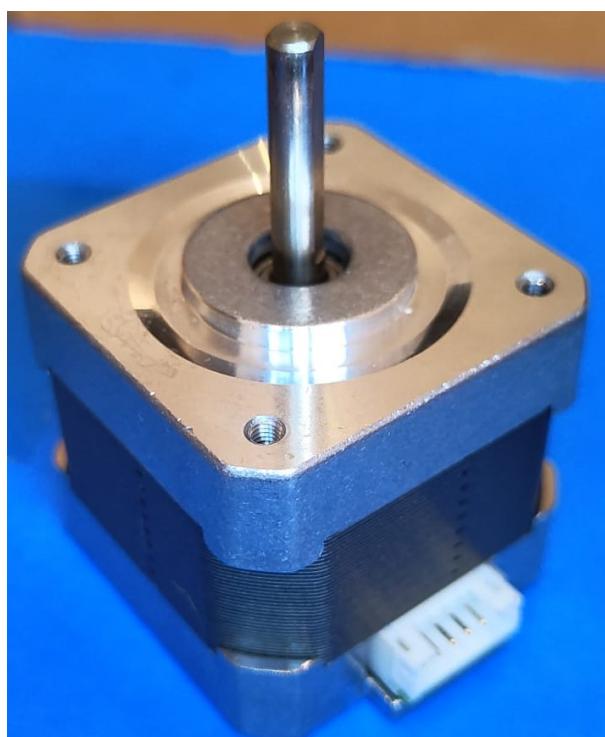


Abbildung 3.4.3: „gewählter Schrittmotor“



Abbildung 3.4.4: „Eckdaten des Schrittmotors“

Dieser Schrittmotor ist ein Zwei-Phasen-Schrittmotor und hat einen Schrittwinkel

von $1,8^\circ$ oder $0,9^\circ$ bei Halbschritten. Soll die Achse also um 90° gedreht werden. sind 50 Schritte bzw. 100 Halbschritte nötig. Damit lässt sich leicht rechnen. Allerdings hat der Schrittmotor nur ein Haltemoment von 0,38Nm. Sollte dieses nicht ausreichen um das Ventil zu bewegen, wird ein kleines Getriebe mittels 3D-Drucker gebaut. Die Nennspannung beträgt 2,77V

3.4.3 Lesen des Drucksensors

Der verbaute Sensor liefert analoge Werte im Bereich von 0 bis 5 Volt. Da der Raspberry Pi Pico allerdings mit 3,3 Volt arbeitet muss die analoge Ausgangsspannung zuerst verkleinert werden. Das kann ganz einfach mit einem Spannungsteiler erreicht werden. Also zwei Widerständen in Serie zur Erde. Wobei die Spannung nach dem ersten wieder abgegriffen wird. Die verkleinerte Spannung beträgt dann:

$$U_{klein} = U_{hoch} * \frac{R_1}{R_1 + R_2}$$

Gewählt wurde: $R_1 = R_2 = 10k\Omega$.

Benutzt man zwei gleich große Widerstände halbiert sich die Spannung. Diese Spannung wird dann an den ADC des Raspberry Pi Picos auf der Steuerplatineangeschlossen, wo der analoge Wert digitalisiert wird und danach der aktuelle Wasserdruck berechnet werden kann. Je nach dem ob der Druck einen unerwünschten Wert erreicht können dann Maßnahmen ergriffen werden. Durch die Halbierung der Mess-Spannung geht am ADC Genauigkeit verloren, da die Referenzspannung aufgrund des Picos 3,3 Volt beträgt, aber die Mess-Spannung nur noch maximal 2,5 Volt erreichen kann. Das bedeutet, in etwa 25% des potenziellen Mess-Bereichs gehen verloren:

$$1 - \frac{2,5}{3,3} = 24,24\%$$

Das könnte verbessert werden, indem der Spannungsteiler so eingestellt wird, dass die verkleinerte Spannung sich 3,3 Volt nähert. Allerdings soll durch die Druckmessung nur verhindert werden, dass der Druck zu hoch wird, deswegen spielt es keine Rolle wenn die Genauigkeit nur 75% der möglichen Genauigkeit erreicht. Es sollen ja nur hohe Drucke abgefangen werden.

3.4.4 Kommunikation mit der Steuerplatine

Die über den Pinheader angeschlossene Steuerplatine steuert den Ablauf und die Treiber-ICs auf der Platine. So wird etwa der aktuelle Wasserdruck eingelesen oder Fehlermeldungen eingelesen. Im Falle des DRV8874 wird auch der Strom im IC durch über einen GPIO gemessen. Außerdem erzeugt sie die PWM-Signale zur Steuerung

der Pumpe und überträgt über den zweiten I2C-Bus die Befehle für den Treiber-IC des Schrittmotors.

3.4.5 Platine

Diese 4 Subsysteme werden auf einer Platine aufgebaut. In Abbildung 3.4.5 sieht man den Schaltplan der Platine.

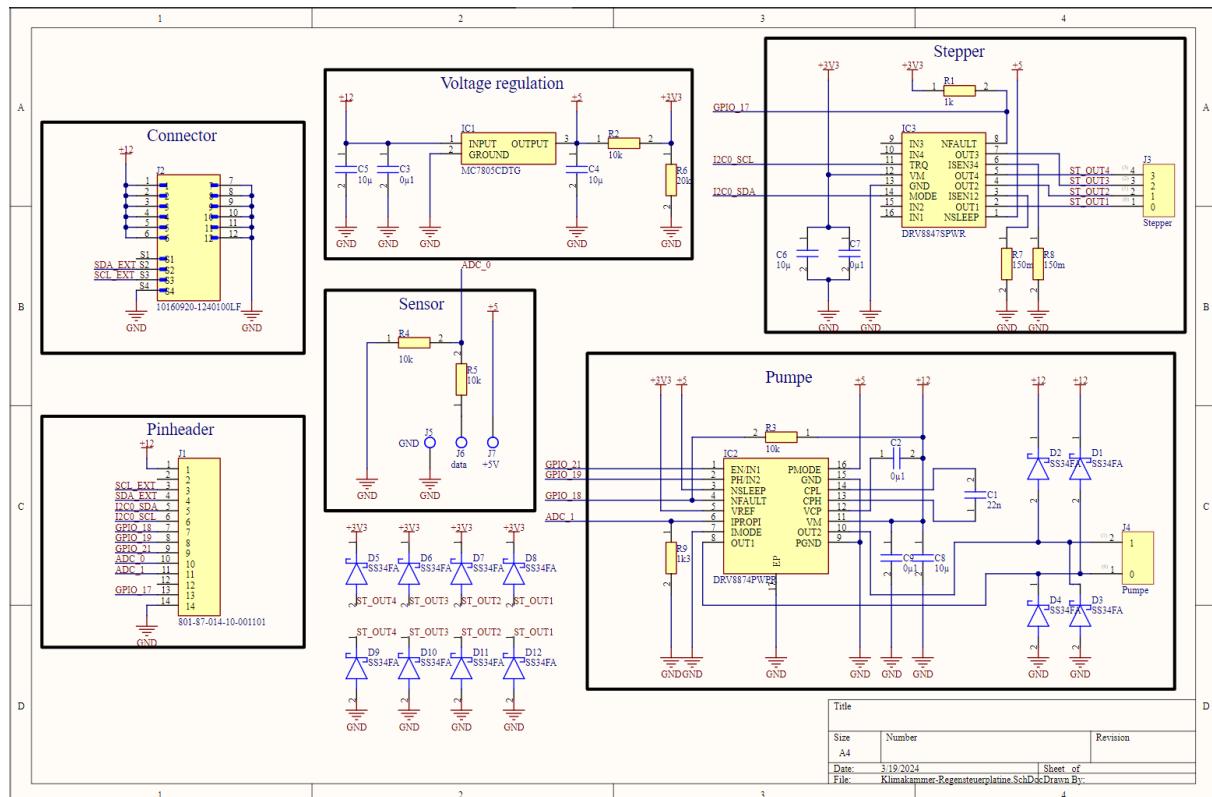


Abbildung 3.4.5: „Schaltplan der Platine zur Steuerung des Wassersystems“

Wie auf dem Schaltplan erkennbar ist, kann die Platine in 6 unterschiedliche Funktionen unterteilt werden:

- **Connector**
- **Pinheader**
- **Voltage regulation**
- **Pumpe**
- **Stepper**
- **Sensor**

Im Folgenden werden diese Teilfunktionen einzeln erklärt.

Connector Beim Connector handelt es sich um den Stecker aus Kapitel 3.1.1. Er sorgt dafür, dass das extended I2C mit den Befehlen des Masters (Raspberry Pi 4B) zur Platine gelangt. Außerdem versorgt er die Platine und die allgemeine Steuerplatine, die über den Pinheader angehängt wird, mit einer Versorgungsspannung von 12 Volt.

Pinheader Durch den Pinheader wird die allgemeine Steuerplatine (Kapitel 3.3) an das System angeschlossen. Das extended I2C wird zur Steuerplatine, für die Bearbeitung der Befehle, weitergeleitet. Es gibt eine weitere I2C-Schnittstelle, von der Steuerplatine kommend, diese kann für verschiedenes eingesetzt werden, im Falle des Wassersystems zur Ansteuerung des Treiber-ICs für den Schrittmotor. Des Weiteren stehen noch ADC-Eingänge und GPIOs zu Verfügung. Alle Anschlüsse die für das Wassersystem gebraucht werden sind in Abbildung 3.4.6 ersichtlich.

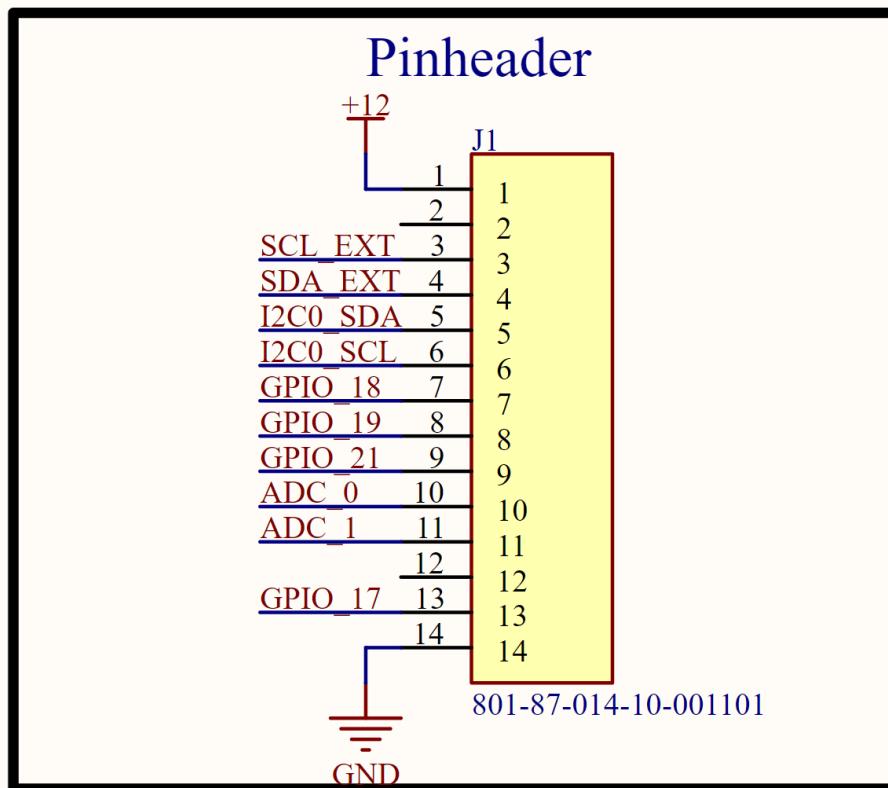


Abbildung 3.4.6: „Benötigte Anschlüsse am Pinheader“

Voltage regulation Die Voltage regulation, zu Deutsch Spannungsregulierung (Abbildung 3.4.7), sorgt dafür, dass die Betriebsspannung von 12 Volt, welche über den Connector zur Platine gelangt, in 5 Volt und 3,3 Volt heruntergeregt wird. Diese

Spannungen werden für die Treiber-ICs der Pumpe und des Schrittmotors benötigt.

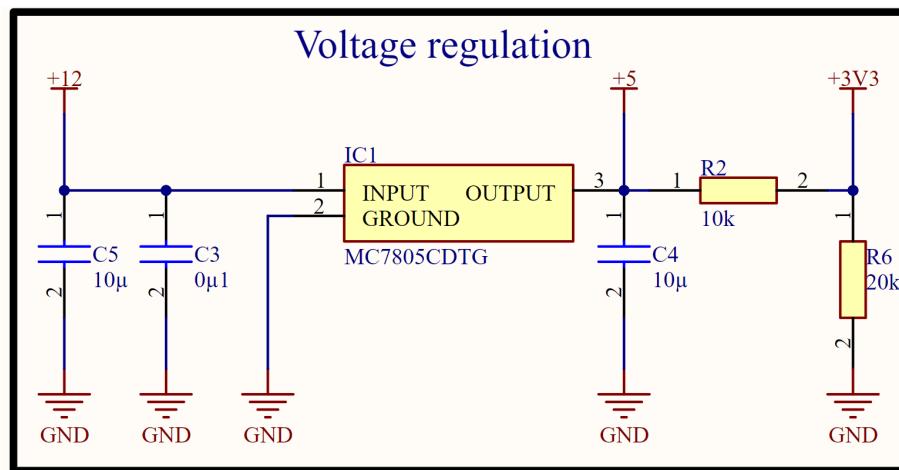


Abbildung 3.4.7: „Spannungsregulierung über den MC7805“

Bei dem in Abbildung 3.4.7 verbauten IC handelt es sich um einen MC7805⁴⁸. MC7805 sind lineare Spannungsregler. Er arbeitet mit einer internen Referenzspannungsquelle. Das 78xx ist die Familienbezeichnung des Spannungsreglers und das xx05 bezieht sich auf die Ausgangsspannung, die der Baustein liefert. In diesem Fall 5 Volt. Aus dem Datenblatt der MC78xx-Familie wird entnommen, dass der Spannungsregler mit Eingangsspannungen von bis zu 35 Volt arbeiten kann (Abbildung 3.4.8). Er ist also für die Betriebsspannung von 12 Volt geeignet. Er liefert Ausgangströme von bis zu 1 Ampere.

MAXIMUM RATINGS (T_A = 25°C, unless otherwise noted)

Rating	Symbol	Value			Unit
		369C	221A	936	
Input Voltage (5.0 – 18 V) (24 V)	V _I		35 40		Vdc
Power Dissipation	P _D	Internally Limited			W

Abbildung 3.4.8: „Auszug aus dem DB der MC78xx-Familie; Seite 1; Maximum Ratings“

Die auf 5 Volt geregelte Spannung wird mit einem Spannungsteiler noch weiter verkleinert, da auch Spannungen von 3,3 Volt gebraucht werden. Wie bereits er-

⁴⁸ MC7800_D-2315963.pdf. URL: https://www.mouser.at/datasheet/2/308/1/MC7800_D-2315963.pdf (besucht am 04.04.2024).

wähnt verkleinert der Spannungsteiler die Spannung um den Faktor $\frac{R_1}{R_1+R_2}$, das bedeutet in diesem Fall um $\frac{20k}{30k}$, also wird die Spannung um ein Drittel kleiner. So werden aus 5 Volt 3,3 Volt. Die Kondensatoren C_3 bis C_5 dienen zur Spannungsstabilisierung.

Pumpe Die Ansteuerung eines Motors mit einer H-Brücke übernimmt im Normalfall ein IC, einfach aus Kosten- bzw. Effizienzgründen und mehr Einstellmöglichkeiten. In unserem Fall ist es der „DRV8874 H-Bridge Motor Driver With Integrated Current Sense and Regulation“ von Texas Instruments.

Anmerkung: Alle Weiteren in diesem Abschnitt genannten Informationen wurden aus dem Datenblatt des DRV8874⁴⁹ entnommen.

In Abbildung 3.4.9 ist das funktionale Blockschaltbild des ICs dargestellt.

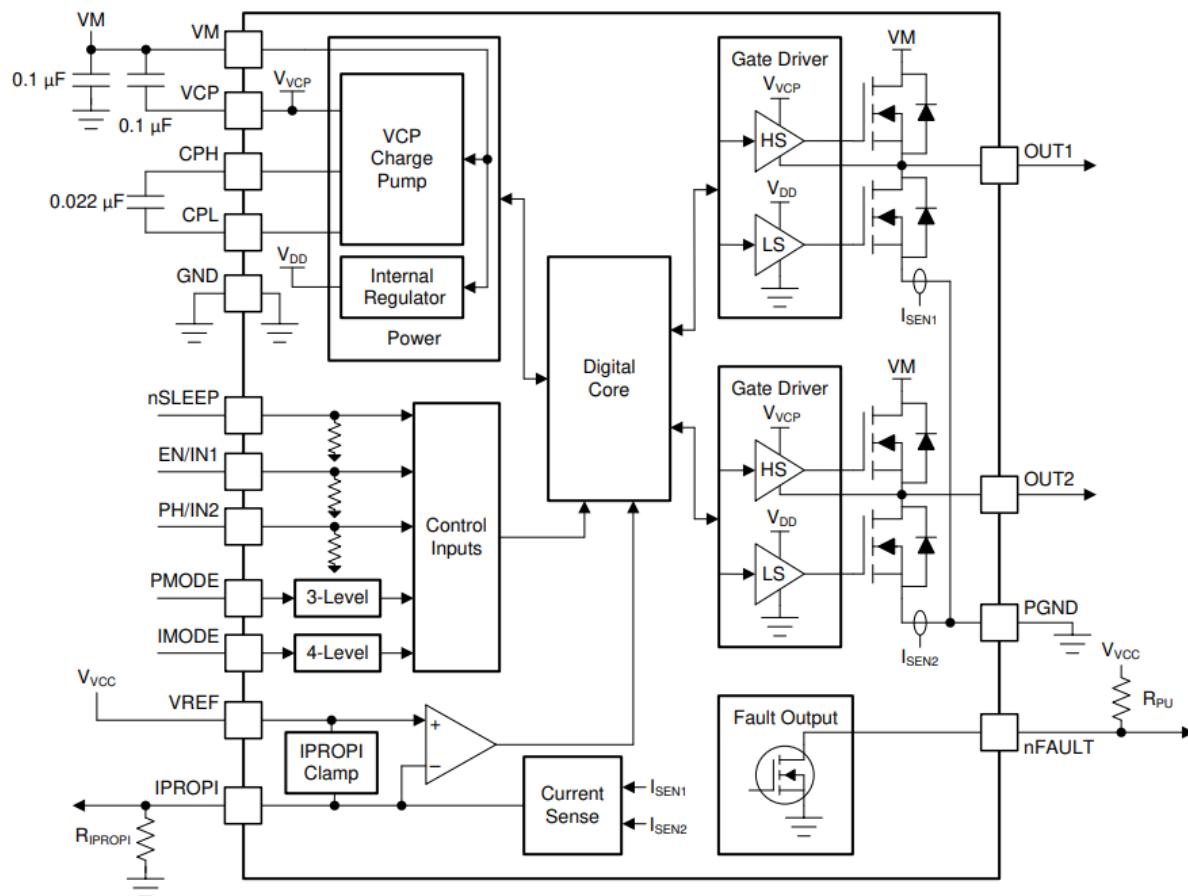


Abbildung 3.4.9: „Auszug aus dem DB des DRV8874; Seite 9, Kapitel 7.2“

⁴⁹ *drv8874.pdf*. URL: https://www.ti.com/lit/ds/symlink/drv8874.pdf?ts=1708861609787&ref_url=https%253A%252F%252Fwww.google.com%252F (besucht am 04.04.2024).

Durch seine große Betriebsspannungs-Toleranz von 4,5 - 37 Volt und dem maximalen Ausgangsstrom von 6 Ampere, ist es dem DRV8874 möglich den hohen Leistungsanforderungen der Pumpe nachzugehen. Außerdem sorgt der $R_{DS_{on}}$ von 200mΩ für minimale Verluste.

Der IC verfügt über zwei relevante Steuerungsmöglichkeiten: Zum Einen kann er über die zwei Pins *EN* und *PH* gesteuert werden. Wobei der EN-Pin (kurz für Enable) High sein muss, um die Steuerung zu ermöglichen. Über den PH-Pin wird dann, je nach logischem Zustand, die Richtung eingestellt, in welche sich der Motor drehen soll. Großer Nachteil dieser Steuerung ist, dass nur die Richtung eingestellt werden kann. Die Geschwindigkeit wird hier vernachlässigt.

Deswegen ist die für diese Anwendung weitaus bessere Art, den Treiber über ein PWM-Signal zu steuern. Dabei werden die Pins *EN* und *PH* umfungiert und durch anlegen eines PWM-Signales an einen der Pins, dreht sich der Motor mit einer Geschwindigkeit, direkt proportional zum Duty-Cycle des PWMs, in die jeweilige Richtung. Vorwärts über den EN-Pin und Rückwärts über *PH*.

Es gäbe noch eine dritte Steuerungsmöglichkeit namens „Independent Half-Bridge“, diese ist für unsere Anwendung allerdings irrelevant.

Über den Pin *PMODE* kann eingestellt werden, welche dieser drei Ansteuerungen verwendet wird. In Abbildung 3.4.10 ist das zusammenfassend noch einmal ersichtlich.

Table 2. PMODE Functions

PMODE STATE	CONTROL MODE
PMODE = Logic Low	PH/EN
PMODE = Logic High	PWM
PMODE = Hi-Z	Independent Half-Bridge

Abbildung 3.4.10: „Auszug aus dem DB des DRV8874; Seite 10, Kapitel 7.3.2, Tabelle 2“

Außerdem bietet der DRV8874 noch weitere Vorteile, wie zum Beispiel eine Fehlererkennung, durch welche der Fehler Indikator Pin *nFAULT* ausschlägt. Alle weiteren Funktionen können in Abbildung 3.4.11 nachgelesen werden.

Pin Functions

PIN		TYPE	DESCRIPTION
NAME	PWP		
CPH	13	PWR	Charge pump switching node. Connect a X5R or X7R, 22-nF, VM-rated ceramic capacitor between the CPH and CPL pins.
CPL	14	PWR	
EN/IN1	1	I	H-bridge control input. See Control Modes . Internal pulldown resistor.
GND	15	PWR	Device ground. Connect to system ground.
IMODE	7	I	Current regulation and overcurrent protection mode. See Current Regulation . Quad-level input.
IPROPI	6	O	Analog current output proportional to load current. See Current Sensing .
nFAULT	4	OD	Fault indicator output. Pulled low during a fault condition. Connect an external pullup resistor for open-drain operation. See Protection Circuits .
nSLEEP	3	I	Sleep mode input. Logic high to enable device. Logic low to enter low-power sleep mode. See Device Functional Modes . Internal pulldown resistor.
OUT1	8	O	H-bridge output. Connect to the motor or other load.
OUT2	10	O	H-bridge output. Connect to the motor or other load.
PGND	9	PWR	Device power ground. Connect to system ground.
PH/IN2	2	I	H-bridge control input. See Control Modes . Internal pulldown resistor.
PMODE	16	I	H-bridge control input mode. See Control Modes . Tri-level input.
VCP	12	PWR	Charge pump output. Connect a X5R or X7R, 100-nF, 16-V ceramic capacitor between the VCP and VM pins.
VM	11	PWR	4.5-V to 37-V power supply input. Connect a 0.1-µF bypass capacitor to ground, as well as sufficient Bulk Capacitance rated for VM.
VREF	5	I	External reference voltage input to set internal current regulation limit. See Current Regulation .
PAD	—	—	Thermal pad. Connect to system ground.

Abbildung 3.4.11: „Auszug aus dem DB des DRV8874; Seite 3, Kapitel 5, Pin-Functions“

Die Beschaltung des DC-Motor Treibers DRV8874 (Abbildung 3.4.12) folgt aus den Beschreibungen des Datenblatts.

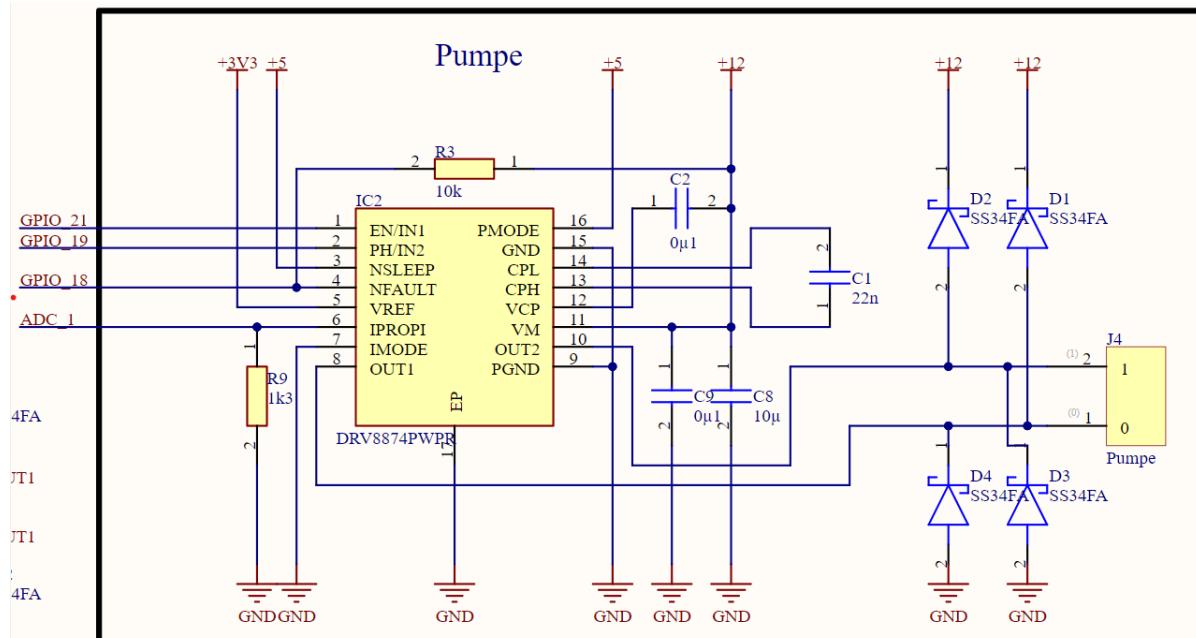


Abbildung 3.4.12: „Beschaltung des DRV8874“

Das mit J4 gekennzeichnete Bauteil, stellt die Schraubklemme dar, an welche die Pumpe schlussendlich angesteckt werden soll. Die 4 Zener-Dioden sind Freilaufdioden. Die zwei Kondensatoren C_8 und C_9 am Pin VM dienen zur Spannungsstabilisierung der Betriebsspannung des ICs. Der 22nF Kondensator C_1 wird im Datenblatt verlangt (Abbildung 3.4.13). Dasselbe gilt für den 100nF C2 (Abbildung 3.4.14).

Pin Functions				
PIN		TYPE	DESCRIPTION	
NAME	PWP			
CPH	13	PWR	Charge pump switching node. Connect a X5R or X7R, 22-nF, VM-rated ceramic capacitor between the CPH and CPL pins.	
CPL	14	PWR		

Abbildung 3.4.13: „Auszug aus dem DB des DRV8874; Seite 3; Kapitel 5; Pin Functions“

VCP	12	PWR	Charge pump output. Connect a X5R or X7R, 100-nF, 16-V ceramic capacitor between the VCP and VM pins.
-----	----	-----	---

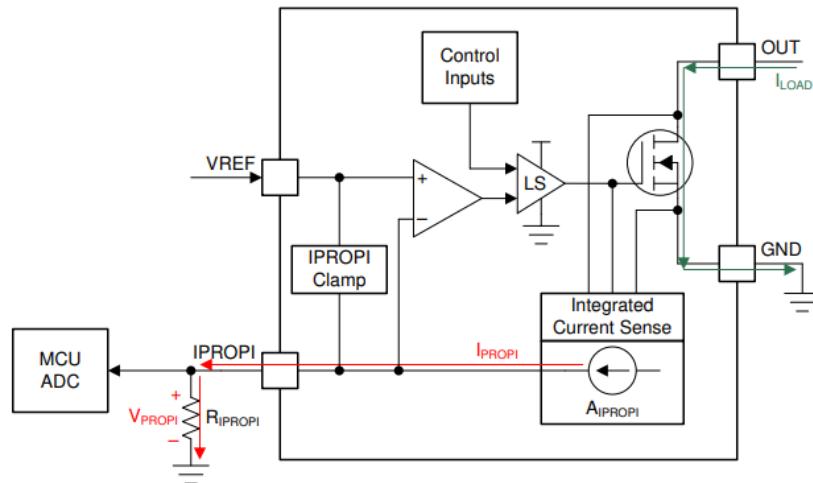
Abbildung 3.4.14: „Auszug aus dem DB des DRV8874; Seite 3; Kapitel 5; Pin Functions“

Der Widerstand R_3 ist ein simpler pull-up Widerstand für den Pin $nFault$.

Der Output $IPropi$ ermöglicht Strommessung, mittels eines Shunt-Widerstandes, innerhalb des ICs über einen ADC. Dafür muss aber die korrekte Spannung anliegen. Deswegen wird der Widerstand R_9 oder $RIPropi$ eingebaut, welcher dafür sorgt, dass der Spannungswert an $IPropi$ nie den erlaubten Maximalwert erreicht. Der Widerstand wird laut Datenblatt wie in Abbildung 3.4.15 dimensioniert.

The corresponding IPROPI voltage to the output current can be calculated by [Equation 2](#).

$$V_{IPROPI} (V) = I_{IPROPI} (A) \times R_{IPROPI} (\Omega)$$



Copyright © 2017, Texas Instruments Incorporated

Abbildung 3.4.15: „Auszug aus dem DB des DRV8874; Seite 12; Kapitel 7.3.3.1; Current Sensing“

Da der ADC auf dem Pico der Steuerplatine läuft darf die Spannung am I_{propri} 3,3 Volt nicht überschreiten. Der Strom $I_{I_{propri}}$ wird im Bereich von 0 bis 3 mA empfohlen (Abbildung 3.4.16).

6.3 Recommended Operating Conditions

over operating temperature range (unless otherwise noted)

			MIN	NOM	MAX	UNIT
V_{VM}	Power supply voltage	VM	4.5	37	37	V
V_{IN}	Logic input voltage	EN/IN1, MODE, nSLEEP, PH/IN2	0	5.5	5.5	V
f_{PWM}	PWM frequency	EN/IN1, PH/IN2	0	100	100	kHz
V_{OD}	Open drain pullup voltage	nFAULT	0	5.5	5.5	V
I_{OD}	Open drain output current	nFAULT	0	5	5	mA
$I_{OUT}^{(1)}$	Peak output current	OUT1, OUT2	0	6	6	A
I_{IPROPI}	Current sense output current	IPROPI	0	3	3	mA
V_{VREF}	Current limit reference voltage	VREF	0	3.6	3.6	V
T_A	Operating ambient temperature		-40	125	125	°C
T_J	Operating junction temperature		-40	150	150	°C

(1) Power dissipation and thermal limits must be observed

Abbildung 3.4.16: „Auszug aus dem DB des DRV8874; Seite 4; Kapitel 6.3; Recommended Operating Conditions“

Beim Einsetzen dieser Werte in die Formel aus Abbildung 3.4.15 ergibt sich der minimale Widertand:

$$\frac{3,3V}{3mA} = 1,1k\Omega$$

Es wurde aber ein Widerstand mit $1,3\text{k }\Omega$ genommen, da das der nächstgrößere im Magazin bei Walter Rauch verfügbare Widerstand war.

Stepper Die Steuerung des Schrittmotors übernimmt der IC „DRV8847S Dual H-Brigde Motor Driver“ von Texas Instruments. Alle Weiteren, in diesem Abschnitt, folgenden Informationen stammen aus dem Datenblatt des ICs⁵⁰. Dieser IC kann wegen den 2 integrierten H-Brücken entweder 2 DC-Motoren oder einen 2-Phasen Schrittmotor angesteuert werden. Siehe Funktionales Blockschaltbild 3.4.17.

⁵⁰ *drv8847.pdf*. URL: https://www.ti.com/lit/ds/symlink/drv8847.pdf?ts=1712239200629&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV8847 (besucht am 05.04.2024).

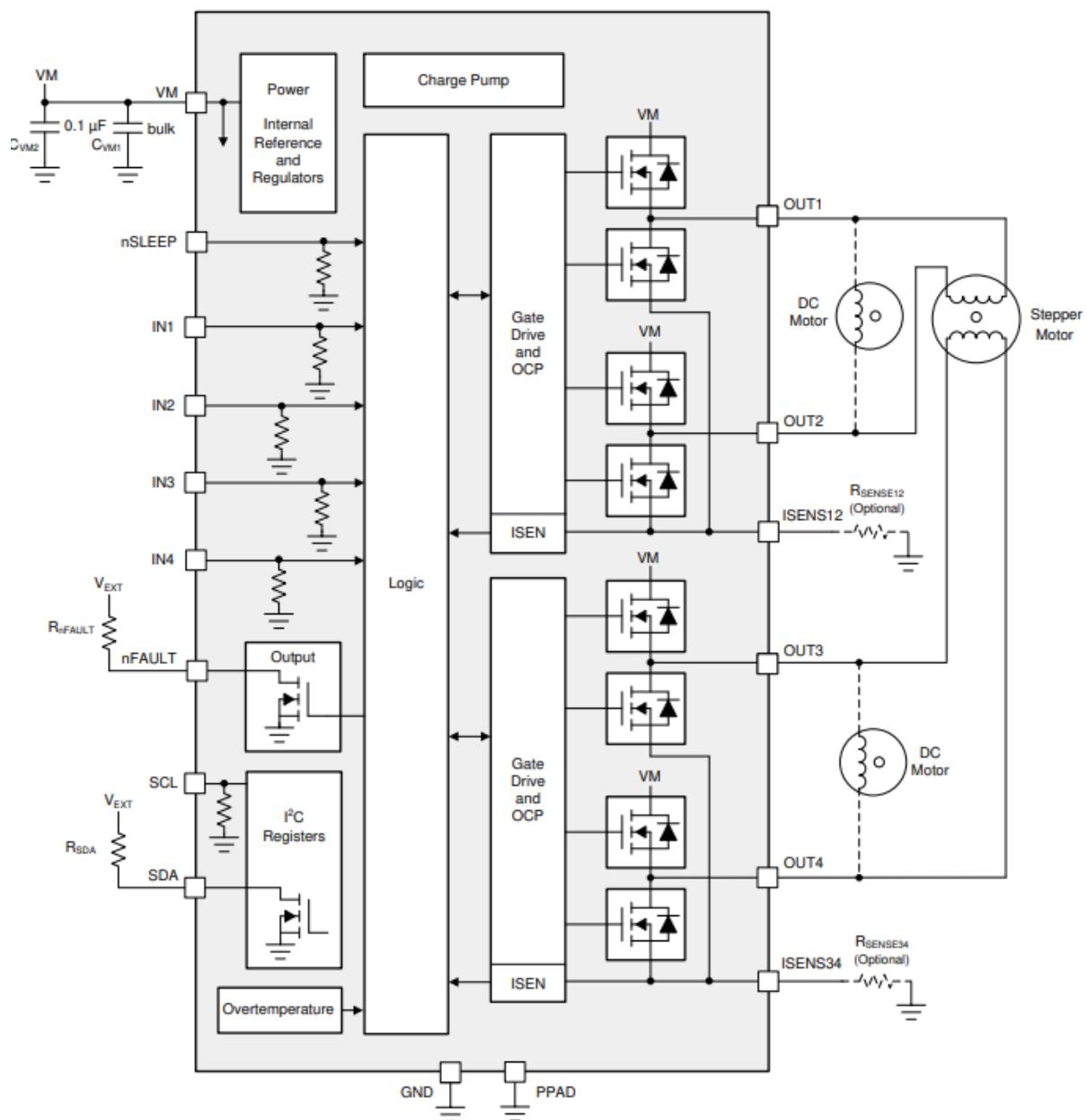


Abbildung 3.4.17: Auszug aus dem DB des DRV8847; Seite 16; Kapitel 7.2; Figure 7-2

Dieser IC wurde vor Allem deswegen gewählt, weil er, in der DRV8847S Ausführung, die Möglichkeit bietet, per I²C gesteuert zu werden. Dafür muss der IC aber richtig beschalten werden (Abbildung 3.4.18).

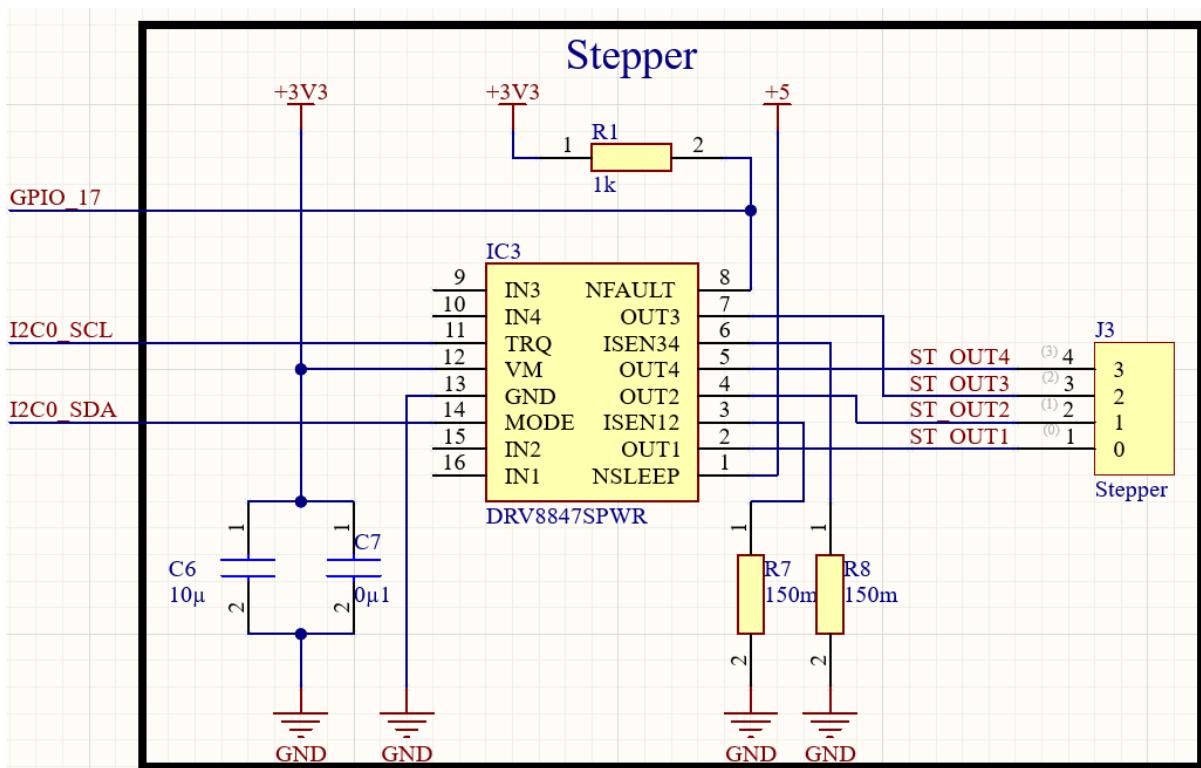


Abbildung 3.4.18: Beschaltung des Schrittmotors inklusive Treiber IC

Die Inputs werden nicht verbunden, weil die Eingänge der H-Brücken über die Pins *SCL* und *SDA* (I₂C) gesteuert werden. Da der DRV8847S eine Version des DRV8847 ist sind diese Pins im Package als *TRQ* und *MODE* gekennzeichnet. Der Pin *NFAULT* schlägt bei internen Fehler aus und wird über einen pull-up Widerstand an die Steuerplatine weitergeleitet. Genaueres siehe Absatz Pumpe.

Der Pin *NSLEEP* wird auf 5V gezogen, damit verhindert wird, dass der IC in den Low-Power Mode geht (Abbildung 3.4.19).

nSLEEP	1	15	1	1	Sleep mode input. Set this pin to logic high to enable the device. Set this pin to logic low to go to low-power sleep mode
--------	---	----	---	---	--

Abbildung 3.4.19: Auszug aus dem DB des DRV8847; Seite 5; Kapitel 5; Tabelle 5.1

Am Pin *VM* wird die Spannung für den Motor angelegt und über zwei Glättungskondensatoren geglättet, wie im DB gefordert (Abbildung 3.4.20).

VM	12	10	12	PWR	Power supply. Connect the VM pin to the motor power supply. Bypass this pin to ground with a VM-rated 0.1- μ F (ceramic) and 10- μ F (minimum) capacitor.
----	----	----	----	-----	---

Abbildung 3.4.20: Auszug aus dem Db des DRV8847; Seite 5; Kapitel 5; Tabelle 5.1

Die Pins *OUT1* bis *OUT4* sind die Ausgänge, an die der Schrittmotor angeschlossen wird, wobei *OUT1* und *OUT2* Phase 1 bilden und *OUT3* und *OUT4* Phase 2. Außerdem werden die Ausgänge noch mit Freilaufdioden bestückt (Abbildung 3.4.21).

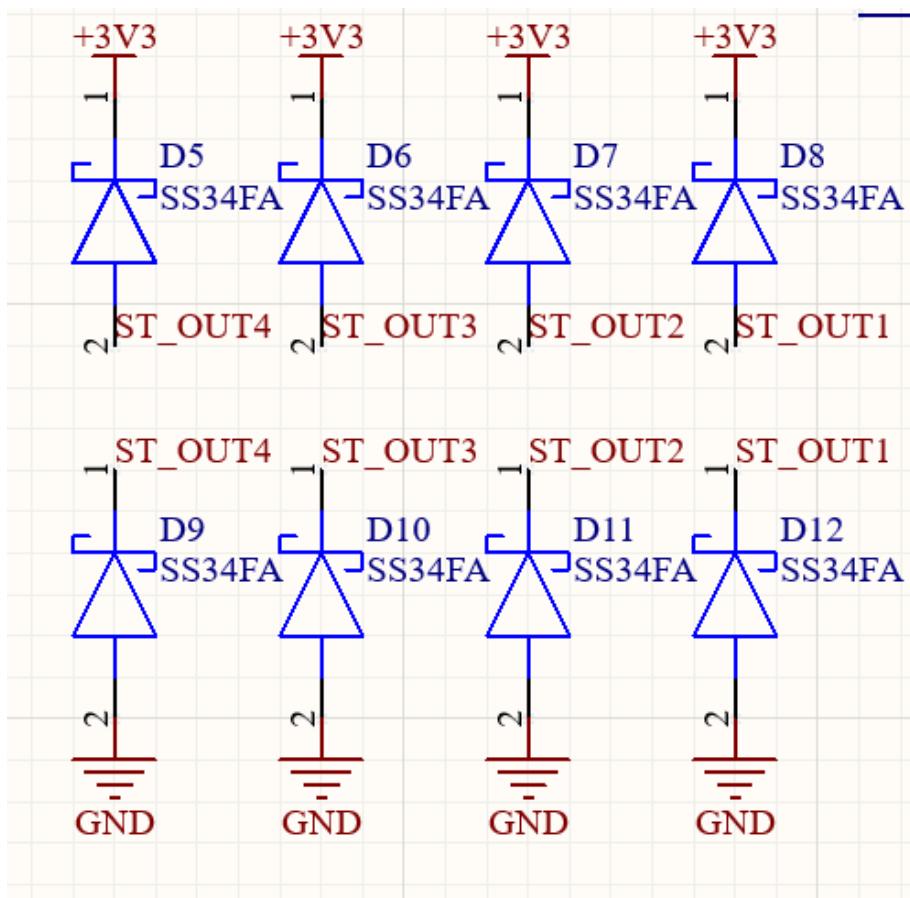


Abbildung 3.4.21: Freilaufdioden für die Anschlüsse des Schrittmotors

An die Pins *ISEN12* und *ISEN34* können Widerstände zur Stromregulierung angeschlossen werden. Dieser Strom wird mittels der in Abbildung 3.4.22 dargestellten Formel berechnet.

$$I_{TRIP} = \text{Torque} \frac{V_{TRIP}}{R_{SENSExx}}$$

Abbildung 3.4.22: Auszug aus dem DB des DRV8847; Seite 28; Kapitel 7.3.4;
Formel 1

Wobei der Faktor Torque 100% ist und somit vernachlässigt werden kann. Für VTRIP kann eine Spannung von 150mV angenommen werden (siehe Abbildung 3.4.23).

PWM CURRENT CONTROL (ISEN12, SEN34)					
V _{TRIP}	ISENx trip voltage	Torque at 100% (TRQ = 0)	140	150	160 mV
		Torque at 50% (TRQ = 1)	63.75	75	86.25 mV
t _{BLANK}	Current sense blanking time			1.8	μs
t _{OFF}	Current control constant off time			20	μs

Abbildung 3.4.23: Auszug aus dem DB des DRV8847; Seite 9; Kapitel 6.5;
Electrical Characteristics

Damit also ein Maximalstrom von 1A erreicht wird werden für beide Widerstände 150mΩ gewählt.

Sensor Die Beschaltung des Drucksensors ist sehr simpel (Abbildung 3.4.24). Die Betriebsspannung des Sensors wird mit 5 Volt verbunden, die von der Spannungsregulierung kommen. Die Erden werden miteinander verbunden. Und das Datensignal muss den in Kapitel 3.4.3 erklärten Spannungsteiler durchlaufen, damit es den ADC oder den Pico auf der Steuerplatine nicht zerstört.

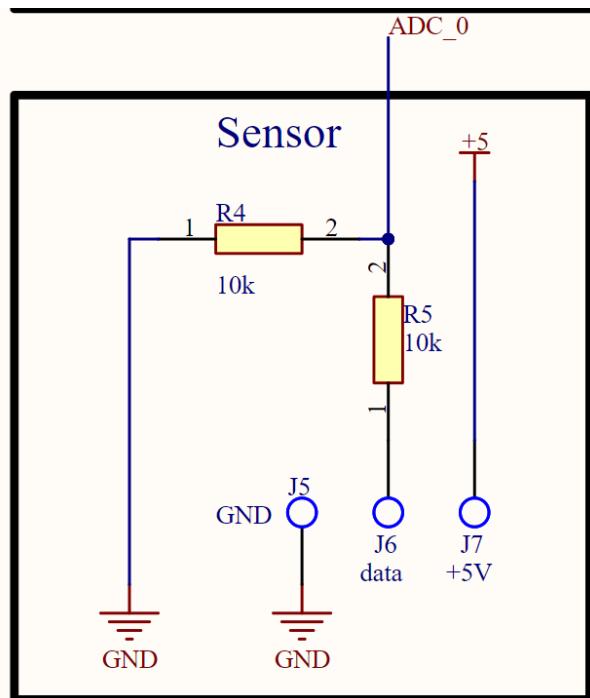


Abbildung 3.4.24: „Beschaltung des Drucksensors“

PCB Um die Pumpe und den Schrittmotor mit der Platine zu verbinden werden kleine Schraubklemmen von Amphenol⁵¹ verwendet (Abbildung 3.4.25). Im Gegensatz zum Sensor welcher direkt an die Platine gelötet wird, soll so verhindert werden, dass durch schlagartige Bewegungsimpulse die Lötpads weggerissen werden. Für die Pumpe wird eine Schraubklemme mit 2 Anschlüssen verwendet und für den Schrittmotor eine mit 4 Anschlüssen.

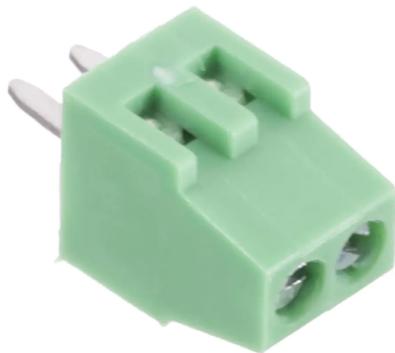


Abbildung 3.4.25: „Schraubklemme mit 2 Anschlüssen von Amphenol“

⁵¹ VN02A1500000G Amphenol Anytek | Mouser. Mouser Electronics. URL: <https://www.mouser.at/ProductDetail/649-VN02A15000J0G> (besucht am 04.04.2024).

Layout Diese Funktionsblöcke müssen nun alle möglichst effizient miteinander verbunden werden. Das heißt Glättungskondensatoren so nah wie möglich an den jeweiligen Pin, I2C-Leitungen so kurz wie möglich. In Abbildung 3.4.26 ist das Layout der Platine dargestellt.

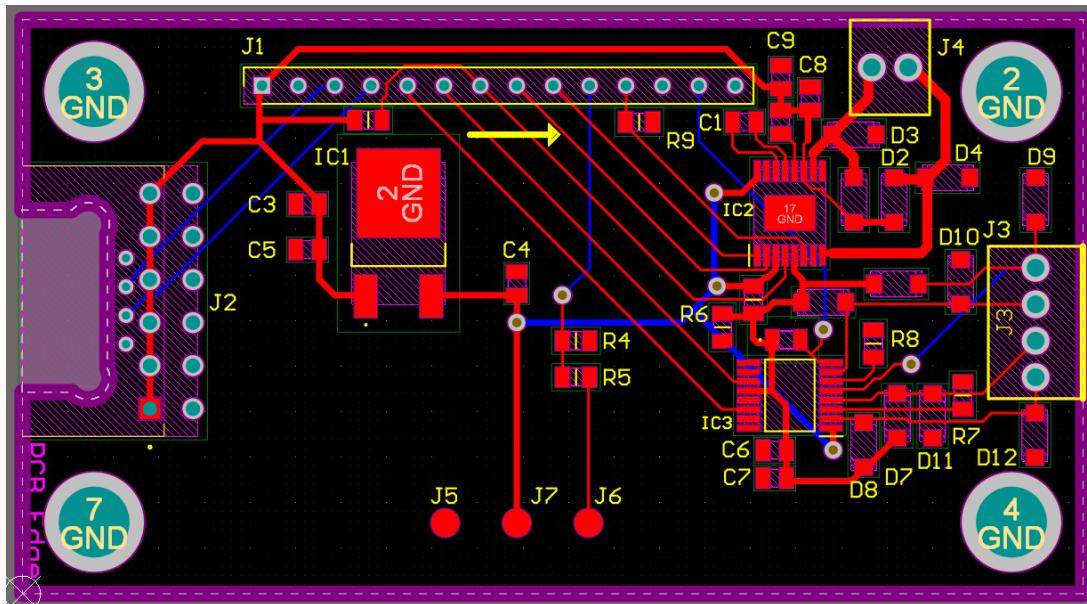


Abbildung 3.4.26: „Board-Layout der Platine“

In Abbildung 3.4.27 ist ein 3D-Modell der Platine ersichtlich. Links kann man den Stecker sehen, durch welchen die Platine mit Strom versorgt wird. An den Pinheadern wird die Steuerplatine angeschlossen. Und an die Schraubklemmen jeweils die Pumpe und der Schrittmotor. Die Kabel des Sensors werden auf die Lötpads J_5 bis J_7 gelötet. Die Maße der Platine sind 75 mal 40 Millimeter.

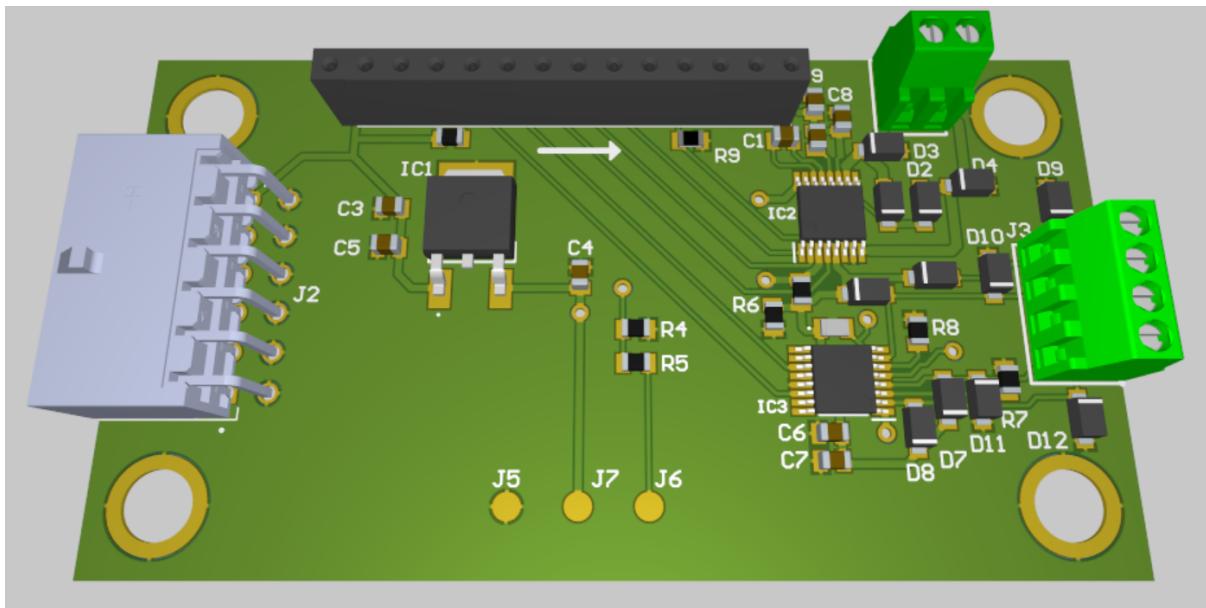


Abbildung 3.4.27: „3D-Modell der finalen Platine“

3.5 Sonnensystem

In einer praxisnahen Umgebung darf eine Form von sonnenähnlicher Beleuchtung nicht fehlen. In diesem Bereich hat uns Pascal Berchthold von der Firma Sanlight maßtragend unterstützt. Unter anderem hat er nützliches Wissen vermittelt, aber auch 2 Q1W-DIM Leuchten von Sanlight zur Verfügung gestellt.

3.5.1 Theorie

Das Sonnenspektrum weist so gut wie jede Wellenlänge von Licht auf. Dieses Spektrum nachzustellen stellt sich als äußerst schwer dar, da keine passende all-in-one Lösung am Markt erhältlich ist, denn solche Leuchten finden in sonst keinem Bereich einen wirklichen Nutzen. Es können jedoch mehrere Leuchten verwendet werden, um die Spektren der Lampen zu vermischen.

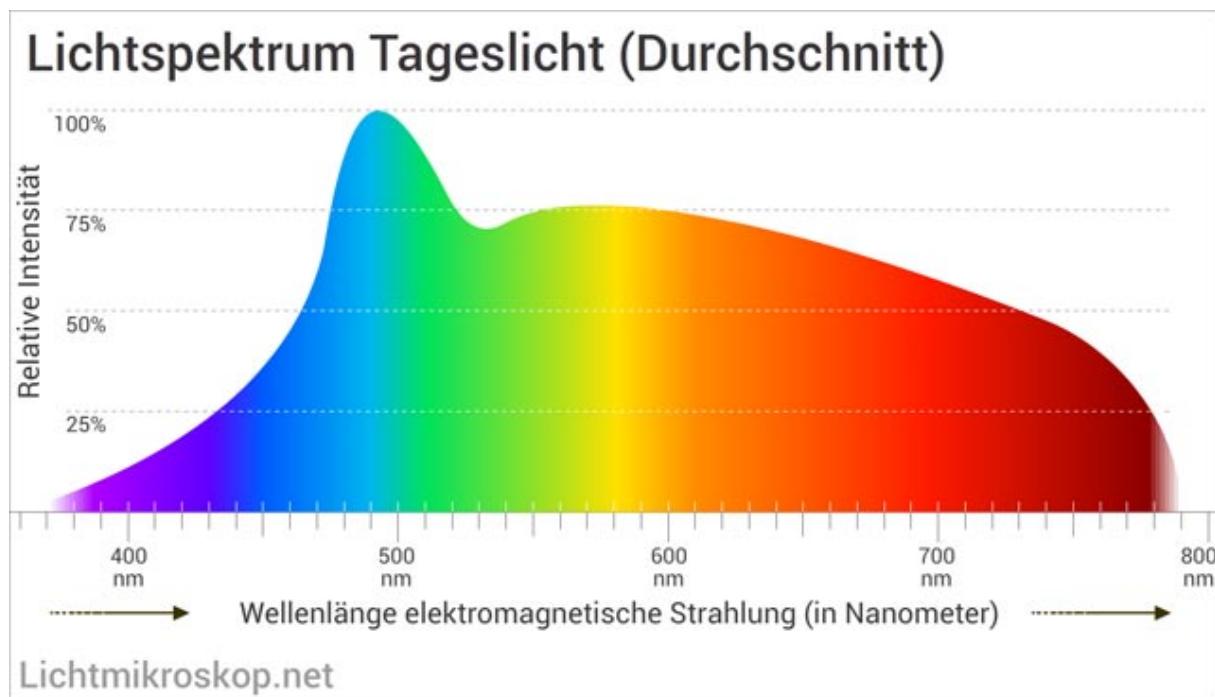


Abbildung 3.5.1: Lichtspektrum des Sonnenlichts⁵²

3.5.2 Ansatz

Darum bietet es sich an, die Leuchten von Sanlight, welche das Spektrum des sichtbaren Lichtes von ca. 400nm bis etwa 700nm abdecken, im Zusammenspiel mit einer UV-Licht Quelle zu verwenden. Damit sollten alle nötigen Wellenlängen abgedeckt sein.

3.5.3 Lampen

Als Representant für das sichtbare Licht wurde die Q1W Dim von SANlight, die uns zur Verfügung gestellt wurde, verwendet. Mit 50 Watt ist sie ein Kompromiss zwischen Realität und dem was praktisch möglich ist. Das Spektrum zeigt dass ausschließlich sichtbares Licht vorhanden ist.

Light Spektrum / Licht Spektrum

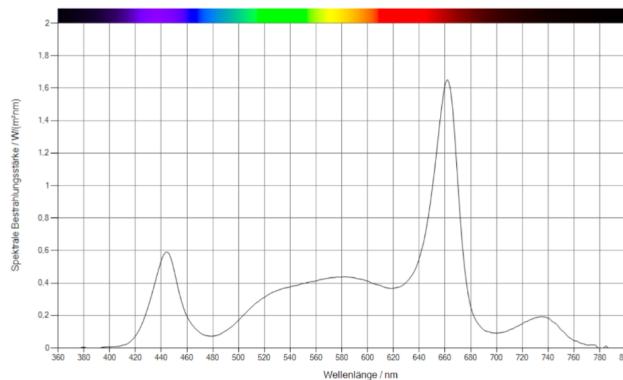


Abbildung 3.5.2: Sanlight Spektrum

Als Representant für UV-Licht gibt es mehrere Ansätze. Zum einen können UV-LEDs benutzt werden. Diese haben den Vorteil, dass sie dimmbar sind. Jedoch sind diese LEDs ziemlich teuer und benötigen außerdem bestimmte Treiber die einen entsprechenden Preis mit sich bringen. Ein anderer Ansatz waren UV-Halogen Lampen. Diese sind zwar zum Teil dimmbar, wird aber nicht empfohlen, da diese durch den Dimmvorgang schneller kaputt gehen.⁵³. Somit bleibt die Frage der UV Lampe offen.

3.5.4 Steuerung

Quellcode⁵⁴

Die Q1W DIM ist, wie der Name schon verrät, über einen Drehschalter in Zwanzigsschritten von 0% bis 100% dimmbar. Jedoch sollte die Leuchtintensität von einem Raspberry Pico automatisiert geändert werden können. Wenn man den Drehschalter der Leuchte genauer betrachtet, kann man die Widerstandswerte erkennen. Der Drehschalter schaltet je nach Bedarf einen anderen Widerstand in Serie

⁵³ Halogenlampen. watt24 GmbH. URL: <https://www.watt24.com/leuchtmittel/halogenlampen/> (besucht am 19.03.2024).

⁵⁴ Kikomtk. Tee6/klimakammer-firmware-SUN. original-date: 2024-02-14T09:03:58Z. 14. Feb. 2024. URL: <https://github.com/Tee6/klimakammer-firmware-SUN> (besucht am 04.04.2024).

um eine andere Intensität zu erzielen. Es bietet sich an, mittels eines Multiplexers den Drehschalter zu ersetzen. Der Raspberry kann mit 3 Adressleitungen einen von 8 Kanälen auswählen, somit den beschalteten Widerstand auswählen und daraus die Lichtintensität steuern.

Dieses Verfahren wurde außerdem mit einer Widerstandsbox getestet wie in Abbildung 3.5.3 zu sehen ist.

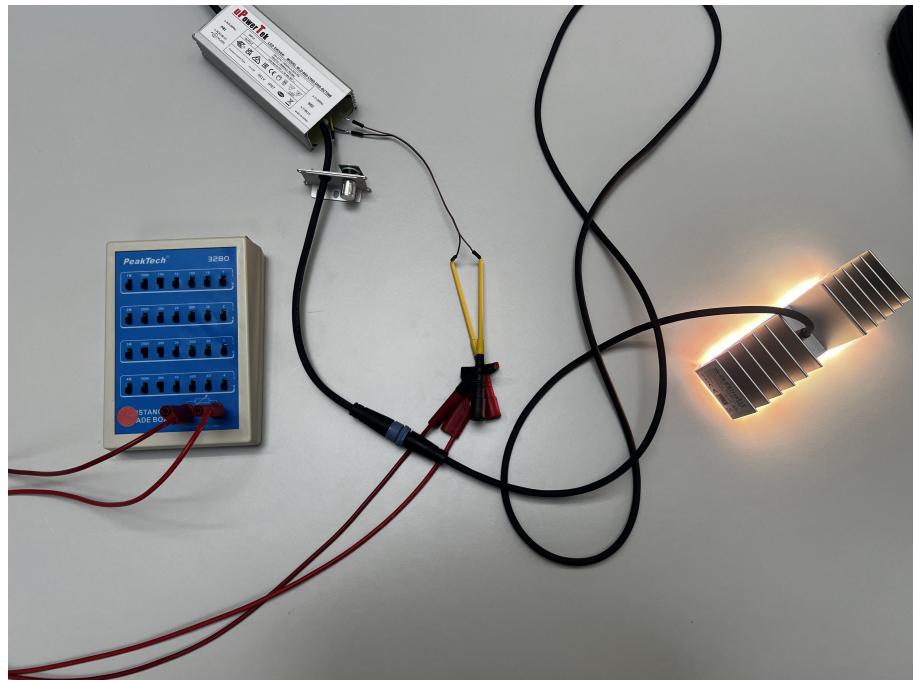


Abbildung 3.5.3: Testung des Steuerungsprinzips

Die Widerstandswerte für die einzelnen Intensitätstufen lauten:

- 0% - 0Ω
- 20% - $19.1\text{k}\Omega$
- 40% - $1.14\text{k}\Omega$
- 60% - $2.3\text{k}\Omega$
- 80% - $5.3\text{k}\Omega$
- 100% - $380\text{k}\Omega$

Als Multiplexer wurde zuerst der 74HC151N IC [Abbildung 3.5.4] Verwendet. Er hat 8 Eingänge die auf einen Ausgang beschaltet werden können.

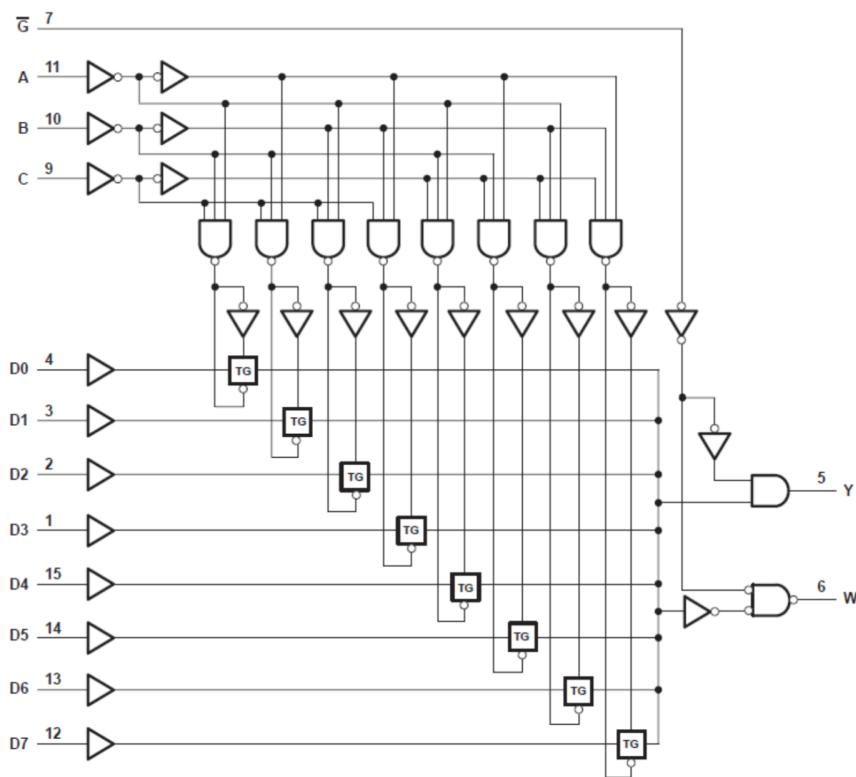


Abbildung 3.5.4: Interne Schaltung des 74HC151N Multiplexers

Die 2 extra Widerstände in Abbildung 3.5.5 sind dazu da, im Falle eines Fehlers des MUX einen Ausweichkanal verwenden zu können. Später stellte sich heraus dass dieser MUX die falsche Entscheidung war, da ein analog-Multiplexer benötigt wird. Diese Fehlentscheidung ist jedoch nicht schwerwiegend, da ein fast äquivalenter Analog Multiplexer CD4051B verfügbar ist. Bei diesem muss ausschließlich der Ausgangspin umgeleitet werden. Dieser liegt nämlich auf dem ehemaligen D1 Eingang. Der Widerstand am D1 Eingang kann jedoch auf einen Backup Widerstand verlegt werden, und mit einem Jumper Draht die korrekte Verdrahtung hergestellt werden. Dieser Analoge MUX funktioniert im Grunde gleichermaßen wie der digitale, nur dass die Verschaltung nicht durch Gatter sondern mit einem Schalter wie in Abbildung 3.5.6

Der Ansatz mit dem analogen Multiplexer funktionierte jedoch nicht vollkommen. Deshalb wurde eine ganz neue Leiterplatte bestellt, wobei die Schaltung der Widerstände mit einfachen Mosfets realisiert wird. Dieser Ansatz ist zwar primitiv, funktioniert aber trotzdem besser als die Ansätze zuvor. Die Schaltung ist in 3.5.7 zu erkennen.

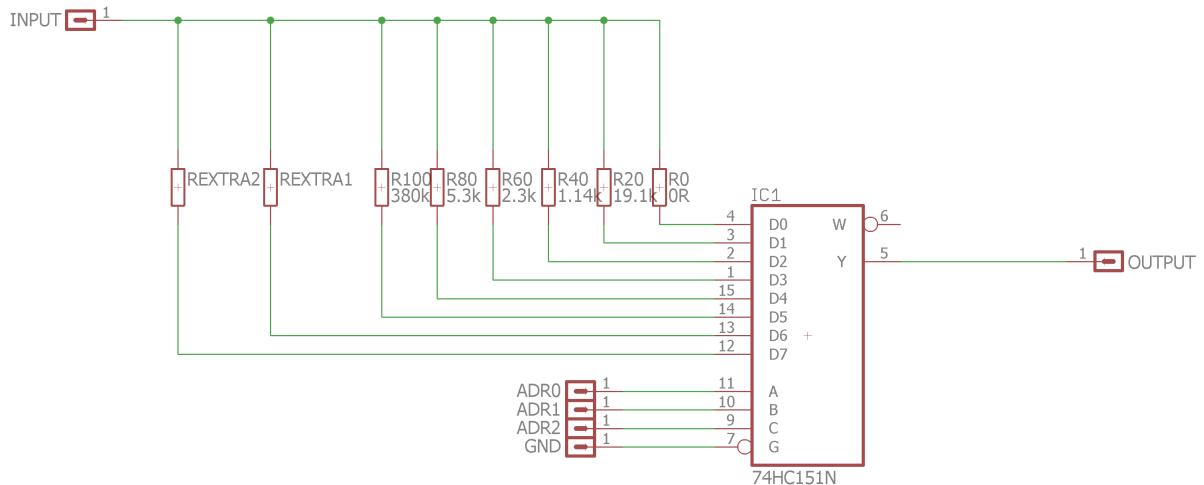


Abbildung 3.5.5: Schaltplan für die Steuerung der Q1W DIM

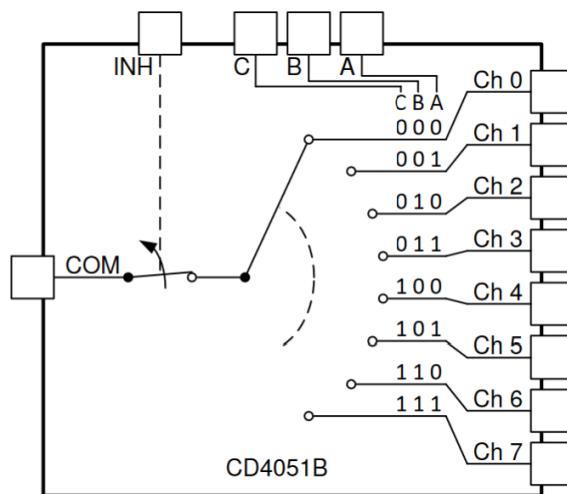


Abbildung 3.5.6: Interne Funktionsweise CD4051B

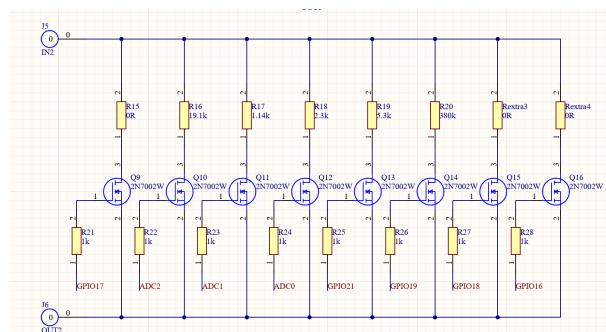


Abbildung 3.5.7: Neue Sonnensteuerung mittels Mosfets

Als Ausblick für die Sonnensteuerung könnte man einen eigenen Treiber für die Lampen realisieren, um diese genauer als zwanziger Schritte steuern zu können.

Außerdem könnte man die Problematik mit dem UV-Licht lösen, indem man die UV-LEDs kauft und vielleicht einen Bereich begrenzt der für UV reserviert ist, damit nur dieser mit UV bestrahlt wird. Dies würde die Leistungsproblematik beseitigen.

3.6 Kameraerfassung

Um das Geschehen in der Kammer überwachen zu können, ohne die Tür zu öffnen und das innere Klima zu beeinflussen, bietet sich ein Livestream einer Kamera im Inneren an, der auf der Website abgerufen wird.

Als Kamera wird die ELP USB Kamera 720P verwendet [Abbildung 3.6.1]. Dies ist eine Wasserdichte Outdoor Kamera welche über einen Nachsichtmodus verfügt der es ermöglicht, auch bei kompletter Dunkelheit einen Einblick in die Kammer zu bekommen. Dieser Nachtmodus wird aktiviert, sobald der Lichtsensor der Kamera nicht genügend Licht misst.



Abbildung 3.6.1: Überwachungskamera

Zuerst war es so implementiert, dass die Kamera am Bildschirm an der Kammer angeschlossen wird. Jedoch vernichtet dies den ganzen Zweck einer Website, wenn nicht auf jedem Gerät das gleiche dargestellt wird. Somit sollte die Kamera am Raspberry pi angeschlossen sein, welcher den Stream auf eine IP sendet die auf der Webseite aufgerufen werden kann.

Dafür wird ein Open Source Programm namens „motion“ verwendet. Mit diesem Programm kann der Kamerastream über eine IP aufrufbar gemacht werden.

Motion wurde zuerst auf dem RaspberryPi installiert. Mit dem Befehl „sudo apt-get

install motion " wird Motion installiert. Anschließend muss in der Konfigurationsdatei unter „/etc/motion/motion.conf“ einige Einstellungen vorgenommen werden. Zum einen muss der daemon Modus aktiviert werden, welcher es erlaubt, dass das Programm im Hintergrund läuft. Zudem muss deaktiviert werden, dass der Stream ausschließlich auf dem Localhost abgefragt werden kann. Die Einstellung lautet „stream_localhost“ welche auf „on“ gesetzt werden muss. Dieselbe einstellung gilt für „webcontrol_localhost“ Nun muss im File „/etc/default/motion“ daemon zusätzlich aktiviert werden, damit das Programm wirklich im Hintergrund startet. Nun kann mit „sudo service motion start“ der Stream gestartet werden. Auf der Website kann nun ein normales -HTML Element benutzt werden bei dem die src die IP des Raspberrys unter dem Port 8081 beträgt um den Stream auf der Website anzeigen zu lassen.

Das Programm wird automatisch beim Start des Raspberrys gestartet. Der Stream kann nun über die IP des Raspberrys mit dem Port 8081 aufgerufen werden.

3.7 Sensoren

Um die eingestellten Parameter auch messen zu können, und um somit genau über den Zustand im Inneren der Kammer Bescheid zu wissen, müssen diese gemessen werden. Hierfür wurde eine Platine entwickelt, an welche bis zu 8 I2C-Sensoren angeschlossen werden können. Diese werden dann von einem Raspberry Pico ausgelesen, und auf Anfrage an den Raspberry Pi zurückgesendet. Die einzige Aufgabe der Platine ist das Verteilen der I2C-Signale und das runter regeln der Spannung auf 3.3V, weshalb hier auf ein Blockschaltbild verzichtet wird. Die wichtigsten Parameter konnten mit nur zwei verschiedenen Sensoren ausgelesen werden, auf welche im folgenden eingegangen wird.

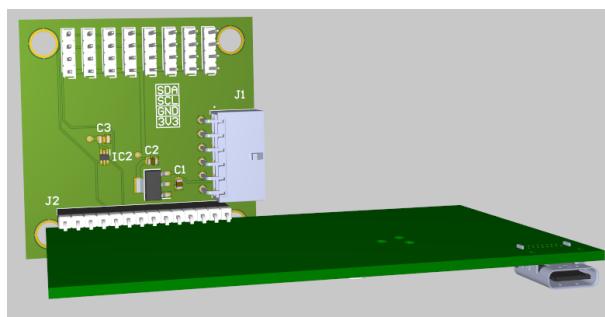


Abbildung 3.7.1: Sensorboard

3.7.1 BME280

Um Luftdruck, Luftfeuchtigkeit und Temperatur innerhalb der Kammer messen zu können, wurde der BME280⁵⁵ von Bosch verwendet. Dieser misst:

- **Temperatur:** Zwischen -0°C bis +65°C mit einer Genauigkeit von 0.5°C.
- **Luftfeuchtigkeit:** Zwischen 0%RH und 100%RH mit einer absoluten Genauigkeit von 3%RH.
- **Luftdruck:** Von 300hPa bis 1100hPa mit einer Genauigkeit von 1hPa.

Wie im Kapitel 5.3 später noch gezeigt wird, kann der Sensor in verschiedenen Modi und mit verschiedenen Konfigurationen für Filter und Oversampling verwendet werden. Diese sind maßgeblich für die hohe Genauigkeit der Messung. Um nicht extra eine Platine für den Sensoren entwickeln zu müssen, wurde ein fertiges Modul von AZDelivery⁵⁶ gekauft. Außerdem wurde noch ein Gehäuse konstruiert und 3D-gedruckt, um den Sensor vor Spritzwasser zu schützen.

3.7.2 LTR-390

Der LTR-390⁵⁷ von LITEON-Optoelectronics ist ein kombinierter ALS (Ambient Light Sensor) und UV-Sensor. Er misst diese Lichtstärken mit einer Auflösung von bis zu 20 Bit. Wenn diese Werte von Sensoren empfangen wurden, können sie mit einer vom Hersteller gegebenen Formel (Formel) in Lux bzw. UV-Intensität umgewandelt werden.

$$Lux_{calc} = \frac{0.6 * ALS_{DATA}}{GAIN * INT} * W_{FAC}$$

$$UVI_{calc} = \frac{UV\ Sensor\ Count}{UV\ Sensitivity} * W_{FAC}$$

Auch hier wurde ein Gehäuse gedruckt. Außerdem wurde der Sensor dann neu kalibriert, da das durchsichtige Plastik, welches den Sensor vor Spritzwasser schützen soll, die Lichtintensität natürlich abschwächt. Dafür wird der Faktor W_{FAC} in den

⁵⁵ Humidity Sensor BME280. Bosch Sensortec. URL: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/> (besucht am 04.04.2024).

⁵⁶ GY-BME280 barometric sensor for temperature, air humidity and air pressure. AZ-Delivery. URL: <https://www.az-delivery.de/en/products/gy-bme280> (besucht am 04.04.2024).

⁵⁷ LTR-390UV_Final_DS_V1.1.pdf. URL: https://optoelectronics.liteon.com/upload/download/DS86-2015-0004/LTR-390UV_Final_%20DS_V1%201.pdf (besucht am 04.04.2024).

Formeln benutzt.

3.8 Windsteuerung

Die Simulation des Windes ist relativ simpel aufgebaut. Dabei werden einfach zwei Lüfter in der Kammer platziert. Der eine bewegt die Luft nach außen, und der andere saugt frische Luft von außen an. Da in der Kammer aber eine sehr hohe Luftfeuchtigkeit auftreten kann, und auch Spritzwasser die Lüfter berühren kann, müssen diese wasserfest sein.

Dafür wurden von uns zwei NF-F12 industrialPPC (Abbildung 3.8.1)⁵⁸ von Noctua verwendet. Dieser funktioniert mit 12V und hat eine Maximaldrehzahl von 2000 RPM. Der ausschlaggebende Faktor für diese Lüfter war aber ihre IP67 Zertifizierung, welche sie vor Staub und Nässe schützt. Außerdem ist dieser Lüfter ein 4-Pin Lüfter, d.h. dass über ein separates PWM-Signal die Geschwindigkeit geregelt werden kann, und über ein anderes Signal (Tacho) die Drehzahl gemessen werden kann.



Abbildung 3.8.1: NF F12 industrial PPC von Noctua

⁵⁸ Noctua. *NF-F12 industrialPPC-2000 IP67 PWM*. URL: <https://noctua.at/de/nf-f12-industrialppc-2000-ip67-pwm> (besucht am 05.04.2024).

Prinzipiell ist jeder Lüfter PWM-fähig, jedoch muss dabei die Versorgungsspannung PWM-moduliert werden, was mindestens einen MOSFET benötigt. Die Spannungslevel und Frequenzen des PWM- und Tachosignals sind in einem von Noctua erstellten Whitepaper⁵⁹ festgelegt. Die Frequenz des PWM-Signals muss zwischen 21kHz und 28kHz liegen, und das Low-Signal darf höchstens 0.8V betragen. Das Tacho-Signal ist als Open-Collector-Output aufgebaut. Dabei ist die Spannung auf welche der benötigte Pullupwiderstand gezogen wird frei wählbar.

Zur Steuerung der beiden Lüfter wurde eigentlich ein IC herausgesucht und dafür eine Platine designed. Zum Zeitpunkt der Bestellung war der IC dann aber nicht mehr lieferbar, weshalb die Steuerung der Lüfter dann direkt über den Raspberry Pico erfolgte.

⁵⁹ Noctua. *Noctua_PWM_specifications_white_paper.pdf*. URL: https://noctua.at/pub/media/wysiwyg/Noctua_PWM_specifications_white_paper.pdf (besucht am 05.04.2024).

4 Software

4.1 Website

Quellcode⁶⁰

Zur Steuerung und Analyse der Simulationsergebnisse, bedarf es einer Website mit einfachem und ansprechendem Design

4.1.1 Anforderungen

Generelle Anforderungen bestehen zum einen in der einfachen Erweiterbarkeit. Das gewährleistet einfaches Austauschen von Komponenten oder ganzen Regelkreisen ohne großen Softwareaufwand. Langfristig bietet dadurch die Website niedrige Instandhaltungskosten. Wie ein neuer Parameter hinzugefügt werden kann, wird in 4.1.5 gezeigt. Zum anderen sind ein ansprechendes und modernes UI eine grundlegende Anforderung wodurch die Website bedienerfreundlicher und intuitiver Gestaltet wird.

Steuerung Jegliche installierte Regler sollen per Webinterface gesteuert werden können. Es sollen Routinen erstellt werden können - um die Automatisierung der Kammer zu ermöglichen - aber auch die manuelle Steuerung durch direkte Eingabe von Parametern ermöglicht werden. Zusätzlich soll über eine WetterAPI realistische Daten von Orten der ganzen Welt in das System eingespielt werden können.

Analyse Jedes Geschehen im inneren der Kammer soll dokumentiert und grafisch dargestellt werden, um die Verläufe der Parameter besser einschätzen zu können, und frühzeitig Fehler zu erkennen.

⁶⁰ Kikomitk. Tee6/Diplomarbeit_Klimakammer. original-date: 2023-10-13T11:24:04Z. 25. Jan. 2024.
URL: https://github.com/Tee6/Diplomarbeit_Klimakammer (besucht am 04.04.2024).

4.1.2 Framework und Libraries

Zur Erstellung der Website werden mehrere Libraries benutzt. Zusätzlich wird ein Javascript Framework verwendet um mehr Funktionalität und modernes Design ausnutzen zu können. Das Framework Vue mit Typescript in der Version Vue3 hat viele Vorteile für die Diplomarbeit. Die einfache Integration und flexible Verwendung von Komponenten ermöglichen eine zeitsparende Entwicklung, ohne ein neues Framework lernen zu müssen. Vue.js' reaktive Variablen sind besonders nützlich für die dynamische Aktualisierung von Messergebnissen und Grafiken auf der Website. Zudem ist die umfangreiche Dokumentation und der Fakt dass uns das Framework schon bekannt ist und somit nicht neu erlernt werden muss von essentiellem Vorteil.

Vue Vue ist ein Javascript Framework welches 2014 von Evan You entwickelt wurde. In Vue können sogenannte „Components“ erstellt werden, welche den Source-Code übersichtlicher und wiederverwendbarer gestalten. In dieser Diplomarbeit wird die aktuellste Version Vue3 zusammen mit Typescript und dem „`<script setup>`“ Syntax verwendet.⁶¹ Eine Vue Datei ist wie in Abbildung 4.1.1 aufgebaut.

```
<template>
</template>

<script setup>
</script>

<style>
</style>
```

Listing 4.1.1: Vue Struktur

Im Template kann normaler HTML Syntax verwendet werden um die Website aufzubauen, im Script kann normaler Javascript bzw. in diesem Falle Typescript verwendet werden und im Style kann normaler CSS Syntax verwendet werden. Was Vue nun besonders macht, ist dass diese drei Bereiche miteinander interagieren, und Daten austauschen können. Durch den „Mustache Syntax“ können Variablen aus dem Script Teil reaktiv (= jederzeit aktuell) auf der Website abgebildet werden, wie

⁶¹ Evan You. *Vue.js - The Progressive JavaScript Framework | Vue.js*. URL: <https://vuejs.org/> (besucht am 23.12.2023).

in Abbildung 4.1.2 veranschaulicht:

```
<template>
  <h1>{{ counter }}</h1>
  <button @click="counter++">Click me</button>
</template>

<script setup>
import { ref } from 'vue'

const counter = ref(0)
</script>
```

Listing 4.1.2: Beispiel eines Reaktiven counters in Vue

Hier wird eine counter Variable bei jedem Klick des Buttons inkrementiert. Durch das Importieren der ref Funktion wird auf der Website jederzeit der neue Wert von Counter ausgegeben, was bei nativem HTML nicht automatisch funktionieren würde und wesentlich mehr Code benötigen würde.

Libraries

- vue-router

Um die Website navigieren zu können, wird der „vue-router“ verwendet. Dieser arbeitet im Browser, also auf Seite des Users, und bietet somit eine schnelle Navigationsmöglichkeit zwischen verschiedenen Ansichten.⁶²

- Pinia

Pinia ist eine „Store“ und „State-Managing“ Library. Das bedeutet, dass Variablen, Funktionen und andere Definitionen in einer externen Datei definiert sind, und mit einer Zeile in jeglichen Dateien benutzt werden können. Somit können reaktive Variablen über mehrere Dateien dynamisch verwendet werden und eine bessere Übersicht verschafft. Früher war VueX der Standard für Vue.js Stores, jetzt wird Pinia aber von Vue empfohlen.⁶³

- Chart.js

Chart.js ist eine Open-Source Javascript Bibliothek, mit der mehrere Diagramme erstellt werden können. Durch den einfachen Syntax bietet sich diese Bibliothek an für die Visualisierung sämtlicher Daten. In dieser Diplomarbeit wird hauptsächlich das Liniendiagramm benutzt.⁶⁴

TypeScript TypeScript ist ein „strict superset“ von Javascript welches von Microsoft entwickelt wurde. Jeder Javascript Code kann in TypeScript verwendet werden und somit auch alle Javascript Bibliotheken und Frameworks. Der entscheidende Unterschied zwischen Javascript und TypeScript besteht darin, dass TypeScript im Gegensatz zu Javascript strikt Typisiert ist, was Fehler in der Entwicklung vermeidet und Code besser lesbar gestaltet.

⁶² Evan You und Eduardo San Morote Morote. *Vue Router | The official Router for Vue.js*. URL: <https://router.vuejs.org> (besucht am 23.12.2023).

⁶³ Eduardo San Morote Morote. *Everything Beyond State Management in Stores with Pinia by Eduardo San Martin Morote - GitNation*. URL: <https://portal.gitnation.org/contents/everything-beyond-state-management-in-stores-with-pinia> (besucht am 23.12.2023).

⁶⁴ *Chart.js | Chart.js*. URL: <https://www.chartjs.org/docs/latest/> (besucht am 23.12.2023).

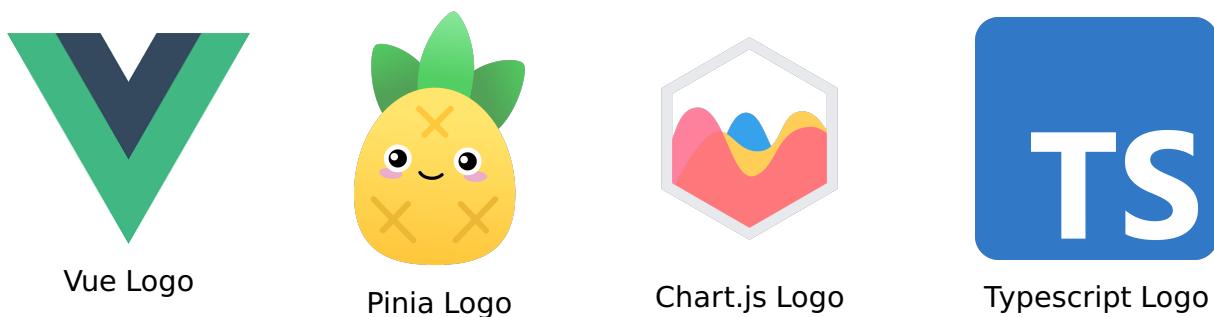


Abbildung 4.1.1: Library und Framework Logos (vue-router hat kein Logo)

4.1.3 Features

In den folgenden Abbildungen werden Beispielwerte benutzt die zum Teil unrealistisch sind. Zudem sind Module wie zum Beispiel die Temperaturregelung auf der Website zwar integriert, jedoch noch nicht in der Diplomarbeit eingebaut. Diese Elemente wurden als Platzhalter und für die zukünftige Erweiterung trotzdem frühzeitig hinzugefügt.

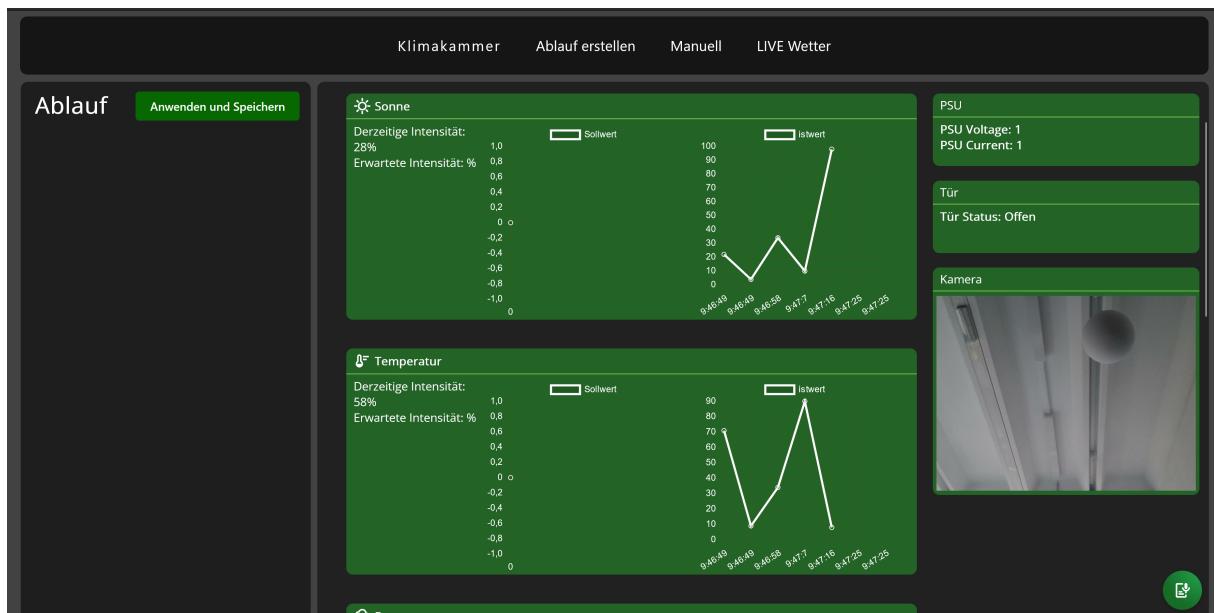


Abbildung 4.1.2: „Startmenu der Website“

Übersicht In der Abbildung 4.1.2 ist die Start- und Hauptansicht der Website zu sehen. Wenn keine Aktion definiert ist, wird ausschließlich die derzeitigen Sensordaten angezeigt. In dieser Übersicht kann jedes Geschehen der Kammer überwacht werden.

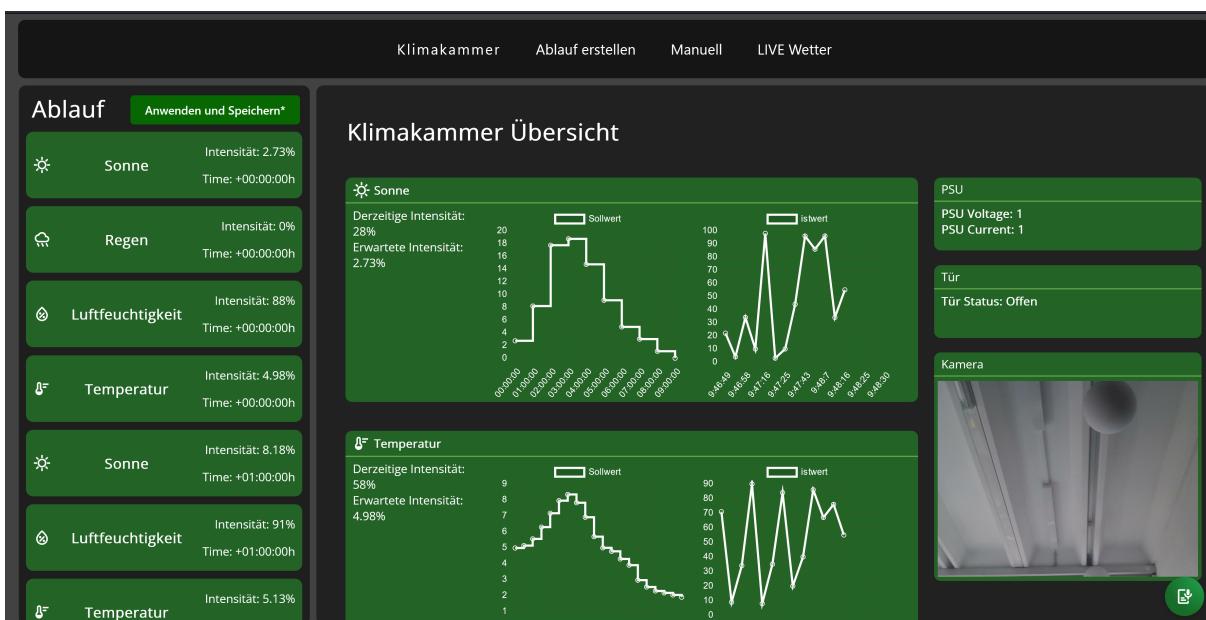


Abbildung 4.1.3: „Startmenü der Website befüllt“

Wenn Aktionen entweder Automatisch, manuell oder per API geladen wurden, wird eine Grafik über den Verlauf der Parameter angezeigt. Je genauer die Parameter definiert sind, desto „analoger“ wird die Kurve aussehen.

Routine erstellen Um das Geschehen der Kammer automatisieren, und beliebige Szenarien erstellen zu können, muss die Möglichkeit der Routinenerstellung gegeben sein. Die Bedienoberfläche in Abbildung 4.1.4 besteht aus einzelnen Knöpfen die jeweils einen Parameter steuern.

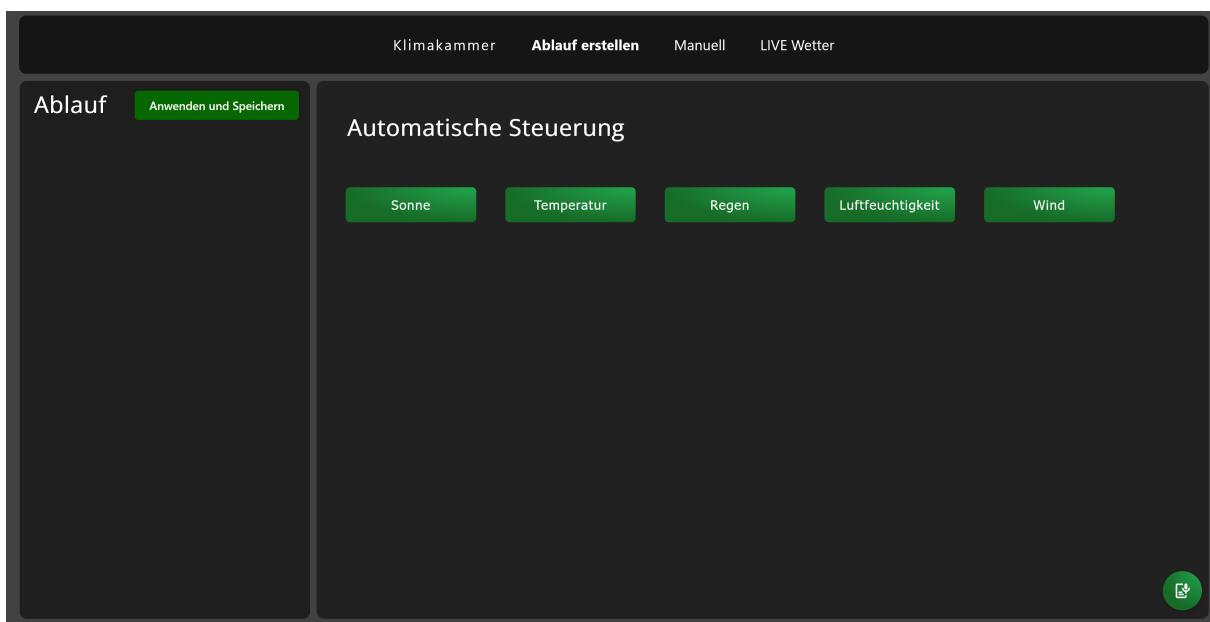


Abbildung 4.1.4: „Bedienoberfläche um Ablauf zu erstellen“

Wenn ein Parameter ausgewählt wurde, erscheint ein PopUp, wie in Abbildung 4.1.5 zu sehen ist, indem die Intensität und Startzeit eingestellt werden kann.



Abbildung 4.1.5: „PopUp für Sonnenparameter“

Manuelle Steuerung Um einzelne Änderungen schnell vorzunehmen, und um den Testprozess einfacher zu gestalten, soll die Funktion der Manuellen Steuerung

eingebaut werden.

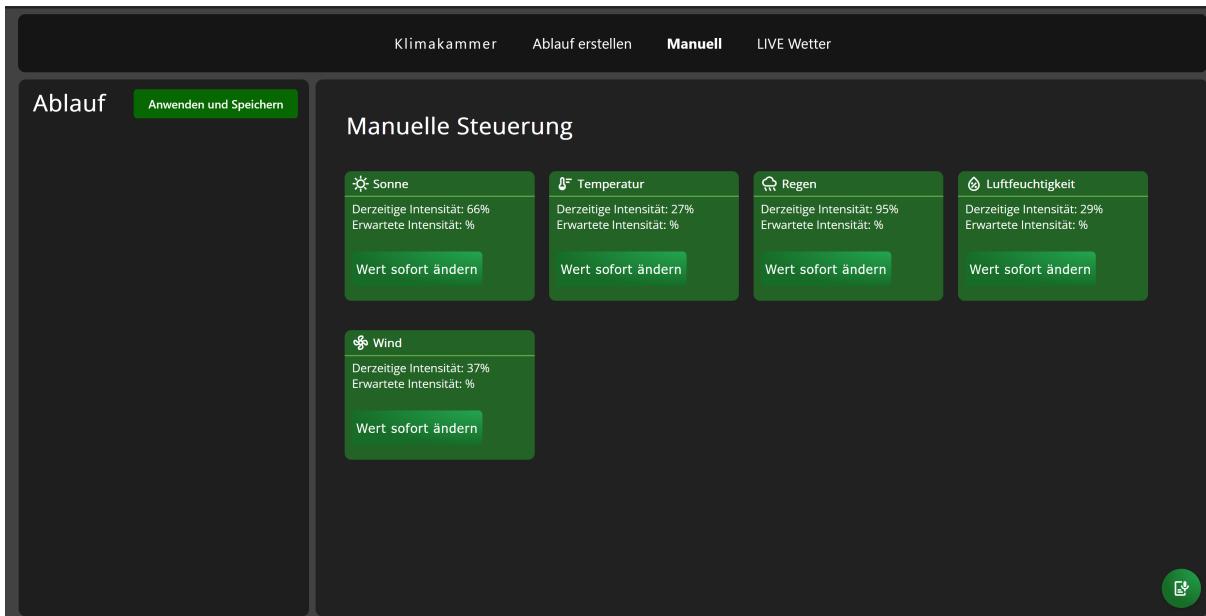


Abbildung 4.1.6: „Manuelle Steuerung Übersicht“

Live Wetter Die letzte Funktion der Website besteht darin, Wetter aus der ganzen Welt nachzubilden. Über die API von „OpenWeatherMap“ werden alle relevanten Daten abgerufen und graphisch dargestellt.

Die API von Openweathermap ist kostenlos für jeden zugänglich. Es muss lediglich ein Account auf der Website⁶⁵ erstellt werden, und ein API Key angefragt werden. Nun kann durch eine einmalige Initialisierung beim Start, jegliche Wetterdaten (je nach Abonnement) abgefragt werden.

Um nun die Wetter Daten eines Ortes zu bekommen, benötigt man zuerst die genauen Koordinaten. Praktischerweise bietet Openweathermap.org hierfür ebenfalls einen API Service. Durch einen HTTP-GET Call

```
http://api.openweathermap.org/geo/1.0/direct?q=[Ortsname]
&limit=5&appid=[API-Key]
```

⁶⁵ openweathermapwebsite. URL: <https://openweathermap.org/> (besucht am 12.03.2024).

bekommt man folgende Antwort:

```
[  
 {  
   "name": "Rankweil",  
   "local_names": {  
     "de": "Rankweil",  
     "hu": "Rankweil"  
   },  
   "lat": 47.2703808,  
   "lon": 9.6426923,  
   "country": "AT",  
   "state": "Vorarlberg"  
 }]  
 ]
```

Listing 4.1.3: Antwort von Openweathermap um die Koordinaten eines Ortes zu bestimmen

Aus dieser Antwort können die Parameter „lat“ und „lon“ abgelesen werden. Anschließend kann durch einen GET Call auf

```
https://api.openweathermap.org/data/2.5/onecall?lat=\[lat\]&lon=\[lon\]&appid=\[API-key\]&units=metric
```

die Wetterdaten des Ortes abgefragt werden. Die Angabe „units=metric“ ist optional, jedoch verwenden wir in Europa das Metrische System wodurch Einheiten wie Celsius angenehmer und gewohnter sind.

Die Gratis Version der API bietet folgende Parameter:

- lat:** Breitengrad der abgefragten geografischen Position, in diesem Fall 47.2704.
- lon:** Längengrad der abgefragten geografischen Position, in diesem Fall 9.6427.
- timezone:** Zeitzone, in der sich die geografische Position befindet, hier "Europe/Vienna".
- timezone_offset:** Zeitunterschied in Sekunden zur UTC-Zeit, hier 3600 Sekunden (1 Stunde).
- current:** Objekt der aktuellen Wetterdaten.
- dt:** Unix-Zeitstempel für die aktuelle Datenabfrage.
 - sunrise:** Unix-Zeitstempel für den Sonnenaufgang.
 - sunset:** Unix-Zeitstempel für den Sonnenuntergang.
 - temp:** Aktuelle Temperatur.
 - feels_like:** Gefühlte Temperatur.
 - pressure:** Luftdruck in hPa.
 - humidity:** Luftfeuchtigkeit in Prozent.
 - dew_point:** Taupunkt in °C.
 - uvi:** UV-Index.
 - clouds:** Bewölkung in Prozent.
 - visibility:** Sichtweite in Metern.
 - wind_speed:** Windgeschwindigkeit in m/s.
 - wind_deg:** Windrichtung in Grad.
 - wind_gust:** Böengeschwindigkeit in m/s.
 - weather:** Wetterzustand.
- id:** Wetterbedingungs-ID.
 - main:** Hauptwetterkategorie um schnell erkennbar zu machen wie das Wetter gerade ist.
 - description:** Beschreibung des Wetters.
 - icon:** Wetter-Icon-Code, hier "04n" mit diesem Code kann auf der Openweathermap.org Website ein Icon zu jedem Wetter abgefragt werden, was auf der Website auch benutzt wird, wie in Abbildung 4.1.7 zu sehen ist.

Die OpenWeatherMap-API bietet außerdem den Vorteil, dass bei diesem API Call 40 weitere Objekte ähnlicher Art mitgeliefert werden, die Vorhersagen für die nächsten 40 Stunden enthalten.

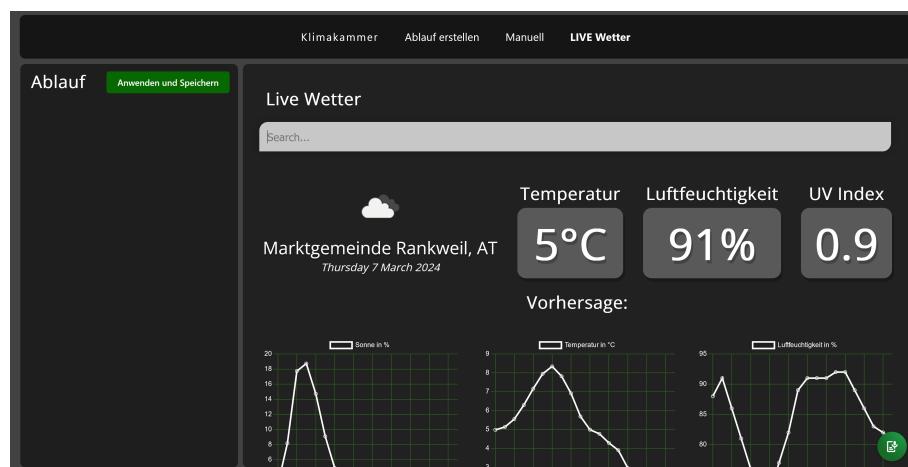


Abbildung 4.1.7: „Live Wetter Übersicht“

Unter den Graphen der Vorhersage befindet sich ein „Apply Button“ mit dem das Wetter in die Kammer eingespielt werden kann.

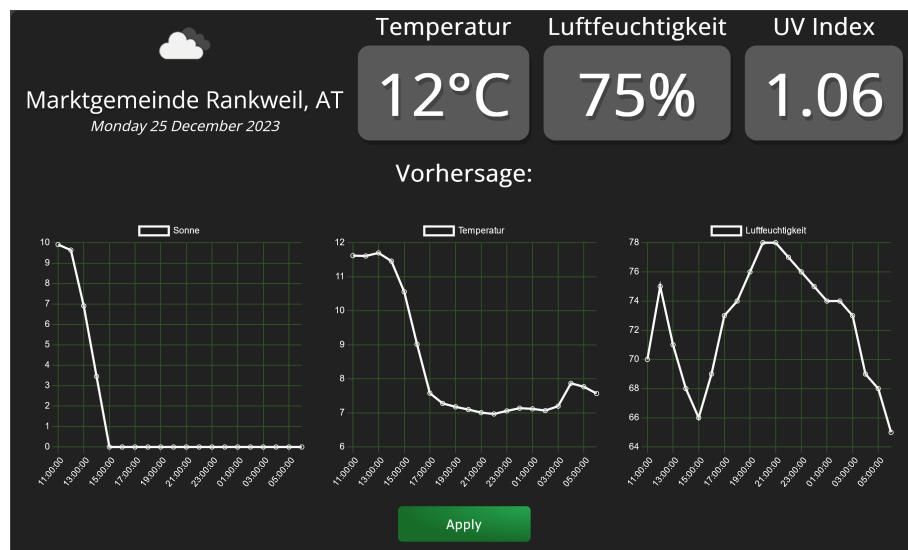


Abbildung 4.1.8: „Wetter Vorhersage inkl. Graphen“

Mit einem Klick auf den Apply Button werden die einzelnen Werte eingegeben, was in einer Übersicht wie in Abbildung 4.1.9 resultiert

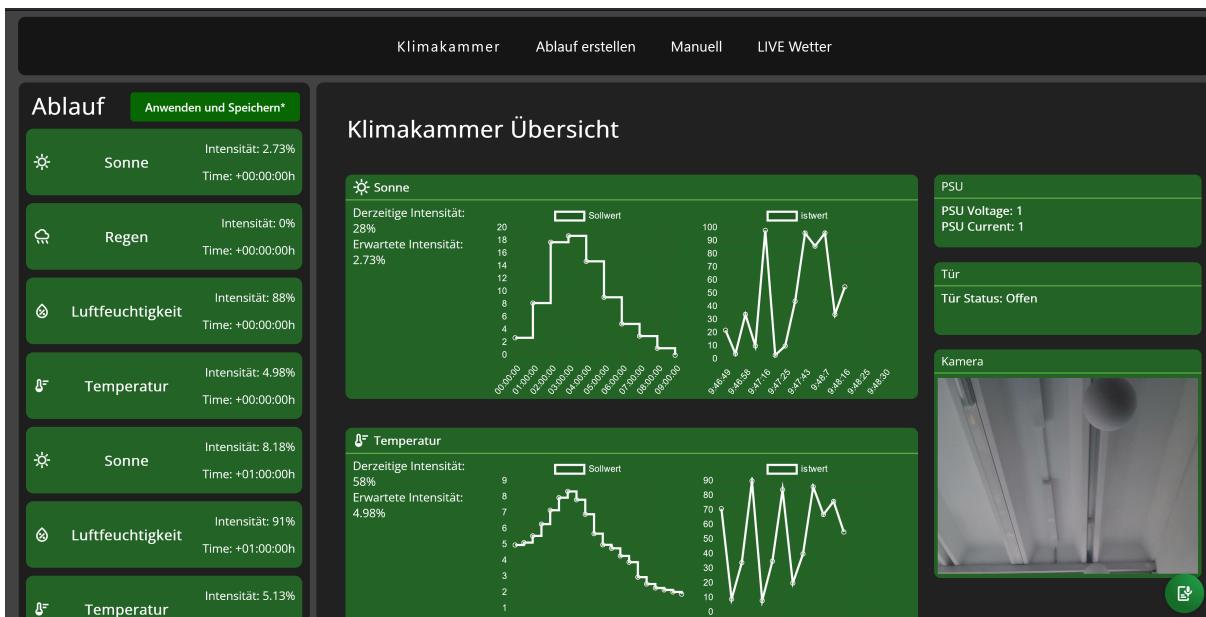


Abbildung 4.1.9: „Übersicht mit gefülltem Ablauf“

Die Graphen sind deshalb so eckig, da unsere Kammer die Parameter zwar updatet, zwischenzeitlich aber keine Interpolation stattfindet, um die Graphen kurviger zu gestalten. Auf der linken Seite ist nun der Ablauf, der aus dem echten Wetter extrahiert wurde, abgebildet. Durch einen Linksklick kann die Aktion bearbeitet oder gelöscht werden.

Einstellungen Außerdem wurde ein letzter Tab hinzugefügt, um dem Nutzer so viel Einstellungsmöglichkeiten wie möglich zu verschaffen. Es können 3 Parameter verändert werden. Zum einen kann die Samplesize verändert werden. Diese ist dazu da zu bestimmen, wie viele Samples des Live Wetters eingespielt werden sollen. Maximal können 40 Samples benutzt werden, weil die API von Openweathermap nicht mehr samples hergibt. Diese 40 Samples beschreiben das Wetter in den nächsten 40 Stunden. Zum anderen kann die Refresh time definiert werden. Dieser Parameter bestimmt, in welchem Abstand neue Ist werte vom Backend abgefragt werden sollen. Zu guter Letzt kann noch eine Einstellung namens „Swift“ aktiviert oder deaktiviert werden. Diese Einstellung bewirkt, dass die Graphen in der Hauptübersicht sich weiter bewegen, wenn sie zu einer gewissen Anzahl an Samples kommen. Dies kann nützlich sein, wenn die Kammer lange läuft, aber nur die neuesten Werte interessant sind.

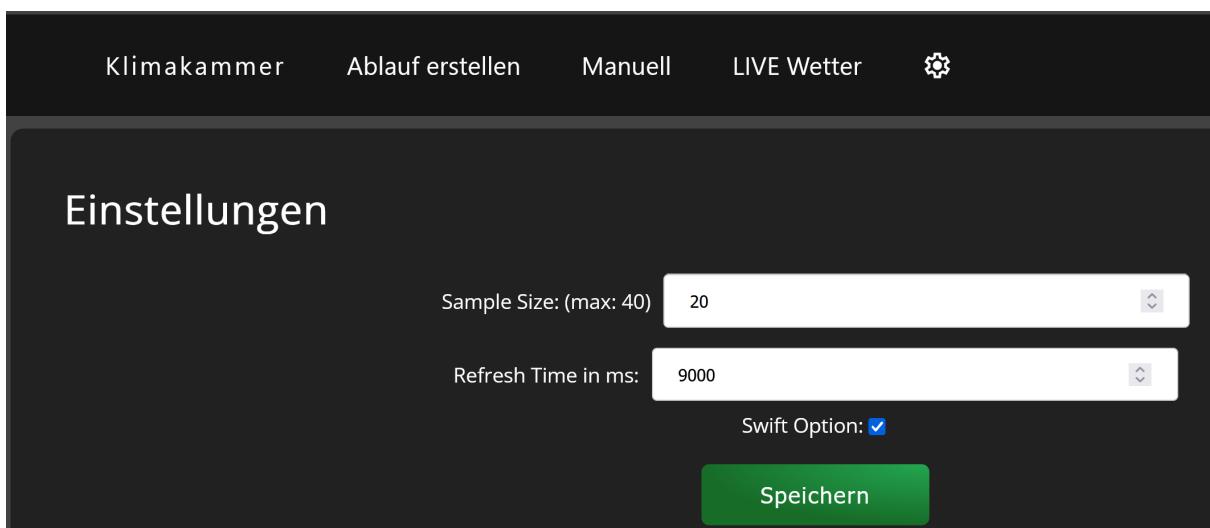


Abbildung 4.1.10: Einstellungs Übersicht

4.1.4 Funktionsweise

Die grundsätzliche Funktion der Website basiert auf den Interfaces, der ActionList und den reaktiven Eigenschaften die durch die Library Pinia und das Framework Vue gegeben.

Erstellen einer Aktion Wenn bei einem PopUp eine Aktion erstellt wird, wird in der ActionList ein neues Objekt erstellt. Dieses Objekt hat eine eindeutige id, einen Namen welcher dem Parameter entspricht (z.B. Sonne), einen Sollwert, und Angaben für die Zeit. Diese Typisierung hilft dabei, Runtime Fehler zu vermeiden, da Fehler während der Laufzeit schon beim Programmieren auffallen.

Auslesen der Regler In den Einstellungen kann eingestellt werden, in welchem Intervall alle Regler ausgefragt werden. Dabei wird für jeden Regelkreis ein API Endpunkt abgefragt, wobei eine Zahl zurückgegeben wird. Diese wird in den Diagrammen auf der Hauptseite angezeigt.

Speichern der Routine ist nun die Wunschroutine erstellt worden, entweder Manuell oder Per API, so kann mit dem Button Ändern und Speichern "die Routine auf die Zentrale Steuerung gespielt werden. Dabei wird ein JSON File per API an den Raspberry pi gesendet. Das JSON File sieht wie folgt aus:

```
{  
    "UpdateID": 0,  
    "Features": {  
        "Regen": [  
            {  
                "id": 3,  
                "name": "Regen",  
                "sollvalue": 33,  
                "time": 11040,  
                "timeString": "03:04:00"  
            }  
        ],  
        "Wind": [{}],  
        "Sonne": [{}],  
        "Luftfeuchtigkeit": [{}],  
        "Temperatur": [{}]  
    }  
}
```

Listing 4.1.4: Json welches die Routine für das Backend beinhaltet

Die UpdateID zählt rau auf jedes einzelne Update identifizieren zu können. Im Features Array sind alle definierten Parameter definiert. Die Ids sind nicht chronologisch, denn die ActionList worauf dieses JSON basiert, sortiert bei jeder Aktualisierung die Reihenfolge der Aktionen, damit zu jeder Zeit die Richtige Reihenfolge gegeben ist. Innerhalb der jeweiligen Arrays (z.B. Regen) sind die Einzelnen Aktionen

definiert. Bei den anderen Parametern ist anstelle des Objekts „[]“ angegeben um hier eine bessere Übersichtlichkeit zu gewähren.

Um diese Funktionen zu realisieren werden sich bestimmte Konzepte zu nutzen gemacht:

Vue Components Im Vue Framework werden sogenannte „Components“ benutzt um den Code dynamischer, wiederverwendbarer, und aufgeräumter zu gestalten. Diese sind Codefiles die wie eine normale Vue Datei aufgebaut sind, jedoch in einen anderen HTML-Codeteil eingefügt werden können wie Funktionen. In dieser Diplomarbeit sind mehrere Components eingebaut. Am meisten benutzt wird der „Actionbutton“ und „Statusbox“. Mit diesen Components muss der Actionbutton wie z.B. auf der Seite der Manuellen Steuerung in Abbildung 4.1.6 und der Apply Button in Abbildung 4.1.8 nicht mehrmals definiert werden, sondern kann wie ein natives HTML Element eingesetzt werden und behält all seine Eigenschaften sowohl in Funktion als auch in Design.

Eingebunden werden diese Components folgendermaßen:

```
<template>
  <ActionButton></ActionButton>
</template>

<script setup>
import ActionButton from './components/ActionButton.vue';
</script>
```

Listing 4.1.5: Einbindung eines Components in Vue

Den Komponenten können auch Objekte oder Variablen mitgegeben werden um Dynamisches Design zu ermöglichen. Beispielsweise wird dem ActionButton ein Objekt eines Features - also eines Parameters - mitgegeben um den Text im Button zu definieren und Variablen zu bestimmen.

Verwendung von Stores Pinia gibt die Möglichkeit für Stores. Mit Stores können globale Variablen und Funktionen definiert werden. Somit kann eine Variable über das ganze Projekt benutzt werden, oder eine Funktion in jeder Datei verwendet werden. Ein Store wird wie folgt definiert:

```
import { defineStore } from 'pinia'

export const useStoreName = defineStore('StoreName', {
    state: () => ({
        x: 0 as number
    }),
    actions: {
        HelloWorld(){
            console.log('Hello World')
        }
    }
})
```

Listing 4.1.6: Definierung eines Pinia Stores

in „state“ können jegliche Variablen, Arrays, ... gespeichert werden. In „actions“ werden Funktionen definiert. Um die Stores in anderen Dateien zu benutzen benötigt es maximal zwei Zeilen. Alle im Store definierten Datenstrukturen können dann einfach ausgelesen und benutzt werden:

```
import { useStoreName } from './stores/globalStore';
const global = useStoreName()

let x = global.x
global.HelloWorld()
```

Listing 4.1.7: Pinia Store verwenden

4.1.5 Hinzufügen eines neuen Features

Die Erweiterung der Website um ein weiteres Feature benötigt nur wenige „Copy-Paste“ Handgriffe. Als Beispiel wird hier der Parameter „Wind“ benutzt. Im Folgenden wird ausschließlich die Anpassung des Frontends veranschaulicht.

Als erstes muss im „src/objects/Feature.ts“ ein neues Feature initialisiert werden. Dazu wird ausschließlich ein neues Feature Objekt erstellt, und anschließend in das „Feat[]“ Array hinzugefügt, wie in Abbildung 4.1.11.

Die „id“ beträgt einfach die id des letzten Features+1. Der Name wird überall Abgebildet wo das Feature verarbeitet wird. „value_name“ beschreibt die Einheit des

Parameters (Hier: Prozent). Mit „iconlink“ kann von Google Icons⁶⁶ ein Icon definiert werden. Zuletzt beschreibt „url“ die URL um Informationen von der API holen zu können.

```

64  export const Wind: Feature = {
65    id: 3,
66    name: 'Wind',
67    value_name: 'Prozent',
68    iconLink: 'mode_fan',
69    url: '/air/fanspeed',
70  };
71
72  export const Feat = [Sonne, Temperatur, Regen, Luftfeuchtigkeit, Wind]

```

Abbildung 4.1.11: neues Feature Initialisieren

Wenn man nun auf der Website den Automatische Steuerung Tab öffnet, sieht man wie in Abbildung 4.1.12 das neue Feature zu sehen ist.

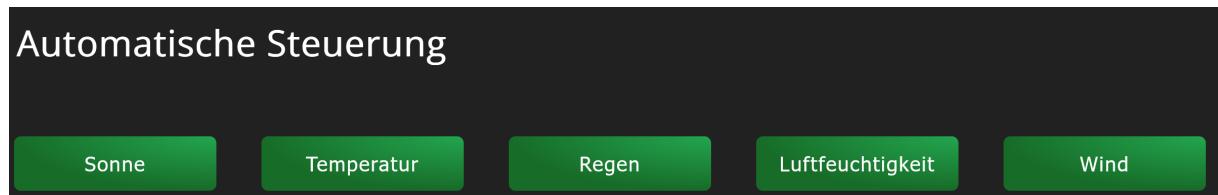


Abbildung 4.1.12: neues Feature wurde hinzugefügt

Nun ist das Feature Theoretisch schon einsatzbereit. Jedoch sollen die Soll- und Ist-Werte auch in Charts zu sehen sein, welche ein wenig Manuelle Copy-Paste Arbeit benötigen.

Erst muss erneut im „src/objects/Feature.ts“ das „StatusUpdate“ Interface wie in Abbildung 4.1.13 aktualisiert werden. Es muss ausschließlich im Stil der anderen Werte ein neues Attribut hinzugefügt werden.

⁶⁶ [GoogleIcons](#).

```
export interface StatusUpdate {  
    Time: number,  
    Sonne: string,  
    Regen: string,  
    Luftfeuchtigkeit: string,  
    Temperatur: string,  
    Tuer: string,  
    PSUstatus: string,  
    PSUVolt: string,  
    Wind: string,  
}
```

Abbildung 4.1.13: StatusUpdate anpassen

Nun muss in den ReglerStore bei „src/stores/CtrlLoopStore.ts“ gewechselt werden. Hier muss in der „getKammerValues()“ Routine getter und setter für den neuen Parameter definiert werden wie in Abbildung 4.1.15 gezeigt. Zusätzlich muss noch im selben File der Endpunkt der API definiert werden (Hier: WindURL) wie in Abbildung 4.1.14 zu sehen ist.

```
HumidURL: '/air/humidity',  
TempURL: '/air/temperature',  
WindURL: '/fan/fanspeed',  
DoorURL: '/misc/door',
```

Abbildung 4.1.14: API Endpunkt definieren

```

actions: {
    getKammerValues() {
        this.CurrentStatus.clear()
        const kammerValues: StatusUpdate = {
            Time: Date.now(),
            Sonne: this.httpGetValue(this.BackEndIP + this.SunURL),
            Regen: this.httpGetValue(this.BackEndIP + this.RainURL),
            Luftfeuchtigkeit: this.httpGetValue(this.BackEndIP + this.HumidURL),
            Temperatur: this.httpGetValue(this.BackEndIP + this.TempURL),
            Tuer: this.httpGetValue(this.BackEndIP + this.DoorURL),
            PSUstatus: this.httpGetValue(this.BackEndIP + this.PSUstatusURL),
            PSUvolt: this.httpGetValue(this.BackEndIP + this.PSUvoltURL),
            Wind: this.httpGetValue(this.BackEndIP + this.WindURL),
        }
        this.CurrentStatus.set('Time', kammerValues.Time)
        this.CurrentStatus.set('Sonne', kammerValues.Sonne)
        this.CurrentStatus.set('Regen', kammerValues.Regen)
        this.CurrentStatus.set('Luftfeuchtigkeit', kammerValues.Luftfeuchtigkeit)
        this.CurrentStatus.set('Temperatur', kammerValues.Temperatur)
        this.CurrentStatus.set('Tuer', kammerValues.Tuer)
        this.CurrentStatus.set('PSUstatus', kammerValues.PSUstatus)
        this.CurrentStatus.set('PSUvolt', kammerValues.PSUvolt)
        this.CurrentStatus.set('Wind', kammerValues.Wind)
        this.History.push(this.CurrentStatus)
    }
}

```

Abbildung 4.1.15: Getter und Setter definieren

Nun müssen noch im „src/stores/ChartStore.ts“ neue Graphen hinzugefügt werden, sowohl für den Soll Wert als auch für den Ist Wert wie in Abbildung 4.1.16

```

istRainData: {
    labels: [] as any[],
    datasets: [
        {
            label: 'istwert',
            data: [] as any[],
            borderColor: '#FFFFFF'
        },
    ],
},
istWindData: {
    labels: [] as any[],
    datasets: [
        {
            label: 'istwert',
            data: [] as any[],
            borderColor: '#FFFFFF'
        },
    ],
},
SunData: {
    labels: [],
    datasets: [
        {
            label: 'Sonne in %',
            data: [],
            borderColor: '#FFFFFF'
        }
    ],
} as any,
WindData: {
    labels: [],
    datasets: [
        {
            label: 'Wind in %',
            data: [],
            borderColor: '#FFFFFF'
        }
    ],
} as any,

```

Abbildung 4.1.16: Im ChartStore die Charts definieren

Außerdem muss in der updateCharts() Funktion die Winddaten hinzugefügt werden, damit diese laufend aktualisiert werden (Abbildung 4.1.17). die im ersten Bild ge-

zeigte „windlist“ muss jedoch noch im „src/stores/GlobalStore.ts“ definiert werden.

```

actions: {
    updateCharts() {
        //#region sollWerte
        this.TempData.labels = useGlobalStore().keyList
        this.TempData.datasets[0].data = useGlobalStore().tempList

        this.HumidData.labels = useGlobalStore().keyList
        this.HumidData.datasets[0].data = useGlobalStore().humidList

        this.RainData.labels = useGlobalStore().keyList
        this.RainData.datasets[0].data = useGlobalStore().rainList

        this.SunData.labels = useGlobalStore().keyList
        this.SunData.datasets[0].data = useGlobalStore().sunList

        this.WindData.labels = useGlobalStore().keyList
        this.WindData.datasets[0].data = useGlobalStore().windList
        //#endregion sollWerte

        keyList: [] as string[], // List of all Keys for the Graphs
        tempList: [] as number[], // List of all Temperatures for the Graphs
        humidList: [] as number[], // List of all Humidities for the Graphs
        rainList: [] as number[], // List of all Rain for the Graphs
        sunList: [] as number[], // List of all Sun for the Graphs
        windList: [] as number[], // List of all Wind for the Graphs
    }
}

```

Abbildung 4.1.17: *UpdateCharts() anpassen*

Wieder im ChartStore.ts muss noch definiert werden, dass die Ist-Werte in die Charts gesetzt werden wie in Abbildung 4.1.18 beschrieben

```

if (useGlobalStore().swift) {
    this.istHumidData.labels = this.istTime
    this.istHumidData.datasets[0].data = (this.StatusBoxCharts.get("Luftfeuchtigkeit") || []).slice(-useGlobalStore().samplesize);
    this.istRainData.labels = this.istTime
    this.istRainData.datasets[0].data = (this.StatusBoxCharts.get("Regen") || []).slice(-useGlobalStore().samplesize);
    this.istSunData.labels = this.istTime
    this.istSunData.datasets[0].data = (this.StatusBoxCharts.get("Sonne") || []).slice(-useGlobalStore().samplesize);
    this.istTempData.labels = this.istTime
    this.istTempData.datasets[0].data = (this.StatusBoxCharts.get("Temperatur") || []).slice(-useGlobalStore().samplesize);
    this.istWindData.labels = this.istTime
    this.istWindData.datasets[0].data = (this.StatusBoxCharts.get("Wind") || []).slice(-useGlobalStore().samplesize);
} else {
    this.istHumidData.labels = this.istTime
    this.istHumidData.datasets[0].data = this.StatusBoxCharts.get("Luftfeuchtigkeit") || []
    this.istRainData.labels = this.istTime
    this.istRainData.datasets[0].data = this.StatusBoxCharts.get("Regen") || []
    this.istSunData.labels = this.istTime
    this.istSunData.datasets[0].data = this.StatusBoxCharts.get("Sonne") || []
    this.istTempData.labels = this.istTime
    this.istTempData.datasets[0].data = this.StatusBoxCharts.get("Temperatur") || []
    this.istWindData.labels = this.istTime
    this.istWindData.datasets[0].data = this.StatusBoxCharts.get("Wind") || []
}

```

Abbildung 4.1.18: *StatusChart() anpassen*

Damit die alten Werte des Parameters immer simultan mit den anderen Parametern gelöscht werden, muss dies in der CleanWeather() Funktion wie in Abbildung 4.1.19 definiert werden.

```
CleanWeather() { // Cleans Weather Variables for new Routine
    console.log('CleanWeather')
    this.keyList = []
    this.tempList = []
    this.humidList = []
    this.sunList = []
    this.rainList = []
    this.windList = []
},
}
```

Abbildung 4.1.19: Cleanweather() Funktion anpassen

Zu guter Letzt muss der Chart noch im „src/components/StatusBox.vue“ wie in Abbildung 4.1.8 hinzugefügt werden.

```
30 |   <LineChart v-if="Global.PopupType == 'main' && pr.F?.name == 'Luftfeuchtigkeit'" style="width: 300px; margin-right: 20px;" :chart-data="useChartStore().istHumidData"></LineChart>
31 |   <LineChart v-if="Global.PopupType == 'main' && pr.F?.name == 'Wind'" style="width: 300px; margin-right: 20px;" :chart-data="useChartStore().istWindData"></LineChart>
32 |
33 |
```

Listing 4.1.8: LineChart im StatusBox.vue einfügen

Nun wurde ein neues Feature erfolgreich hinzugefügt.

Ausblick Als Ausblick für die Website wäre eine Datenbankanbindung, bei der man jegliche Messdaten langfristig speichern könnte. Dazu müsste man am Raspberry einen SQL Server aufsetzen, der die Daten laufend speichert. Zusätzlich wäre eine leichte Erweiterung der Website, dass die Konfiguration gespeichert wird. Man müsste ausschließlich sobald die Website geladen ist, einen API Call zum Raspberry senden, der die zuletzt gesendete Konfiguration sendet. Ein weiterer Ausblick wäre es, die Website korrekt zu hosten. Derzeit wird sie dem Befehl „npm run serve“ lokal gehostet. Dieser Befehl führt einen Vue Command „vue-cli-service serve“ aus welcher mit Vue installiert wird.

4.2 API

Um der Website die Möglichkeit zu geben, mit dem Raspberry Pi und somit mit allen Modulen kommunizieren zu können, wird eine API benötigt. Diese soll auf dem Raspberry Pi gehostet werden, und als Zentrale zwischen den Modulen und der

Website fungieren. Dabei fiel unsere Wahl auf das FastAPI-Framework⁶⁷. FastAPI ist ein Framework, welches es ermöglicht in Python APIs zu programmieren. Die Vorteile dabei bestehen vor allem in der Einfachheit und Erweiterbarkeit der Programmiersprache Python.

Grundlagen von FastAPI Nachdem FastAPI mittels „pip“ installiert und in den Python-Code eingebunden wurden, muss es zuerst initialisiert werden, wie in Abbildung 4.2.1 zu sehen.

```
app = FastAPI(
    title="Klimakammer API",
    description=description,
    summary="API for controlling the Klimakammer",
    version="0.0.1",
    license_info={
        "name": "GPL-3.0 License",
        "url": "https://www.gnu.org/licenses/gpl-3.0.html",
    },
    openapi_tags=tags_metadata
)
```

Listing 4.2.1: Initialisierung von FastAPI

Dabei können verschiedene Parameter angegeben werden, in unserem Fall:

- **title:** Name der API
- **description:** Beschreibung der API, unterstützt Markdown Formatierung
- **summary:** Weitere Beschreibung der API, anders formatiert
- **version:** Version der API
- **license_info:** Info über die verwendete Lizenz der API
- **openapi_tags:** Beschreibungen für die verschiedenen Gruppen von Endpoints

CORS Dann muss die API für „CORS“ vorbereitet werden. „CORS“ steht für „Cross-Origin-Ressource-Sharing“ was Webserver den Zugriff auf andere Webserver mit einer anderen Domain erlaubt. Diese sind nämlich durch „SOP“, „Same-Origin-Policy“ standardmäßig nicht möglich, da dies ein Sicherheitsrisiko darstellt. Wie in Abbildung 4.2.2 zu sehen, müssen dafür zuerst die „Origins“ definiert werden, also woher die Anfragen erlaubt werden sollen. Dann können sie der API hinzugefügt werden. Es müssen aber keine „Origins“ definiert werden, sondern es kann, wie von uns bei den Methoden und den Headers verwendet, die Wildcard „[“*”]“ verwendet

⁶⁷ FastAPI. URL: <https://fastapi.tiangolo.com/> (besucht am 19.03.2024).

werden, welche alle Möglichkeiten erlaubt.

```
origins = [
    "http://localhost.tiangolo.com",
    "https://localhost.tiangolo.com",
    "http://localhost",
    "http://localhost:8080",
]

app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

Listing 4.2.2: CORS in FastAPI hinzufügen

Endpunkte definieren API-Endpunkte werden in FastAPI sehr einfach definiert, wie in Abbildung 4.2.3 zu sehen ist. Dabei muss die vorhin erstellte Instanz mit dem „@“-„Decorator“ und der gewünschten „Request-Methode“ als Methode der Instanz aufgerufen. Danach muss eine Funktion definiert werden, welche ausgeführt wird, wenn der eben erstellt Endpoint aufgerufen wird. Der Rückgabewert dieser Funktion wird an den Server welcher die API aufgerufen hat zurückgegeben.

```
@app.get("/air/pressure", tags=["Air"])
def get_air_quality():
    data = get_data("Sensors", "AirPressure")
    return {"AirPressure": data}
```

Listing 4.2.3: Endpunkt in FastAPI

Für API-Endpoints bei denen Daten mit übergeben werden, können „Pydantic-Modelle“ verwendet werden. Pydantic ist eine Bibliothek für Python welche zur Datavalidierung verwendet wird. In Abbildung 4.2.4 ist zu sehen, wie eine Klasse erstellt wird, mit welcher die Daten welche in der Klimakammer eingestellt werden sollen gespe-

chert werden sollen.

```
class Value(BaseModel):
    intensity: int
    time: int

class ScheduleSet(BaseModel):
    UpdateID: int
    Sonne: list[Value] | None = None
    Regen: list[Value] | None = None
    Wind: list[Value] | None = None
```

Listing 4.2.4: Klassen für API-Objekte

Die übergebenen Daten vom Server an die API können einfach als Parameter der Funktion entgegengenommen werden, wobei der Typ des Parameters die eben erstellte Klasse ist. Falls das Format der gesendeten Daten nicht mit dem Format der Klasse übereinstimmen, gibt der Server den Fehlercode „422 Unprocessable Entity“ zurück.

Daten von den Modulen lesen Um auf den I2C-Bus des Raspberry Pis in Python zugreifen zu können, wird die Bibliothek SMBUS2 benötigt. Es wird eine Funktion „get_data“ (Abbildung 4.2.5) erstellt, welche als Überabeparameter den Name des Moduls und das Register das gelesen werden soll hat. Diese werden dann zuerst in Adressen umgewandelt, bevor damit 8 Byte vom Modul angefordert werden. 8 Byte, weil ein Float-Datentyp 4 Byte groß ist, und zwei davon zurückgegeben werden. Diese werden dann zusammengesetzt und als Rückgabewert zurückgegeben. Im Code ist noch ein Workaround vorhanden, mit welchem die Daten immer doppelt gelesen werden, und das erste Ergebnis verworfen wird, weil die Firmware immer einen Zyklus hinterher hängt, und wir dieses Problem nicht anders beheben konn-

ten.

```
# Change to use 32bit floats
def get_data(module, sensor):
    module_address = modules.get(module)
    sensor_code = sensors.get(sensor)
    if not is_raspberry_pi():
        return randrange(100), randrange(100)
    bus = SMBus(1)
    # Workaround for bug in Firmware, read is always one step behind
    bus.read_i2c_block_data(module_address, sensor_code, 8)
    time.sleep(0.1)
    data_bytes = bus.read_i2c_block_data(module_address, sensor_code, 8)
    bus.close()
    data_bytes1 = data_bytes[0:4]
    data_bytes2 = data_bytes[4:8]
    data1 = struct.unpack("<f", bytes(data_bytes1))[0]
    data2 = struct.unpack("<f", bytes(data_bytes2))[0]
    return data1, data2
```

Listing 4.2.5: Code um Daten von den Modulen zu lesen

Daten an die Module senden Um Daten an die Module zu senden, wurde ein einziger Endpoint erstellt, welcher den „Schedule“ der Module entgegen nimmt. Dieser wird dann als JSON auf dem Raspberry gespeichert, und von einem anderen Python-Code bearbeitet.

Bearbeiten des Schedules Der gespeicherte Schedule der Module wird minütlich von einem Python-Code eingelesen. Falls einer der Einträge älter als die jetzige Uhrzeit ist, werden wird der darin gespeicherte Intensity-Wert an das entsprechende Modul gesendet und der Eintrag gelöscht. So können minütlich neue Werte gesetzt werden. Die minütliche Bearbeitung der Werte wird mit einem „Cronjob“ gelöst. Mit Cronjob können Prozesse auf Linux zu festen Zeiten durchgeführt werden. Dabei wurde mit Crontab (einem Tool zur Verwaltung von verschiedenen Cronjobs)

der Job so mit dem Command:

```
* * * * * python3 /usr/klimakammer/klimakammer-API/schedule.py
```

konfiguriert dass er jede Minute ausgeführt wird.

Starten der API In unserem Fall wird die API mit „Uvicorn“⁶⁸ gehostet. Uvicorn ist ein Webserver für Python-Anwendungen. In unserem Fall wird der Server mit `uvicorn main:app` gestartet. Dass der Server bei jeden Systemstart automatisch startet, wurde ein Service geschrieben. Dieser muss unter `/etc/systemd/system/` gespeichert werden, und ist in Abbildung 4.2.6 zu sehen. Der Nutzernname ist in dieser Abbildung „ubuntu“ weil der Screenshot nicht von Raspberry Pi stammt, sondern von einem bei AWS gehosteten Server, der für Debugzwecke verwendet wird, welcher weiter unten beschrieben wird.

```
[Unit]
Description=Uvicorn Klimakammer API
After=network.target

[Service]
WorkingDirectory=/home/ubuntu/klimakammer-API
ExecStart=/usr/bin/python3 -m uvicorn main:app
User=ubuntu
Group=ubuntu
Restart=always

[Install]
WantedBy=multi-user.target
```

Listing 4.2.6: *Systemd Service*

Weiterleiten der Adresse mit NGINX Da die API nur auf dem Localhost gehostet werden kann, Anfragen von außen aber auf die IP-Adresse des Raspberry Pis ankommen, müssen diese auf den Localhost weitergeleitet werden. Wir verwenden dafür NGINX⁶⁹. NGINX ist ein Reverse-Proxy, er gewährt also Anfragen von außen Zugriff auf interne IP-Adressen. Die Konfiguration kann aus Abbildung 4.2.7 entnommen werden. Auch diese Konfiguration stammt wieder vom AWS-Server, unterscheidet sich aber nur beim „server_name“ vom Raspberry Pi. Dies ist nämlich die IP-Adresse des Gerätes.

⁶⁸ Uvicorn. URL: <https://www.uvicorn.org/> (besucht am 19.03.2024).

⁶⁹ Advanced Load Balancer, Web Server, & Reverse Proxy. NGINX. URL: <https://www.nginx.com/> (besucht am 19.03.2024).

```
server {
    listen 80;
    server_name 13.48.59.20;
    location / {
        proxy_pass http://127.0.0.1:8000;
    }
}
```

Listing 4.2.7: Konfiguration von NGINX

Aufrufen der Dokumentation Ein weiterer Vorteil von FastAPI ist dass automatisch eine OpenAPI⁷⁰ Dokumentation erstellt und gehostet wird. Diese ist unter dem Endpoint /docs verfügbar, und ist in Abbildung 4.2.1 sichtbar. Dabei werden alle Endpoints angezeigt, und auch wie in Abbildung 4.2.2 sichtbar, die Möglichkeit zum testen des Endpoints gegeben.

Klimakammer API 0.0.1 OAS 3.1

/openapi.json

API for controlling the Klimakammer

All returnvalues are in the range of 0-100.
Inputvalues are in the range of 0-100, except for the schedule starttime and endtime, which are in the range of 0-4294967295 (Unix timestamp).

GPL-3.0 License

Air Endpoints related to air quality and control

^

GET	/air/quality	Get Air Quality	▼
GET	/air/co2	Get Air Co2	▼
GET	/air/temperature	Get Air Temperature	▼
GET	/air/humidity	Get Air Humidity	▼
GET	/air/fanspeed	Get Air Fanspeed	▼

Abbildung 4.2.1: Dokumentation der API

⁷⁰ OpenAPI Specification - Version 3.1.0 | Swagger. URL: <https://swagger.io/specification/> (besucht am 19.03.2024).

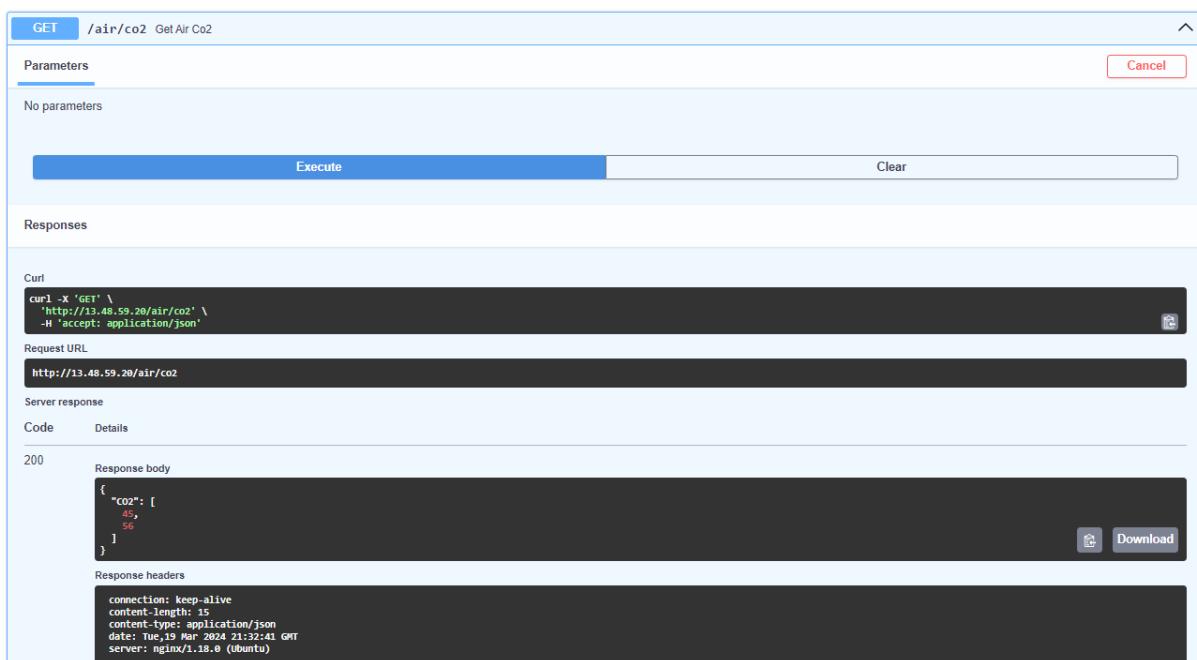


Abbildung 4.2.2: Testen der API

Hosten der API bei AWS Um bei der Entwicklung der Website immer auf die API zugreifen zu können, wurde die API auf einem Server bei AWS gehostet. Dieser Server ist äußerst kostengünstig (ca. 5€ im Monat) und wartungsfrei. Dabei wird eine öffentliche Adresse zur Verfügung gestellt, über welche die API dann aufgerufen werden kann. Da die API auf den I2C-Bus zugreifen will, dieser aber natürlich bei einem gemieteten Server nicht verfügbar ist, musste der Code so abgeändert werden, dass er erkennt ob er auf einem Server oder dem Raspberry Pi ausgeführt wird, und falls er auf dem Server ausgeführt wird, Dummywerte zurück gibt. Diese Erkennung erfolgt mit der Python Bibliothek „platform“ mit welcher einige Systemparameter abgefragt werden können. Dabei wurde von uns abgefragt, ob die Architektur der CPU „Aarch64“ ist, und ob das Betriebssystem Linux ist. Da der AWS-Server wie in Abbildung 4.2.3 zu sehen, einen Intel Xeon Kern , also x86-Architektur hat, wird der Dummycode ausgeführt.

```
ubuntu@ip-172-31-43-122:~$ neofetch
      .-/+oossssoot/-.
      `:+ssssssssssssssssssssss+`:
      -+ssssssssssssssssssssyssss+-+
      .osssssssssssssssssdMMMNysssso.
      /sssssssssssshdmmNNmmyNMMMHssssss/
      +ssssssssshmydMMMMMMNddddyssssssss+-
      /sssssssshNMMyhyyyyhmNMMNhssssssss/
      .ssssssssdMMMNhsssssssssshNMMMdssssssss.
      +sssshhhyNMMNyssssssssssyNMMMyssssssss+-
      ossyNMMNyMMhsssssssssssssshmmhssssssso
      ossyNMMNyMMhsssssssssssssshmmhssssssso
      +sssshhhyNMMNyssssssssssssyNMMMyssssssss+-
      .ssssssssdMMMNhsssssssssshNMMMdssssssss.
      /sssssssshNMMyhyyyyhdNMMNhssssssss/
      +ssssssssshmydMMMMMMNdyyssssssss+-
      /sssssssssssshdmmNNmmyNMMMHssssssss/
      .osssssssssssssssdMMMNysssso.
      -+ssssssssssssssssyssss+-+
      `:+ssssssssssssssssss+`:
      .-/+oossssoot/-.
```

```
ubuntu@ip-172-31-43-122
-----
OS: Ubuntu 22.04.3 LTS x86_64
Host: t3.micro
Kernel: 6.5.0-1014-aws
Uptime: 18 hours, 35 mins
Packages: 765 (dpkg), 6 (snap)
Shell: bash 5.1.16
Terminal: /dev/pts/0
CPU: Intel Xeon Platinum 8259CL (2) @ 2.500GHz
GPU: 00:03.0 Amazon.com, Inc. Device 1111
Memory: 215MiB / 924MiB
```



Abbildung 4.2.3: Systemparameter des AWS-Servers

Down-Checker mit Betterstack.com Um die Verfügbarkeit des AWS-Servers zu checken, um im Falle eines Ausfalls sofort reagieren zu können, wurde die IP-Adresse des Servers bei „BetterStack“ eingetragen. Mit BetterStack kann die Verfügbarkeit eines Services überwacht werden, und bei Ausfall die zuständigen Personen alarmiert werden. Außerdem wird eine öffentliche Statusseite des Services angeboten, welche in Abbildung 4.2.4 zu sehen ist.

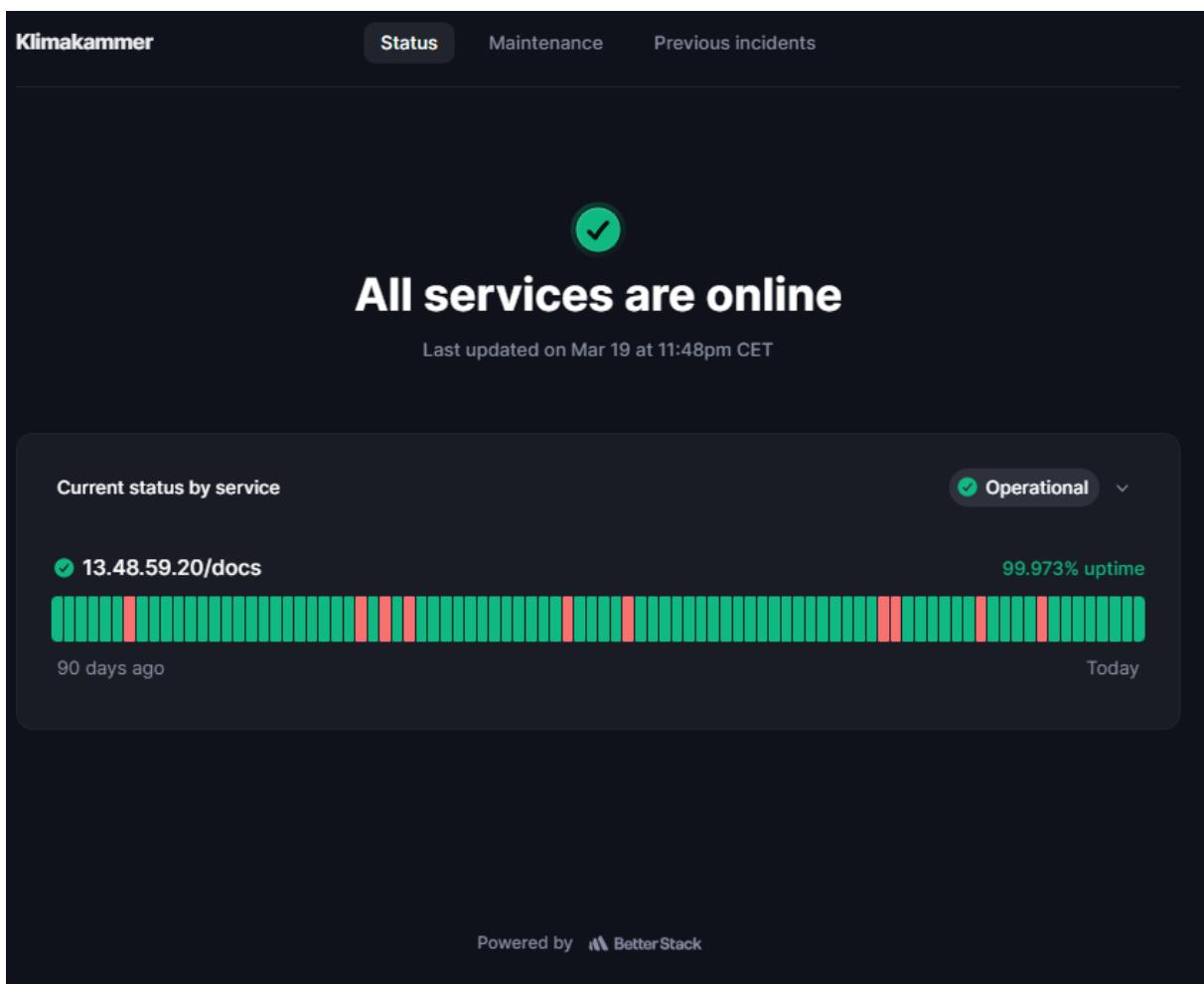


Abbildung 4.2.4: Statusseite des Servers bei BetterStack.com

4.2.1 Sprachsteuerung

Quellcode⁷¹

Im Rahmen des KI Unterrichts sollte ein Projekt mittels Künstlicher Intelligenz realisiert werden. Die erste Idee war ein Routinenersteller, der mittels Chat eine Parameterroutine erstellt. Diese Idee stellte sich aus Leistungs- und Datenerfassungsgründen als zu hochgesteckt heraus. Alleine die Beschaffung von genug Daten um dieses Projekt sinnvoll einzubauen zu können hätte zu viel Zeit in Anspruch genommen, ohne eine Garantie dass diese Funktion wirklich funktioniert.

Entschlossen wurde sich schlussendlich eine Sprachsteuerung der Parameter einzulegen.

⁷¹ Kikomitk. Tee6/improved-Klima-SST. original-date: 2024-03-19T16:30:39Z. 19. März 2024. URL: <https://github.com/Tee6/improved-Klima-SST> (besucht am 04.04.2024).

binden. Erstens ist der Leistungsbedarf dieser Anwendung im Vergleich zu anderen Projekten relativ klein, zudem ist die Datenbeschaffung redundant, bei den Unmengen an Daten im Internet.

Spracherkennung kann mit mehreren Methoden erfolgen. Zum einen gibt es fertige Libraries mit denen man in einer Zeile eine Audio Datei transkribieren kann. Diese Methode war für das Projekt nicht geeignet, da ein gewisser Schwierigkeitsgrad vom Lehrer vorgegeben wurde. Ein anderer Ansatz war CMU Sphinx. Eine Bibliothek in C die aufwendig konfiguriert und trainiert werden muss. Dieser Ansatz wurde wegen des hohen Datenerfassungsaufwands verworfen. Außerdem ist CMU Sphinx nach Bewertungen im Internet nicht Benutzerfreundlich, da oft sehr klar und deutlich und in einer hohen Lautstärke gesprochen werden muss, dass der Satz annähernd richtig erkannt wird.

Geplant war, die Sprachsteuerung mittels OpenAI Whisper⁷² zu realisieren. Dieses KI Modell stellte sich jedoch als zu groß und langsam für unser System heraus. Das Open-Source Modell „faster-whisper“⁷³ stellte sich als leistungsstark und nicht speicherintensiv heraus, weshalb der Code auf dieses Modell umgeschrieben wurde. Der Umstieg ist jedoch nicht kompliziert, faster-whisper benutzt annähernd den selben Syntax wie OpenAI-Whisper

Grundsätzlich funktioniert Whisper so, dass es die Audio Datei in kleinere Segmente teilt. Von diesen Segmenten wird das Spektrum analysiert. Durch die vielen Daten mit denen das Modell trainiert wurde, kann Whisper anhand des Spektrums erkennen welche Buchstaben oder Wörter gesagt wurden. Ein großer Vorteil von Whisper ist außerdem, dass es das vorherige Wort in die Vorhersage des nächsten Wortes miteinbezieht, und somit um einiges Genauer ist als andere kleinere Modelle.

Genauer kann das Whisper-Modell von OpenAI in Anbetrach des Blockschaltbildes aus Abbildung 4.2.5 erklärt werden

⁷² *openai/whisper*. original-date: 2022-09-16T20:02:54Z. 6. März 2024. URL: <https://github.com/openai/whisper> (besucht am 06.03.2024).

⁷³ *SYSTRAN/faster-whisper*. original-date: 2023-02-11T09:17:27Z. 6. März 2024. URL: <https://github.com/SYSTRAN/faster-whisper> (besucht am 06.03.2024).

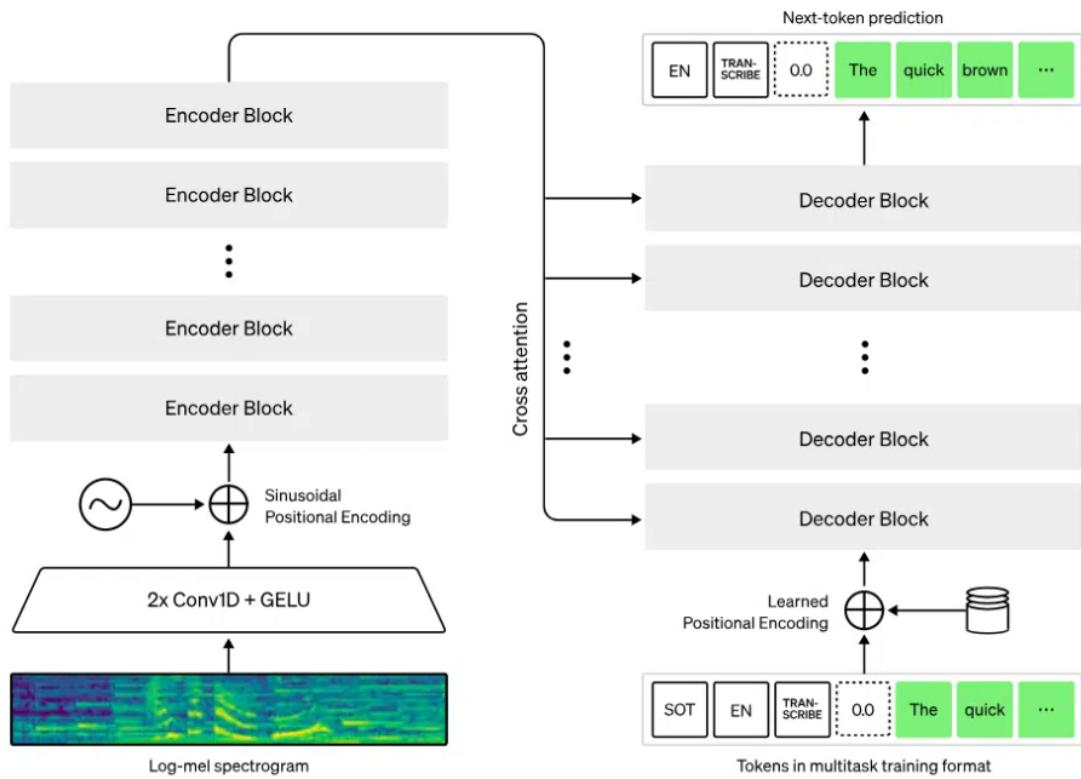


Abbildung 4.2.5: „Grobes Blockschaltbild der Funktion des Whisper-Modells (quelle)“

Wie bereits erwähnt werden die Audiodateien in kleinere Schnipsel unterteilt, genau gesagt in 30 Sekunden lange. Diese werden dann in ein logarithmisches Spektrum umgewandelt. Das bedeutet ein Mensch kann damit nichts mehr anfangen, für einen Rechner allerdings, bietet das erst die Möglichkeit mit der Berechnungen und Analysen durchzuführen. Dieses Spektrum wird in weiterer Folge durch 2 Convolutional Layer weiterverarbeitet. Einfach gesagt: Es werden 2 Filter auf das „Bild“, was in diesem Fall das Spektrum ist, angewandt. Das bringt insofern etwas, da Störungen wie Rauschen oder Stimmen im Hintergrund herausgefiltert werden. Das gefilterte Spektrum wird dann den Encoder-Blöcken übergeben, diese sind dafür Zuständig aus dem Spektrum Worte zu erkennen. Wurden alle Worte erkannt, ist es die Aufgabe der Decoder-Blöcke die erkannten Wörter in einen Kontext zu stellen. Deswegen ist es auch wichtig, dass Whisper, wie bereits erwähnt, das vorhergehende Wort noch einmal in die Kontexterkennung miteinbezieht.

Im Prinzip funktioniert die Sprachsteuerung wie folgt: Auf der Website ist unten rechts ein Button, beim Drücken dieses Buttons kann ein Befehl wie zum Beispiel „Bilde das Wetter in Rankweil nach“ eingesprochen werden. Dieser Satz wird an das

AWS Backend gesendet, wo der Befehl mittels KI verarbeitet wird. Die Antwort wird wieder zur Website geschickt und in die entsprechenden Parameter umgewandelt.

Um die Zentrale Steuereinheit zu entlasten, läuft im Hintergrund ein AWS Server der die Spracherkennung behandelt.

Frontend Quellcode auf GitHub⁷⁴

Im Frontend ist wie bereits erwähnt ein Button implementiert, wie in Abbildung 4.1.9 unten rechts zu sehen ist. Wird dieser gedrückt, wird man gefragt, ob das Mikrofon verwendet werden darf. Auf Zusage beginnt die Aufnahme. Wenn nun ein Befehl eingesprochen wird, und anschließend der Button nochmals gedrückt wird, wird das Audio in einen sogenannten Blob (= Binary Large Object) umgewandelt. Dieser Blob wird anschließend als FormData an die API mittels HTTP Post versendet. Dieser Blob wird im Backend verarbeitet, während das Frontend auf eine Antwort wartet. Der Vorteil, wenn die Verarbeitung auf das Backend verlegt wird liegt dabei, dass das Frontend flüssig weiter läuft, ohne dass der User Leistungseinbrüche spürt.

Backend Quellcode auf Github⁷⁵

Im Backend wird das Faster-Whisper Modell angewendet und aus der Audio-Datei, welche vom Frontend übergeben wird, eine Text-Vorhersage getätigt. Das Ganze wird in der Programmiersprache Python programmiert. Das Faster-Whisper Modell wäre auch in Javascript einbindbar, jedoch ist der Befehlssatz in Python erweitert und einfacher.

In Abbildung 4.2.8 ist die Funktion executeWhisper() ersichtlich. Mit diesen wenigen Code-Zeilen ist es möglich aus einem Audiofile das Gesprochene zu interpretieren („vorherzusagen“). In der Variable detected_text wird die Vorhersage als Zeichenkette (string) gespeichert und schlussendlich zurückgegeben.

⁷⁴ Kikomitk, Tee6/Diplomarbeit_Klimakammer.

⁷⁵ Simon Koller. InZ8ne/Speech-to-Klima. original-date: 2024-03-04T07:15:28Z. 4. März 2024. URL: <https://github.com/InZ8ne/Speech-to-Klima> (besucht am 05.04.2024).

```

def executeWhisper(audioFile):
    #Load model and audio
    model = WhisperModel("tiny", device="cpu", compute_type="int8")
    segments, info = model.transcribe(audioFile, beam_size=5)

    for segment in segments:
        print("[%.2fs -> %.2fs] %s" %
              (segment.start, segment.end, segment.text))
        result = segment.text

    # print the recognized text
    print(result.text)
    detected_text = result.text.lower()
    return detected_text

```

Listing 4.2.8: Funktion zur Ausführung einer Spracherkennung

Diese Zeichenkette durchläuft dann Filter-Funktionen um die relevanten Informationen zu extrahieren. Diese sind in Abbildung 4.2.9 dargestellt.

```
def executeWhisper(audioFile): ...
```

```
def GetRegexTime(text): ...
```

```
def GetRegexIntensity(text): ...
```

```
def GetRegexPlace(text): ...
```

```
def Get CommandType(text, keywords): ...
```

```
def Get Feature(text, keywords): ...
```

```
def TimeToNumerical3(time): ...
```

Listing 4.2.9: Funktionen für die Filtrierung der Parameter

Mit Regular Expressions lassen sich aus Zeichenketten leicht Informationen extrahieren. Das „Regex“ in den Funktionsnamen signalisiert, dass darin Informationen mithilfe von ebendiesen Regular Expressions herausgefiltert werden. Diese Funktionen geben jeweils die extrahierte Funktion zurück. So ist am Schluss bekannt welches Feature umgestellt werden soll und zu welcher Intensität, ob Planungsmodus, instantan oder der Live-Modus erwünscht ist. Ebenfalls ist danach bekannt in wie viel Sekunden (relative Unix-Time zum Zeitpunkt der Berechnung) das Feature umgestellt werden soll (falls der Planungsmodus angesprochen wird). Falls der Live-Modus angesprochen wird, filtert der Algorithmus auch den gewünschten Ort aus der Zeichenkette.

Es kann passieren, dass ausgesprochene Zahlen vom KI-Modell als ausgeschriebene Zahlen in die Zeichenkette eingefügt werden. Die Funktion TimeToNumerical13() sorgt dafür, dass diese dann wieder in Integers umgewandelt werden.

Die gefilterten Parameter werden zum Schluss in ein Json-Objekt gepackt, mit dem das Frontend dann wieder weiter arbeiten kann. In Abbildung 4.2.10 ist ersichtlich wie dieses Objekt erstellt wird.

```
#Write all data into a dictionary
dictionary = {
    "command": CommandType,
    "feature": Features[0],
    "data": [
        {
            "value": Intensity,
            "Time": relative_unixTime
        },
        {
            "value": Intensity
        },
        {
            "place": Place
        }
    ]
}
```

Listing 4.2.10: Erstellung eines Json-Objektes (Python-Dictionary)

AWS Server Der AWS Server für die Sprachsteuerung dient der Entlastung der

Zentralen Steuerung. Dazu wurde ein Linux Server mit AWS (Amazon Web Services) aufgesetzt. Dieser läuft 24/7 und nimmt unter seiner Öffentlichen IPv4 Adresse „16.171.30.95“ unter dem API Endpunkt „/translate“ eine Audiodatei im mp3 Format an. Der Code welcher auf diesem Server läuft ist auf dem GitHub Repository „improved-Klima-SST⁷⁶“ zu sehen. Zustätzlich wurde konfiguriert, dass beim start des Servers automatisch eine bash Datei ausgeführt wird, welche die API startet. Dies wird mit Crontab realisiert.

Eine große Herausforderung des AWS Servers war die Konfiguration mit NGINX. Obwohl die Konfiguration gleich wie beim Backend Server vorgenommen wurde, funktionierte die API nicht. Die Lösung war, in der Datei „/etc/nginx/nginx.conf“ die selbe Konfiguration wie in 4.2 unter „http“ einzufügen.

Ergebnisse Um den Code zu testen, wird er lokal über Uvicorn und VSCode gehostet. Wenn man den Localhost als STT-URL im Source Code der Website definiert, kann somit die Sprachsteuerung getestet werden.

Im Folgenden sind die Ergebnisse zu sehen:

⁷⁶ Kikomitk, Tee6/improved-Klima-SST.

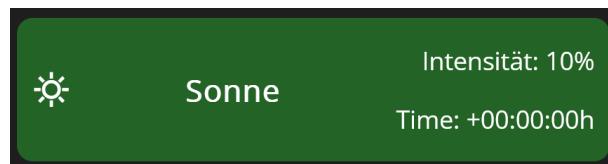


Abbildung 4.2.6: gesprochener Befehl: „Stelle die Sonne auf 10 Prozent“

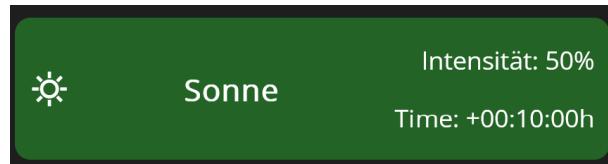


Abbildung 4.2.7: gesprochener Befehl: „Stelle die Sonne in 10 Minuten auf 50 Prozent“

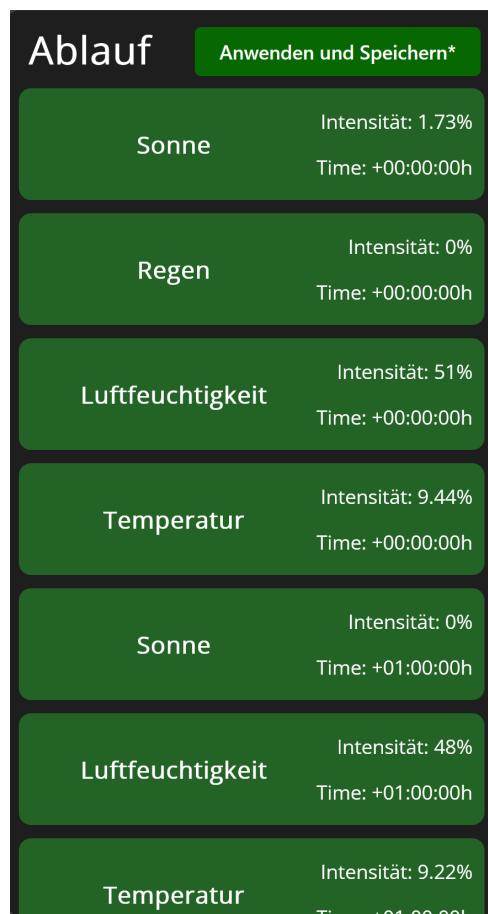


Abbildung 4.2.8: gesprochener Befehl: „Bilde das Wetter in Wien ab“

Abbildung 4.2.9: Sprachsteuerung Ergebnisse

Derzeit ist die Konfiguration des AWS Servers bis auf die einstellung von NGINX fertiggestellt. NGINX wird dazu benutzt, die Öffentliche IPv4 Adresse von Amazon zu benutzen, um auf die API zugreifen zu können.

Ausblick Als Ausblick wäre eine Erkennung des Commands mittels Large Language Models möglich, zum Beispiel mit der API von OpenAI. Ein „LLM“ selbst zu hosten und Trainieren ist sehr Leistungsintensiv, deshalb wäre die API ein Lösungsansatz.

5 Firmware

Bei der Programmierung der Firmware standen verschiedene SDKs zur Auswahl.

- **Pico SDK:** Beim Pico SDK⁷⁷ handelt es sich um eine Programmierumgebung, welche von Raspberry Pi zur Verfügung gestellt wird, und die Programmierung des RP2040 in C, C++ und Assembly ermöglicht.
- **Arduino IDE:** Ein quelloffenes IDE, welches die Programmierung von Mikrocontrollern mit dem Arduino Framework erlaubt. Das Arduino Framework erlaubt es, dass ein Code auf allen Arduino-kompatiblen Mikrocontrollern ausführbar ist.
- **PlatformIO:** Mit PlatformIO können verschiedene Frameworks verwendet werden, um Mikrocontroller zu programmieren. Es kann in verschiedene Editoren wie z.B. Visual Studio Code eingebunden werden.

In unserem Fall wurde PlatformIO in Visual Studio Code eingebunden, und als Framework das Arduino Framework verwendet. Dabei gibt es beim RP2040 zwei verschiedene Implementationen des Arduino Frameworks:

- **Arduino Core:** Der Arduino Core basiert auf Mbed OS, einem Realtime-Betriebssystem für Mikrocontroller
- **Philhower Core:** Der „Philhower-core“⁷⁸ funktioniert direkt auf dem RP2040 ohne die Zwischenstufe eines Betriebssystems.

Zuerst wurde der Arduino Core verwendet, welcher in Platformio standardmäßig aktiviert war. Dabei wurde aber bemerkt dass, sobald der Code etwas größer wurde, das Betriebssystem abstürzte, was sich dadurch äußert, dass die Onboard-LED 4 mal kurz, und dann 4 mal lang blinkte. Außerdem wurde insbesondere beim Verwenden des zweiten I2C-Busses bemerkt, dass er schlecht implementiert ist, da der Quellcode des Frameworks umgeschrieben werden musste, um ihn zu verwenden. Dann wurde auf den „Philhower-Core“ umgestellt, und beide vorherigen Probleme wurden sofort gelöst. Wenn im folgenden das Arduino-Framework erwähnt ist, ist damit das Arduino Framework mit „Philhower-Core“ gemeint.

⁷⁷ *picosdk*. URL: https://www.raspberrypi.com/documentation/pico-sdk/index_doxxygen.html (besucht am 19.03.2024).

⁷⁸ Earle F. Philhower III. *earlephilhower/arduino-pico*. original-date: 2021-02-25T04:20:27Z. 19. März 2024. URL: <https://github.com/earlephilhower/arduino-pico> (besucht am 19.03.2024).

5.1 Vorlage

Quellcode⁷⁹

Da alle Steuerplatinen im Grunde genommen die gleiche Aufgabe haben, nämlich Daten von Raspberry entgegen zu nehmen, und Daten an den Raspberry Pi zurück zu senden, wurde eine Codevorlage geschrieben, welche dies bereits macht. Vom Arduino-Framework gibt es für den I2C-Slave Modus zwei Callbackfunktionen:

- „**onReceive()**“: Diese Funktion wird aufgerufen wenn die I2C-Adresse mit Write-Bit gesendet wird. Der Raspberry Pi will also Daten übermitteln/schreiben
- „**onRequest()**“: Diese Funktion wird aufgerufen wenn die I2C-Adresse mit Read-Bit gesendet wird. Es sollen also Daten an den Raspberry Pi zurückgesendet werden.

Im folgenden wird die Funktionsweise dieser zwei Funktionen erklärt.

⁷⁹ [Joe105.Joe105/klimakammer-firmware](https://github.com/Joe105/klimakammer-firmware). original-date: 2024-02-12T21:21:55Z. 13. Feb. 2024. URL: <https://github.com/Joe105/klimakammer-firmware> (besucht am 05.04.2024).

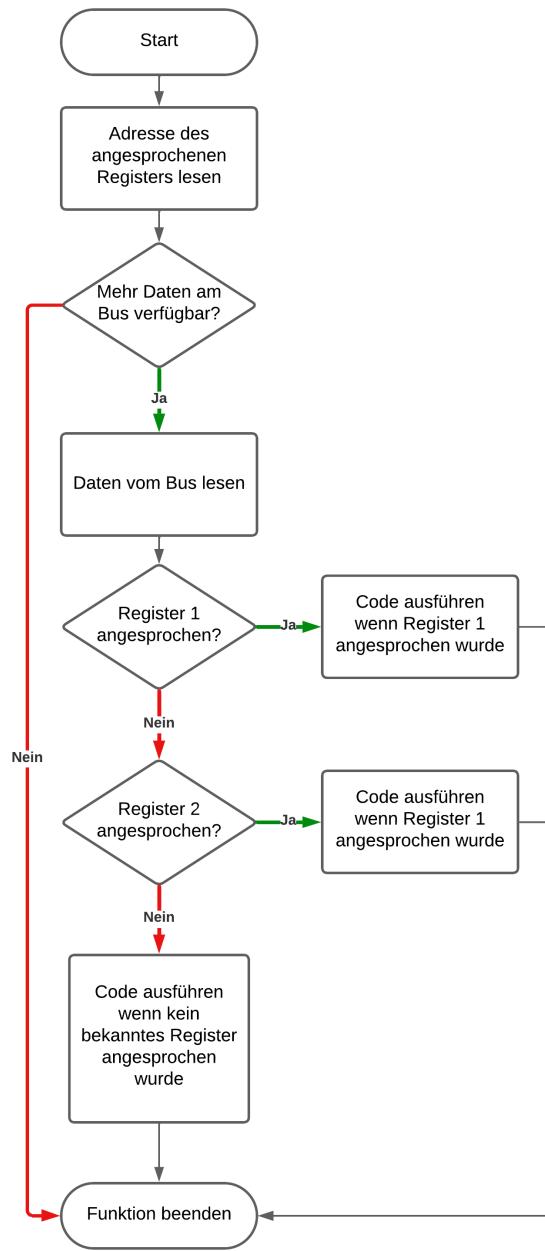


Abbildung 5.1.1: Flussdiagramm der „onReceive“Funktion

„onReceive“ Wie in Abbildung 5.1.1 zu sehen, wird zuerst die Adresse des angeprochenen Registers gelesen. Dann erfolgt eine Abfrage, ob am Bus mehr Daten verfügbar sind. Diese Abfrage ist nötig, weil wie in Abbildung 5.1.2 zu sehen, wenn Daten vom Slave gelesen werden sollen, er zuerst im Write-Modus adressiert wird und mit der Adresse des Registers beschrieben wird, und erst dann im Read-Modus aufgerufen wird. Falls also vom Pico gelesen wird, muss die Adresse des gewünschten Registers zwischengespeichert und die „onReceive“Funktion beendet werden.

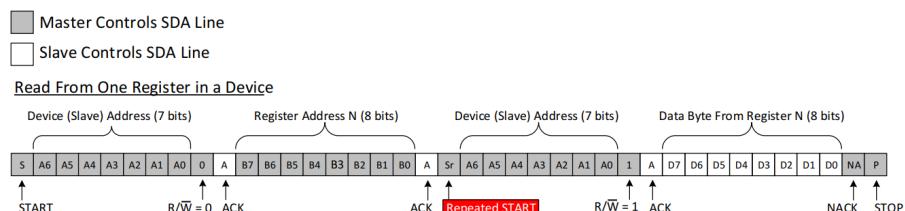


Figure 9. Example I²C Read from Slave Device's Register

Abbildung 5.1.2: I²C-Read (Understanding the I²C Bus⁸⁰ von Texas Instruments)

„onRequest“ In Abbildung 5.1.3 ist zu sehen, dass nun sofort mit der Verarbeitung der Daten begonnen werden kann, da das benötigte Register in der vorhergegangenen „onReceive“Funktion ausgelesen wurde, und als globale Variable gespeichert wurde.

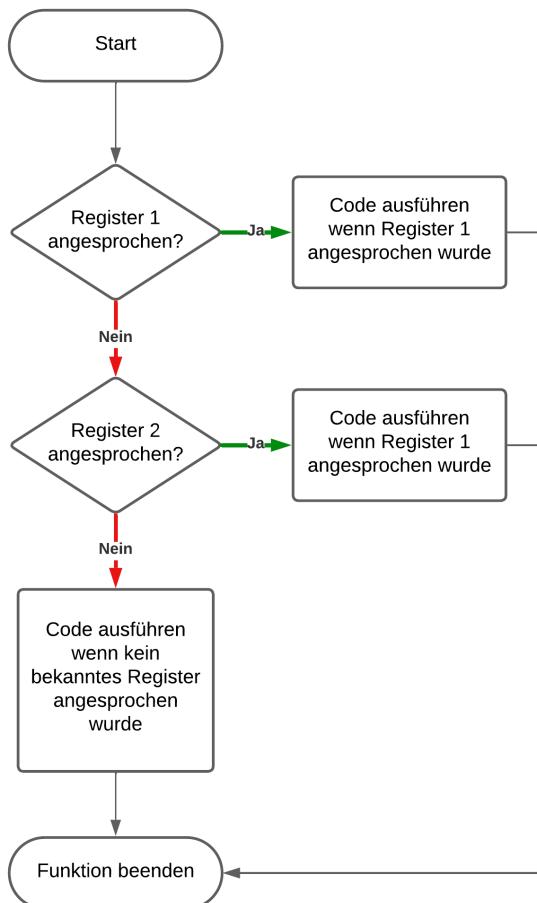


Abbildung 5.1.3: Flussdiagramm der „onRequest“Funktion

Neben diesen beiden Funktion wird auch noch eine „sendData“Funktion zur Verfügung gestellt, welche bis zu zwei Floats an den Master zurücksendet. Dabei werden auch Overloads für Integer und Char Datentypen zur Verfügung gestellt.

5.1.1 Verwendung von zwei I2C-Bussen

Auf dem Raspberry Pico sind zwei verschiedene I2C-Busse vorhanden. Diese funktionieren komplett unabhängig voneinander. Falls der Raspberry Pi also auf dem Bus kommuniziert, bleibt der andere Bus des Raspberry Picos frei, und kann von ihm für eine eventuelle Kommunikation mit seinen Sensoren und Aktoren verwendet werden. Beim Programmieren stehen die zwei Busse als verschiedene Objekte zur Verfügung: Wire und Wire1.

Für die Kommunikation mit dem Raspberry Pi wird der zweite Bus verwendet, um den ersten frei zu halten. Dies wurde so realisiert, weil einige Libraries so programmiert sind, dass sie immer den ersten Bus verwendet.

5.2 Wassersystem

Quellcode⁸¹

Basierend auf der Vorlage kann nur das Modul rundum das Wassersystem programmiert werden. Zur Steuerung des Wassersystems wird von der Vorlage nur die onReceive()-Funktion benötigt. Denn der Drucksensor soll nur intern Informationen übergeben. Sollte zu einem späteren Zeitpunkt allerdings entschieden werden, dass der Master doch den aktuellen Wasserdruck anfragen können soll, kann die Firmware leicht angepasst werden.

5.2.1 Setup der Pins

Zuerst muss definiert werden ob die verwendeten PINs (ADC, GPIO, I2C) Eingänge oder Ausgänge sind. Dabei wird für verschiedene Module unterschieden:

- Pumpe
- Schrittmotor
- Drucksensor
- I2C

⁸¹ Simon Koller. *InZ8ne/klimakammer-firmware-water-system*. original-date: 2024-02-19T15:27:24Z. 19. Feb. 2024. URL: <https://github.com/InZ8ne/klimakammer-firmware-water-system> (besucht am 05.04.2024).

In Abbildung 5.2.1 ist das Pinout des Raspberry Pi Picos zu sehen, damit es übersichtlicher wird, welche Pins wofür verwendet wird.

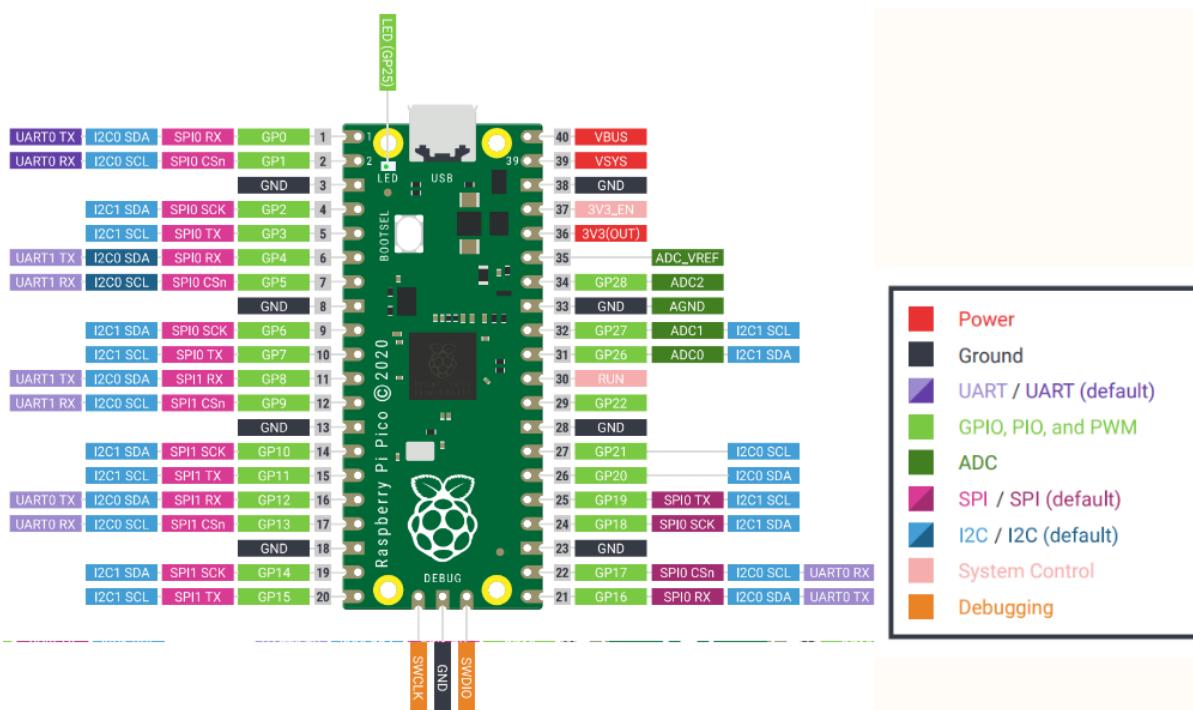


Abbildung 5.2.1: „Raspberry Pi Pico Pinout“

Da der Treiber-IC des Schrittmotors über I2C angesteuert wird, werden 2 I2C-Interfaces gebraucht, wozu der Pico im Stande ist. Für das Haupt-Interface also das extended I2C werden standardmäßig Pin 6 und Pin 7 verwendet. Für das zweite werden die Pins 10 und 11 verwendet.

Vier weitere Pins werden für die Pumpe benötigt wobei die GPIOs 19 und 21 für die Ausgabe der PWM-Signale zur Steuerung des DC-Motors verwendet werden. Dabei wird über GPIO 19 der Motor rückwärts betrieben und bei GPIO 21 respektive vorwärts. Der GPIO 18 wird zur Fehlererkennung verwendet und über den Pin ADC1 (GPIO 27) wird der interne Strom im Treiber-IC der Pumpe gemessen.

Der ADC0-Pin (GPIO 26) ist jener Pin an welchem der Wasserdruck gemessen wird und GPIO 17 wird zur Fehlererkennung beim Schrittmotor verwendet.

Diese Pins werden in der setup-Funktion der Firmware je nach Aufgabe als Eingang oder Ausgang initialisiert. In Abbildung 5.2.1 wird diese setup-Funktion gezeigt. Im Anschluss an die Initialisierung der Pins, wird noch festgelegt, dass die Frequenz der PWM-Signale 1kHz sein soll, und der Duty-Cycle des PWMs mit 8 Bit Variablen

einstellbar sein soll.

```
void setup() {  
    #ifdef DEBUG ...  
    #endif  
    Serial.begin(115200);  
  
    setupPump(); //Setup Raspi Pins for the Pump  
    setupPressureSensor(); //setup Raspi Pins for the pressure sensor  
    setupStepper(); // setup Raspi pins for the StepperController  
    setupWire1(); //setup Raspi pins for the second I2C Interface  
  
    analogWriteFreq(1000); // PWM Frequency 1kHz  
    analogWriteRange(255); //8bit PWM  
}
```

Listing 5.2.1: Firmware Wassersystem; Setup

5.2.2 Steuerung der Module

Wenn der Master einen neuen Befehl schickt, wird die onReceive()-Funtion der Vorlage ausgeführt. Darin wird dann je nach dem, welches Modul verändert werden soll, die Abarbeitung des Commands und die Aktualisierung des Moduls vorgenommen. Eine grobe Übersicht gibt Abbildung 5.2.2.

```

switch(module){
    case Pump:
        #ifdef DEBUG
        | Serial.println("Pump called");
        #endif
        //Code to execute when Pump is being called
        analogWrite(PWMPINForward, data); //Sets the speed of the DC Pump based on what the master wants
        break;
    case Stepper:
        #ifdef DEBUG
        | Serial.println("Stepper called");
        #endif
        //Code to execute when Stepper is being called
        direction = data>>7; //The MSB signifies the direction
        newAngle = data & 0b01111111; //The other 7 bit are the desired new angle
        steps = (int)(currentAngle-newAngle)/HalfStep; //calculate how many steps are needed
        currentAngle = newAngle; // Update the current Angle of the Stepper
        transmitToStepperController(steps, direction); //Move the stepper to the new Position
        break;
    case Sensor:
        #ifdef DEBUG
        | Serial.println("Sensor called");
        #endif
        //Code to execute when Sensor is being called
        waterPressure = getWaterPressure(); //get the value of water pressure
        // Code if pressure too high not defined yet
        break;
}

```

Listing 5.2.2: Steuerung der einzelnen Module

Steuerung der Pumpe Soll die Geschwindigkeit der Pumpe umgestellt werden, wird jediglich der neue 8bit Wert für den Duty-Cycle des PWMs im Command übergeben und das PWM-Signal, welches zum Treiber-IC DRV8874 geht, wird auf diesen angepasst.

Steuerung des Schrittmotors Die Steuerung des Schrittmotors ist etwas komplizierter. Das MSB im Command gibt die Richtung vor in welche sich der Schrittmotor drehen soll, also Ventilstand erhöhen oder verkleinern. Über die restlichen 7 Bit wird der neue Ventilstand übergeben (0-90°). Das funktioniert, weil 7 Bit genug sind um 128 verschiedene Werte zu bilden: $2^7 = 128$. Mit diesem neuen Ventilstand wird die Winkel-Änderung in Grad berechnet und durch den Winkel dividiert, der durch einen Halb-Schritt zurückgelegt wird. Damit wird die Anzahl an benötigten Schritten berechnet. Die Funktion „transmitToStepperController()“ stellt den Schrittmotor mit Hilfe des I2C1-Interfaces auf die neue Position. Um den Stellwinkel des Schrittmotors über I2C zu verändern muss das IC1 Kontrollregister des DRV8847S überschrieben werden. Die einzelnen Bits des Registers sind in Abbildung 5.2.2 ablesbar.

Table 7-18. IC1 Control Register Field Descriptions

Bit	Field	Type	Reset	Description
7	TRQ	R/W	0b	0b = Torque scalar set to 100% 1b = Torque scalar set to 50%
6	IN4	R/W	0b	The INx bits are used to control the bridge operation.
5	IN3	R/W	0b	The INx bits are used to control the bridge operation.
4	IN2	R/W	0b	The INx bits are used to control the bridge operation.
3	IN1	R/W	0b	The INx bits are used to control the bridge operation.
2	I2CBC	R/W	0b	0b = Bridge control configured by using the INx pins 1b = Bridge control configured by using the INx bits
1-0	MODE	R/W	00b	00b = 4-pin interface 01b = 2-pin interface 10b = Parallel interface 11b = Independent mode

Abbildung 5.2.2: Auszug aus dem DB des DRV8847; Seite 47; Kapitel 7.6.2;
Tabelle 7-18

Über das *TRQ* Bit lässt sich das Drehmoment halbieren, das ist für diese Anwendung irrelevant. Über die IN-Bits werden die einzelnen Eingänge der H-Brücken gesetzt. Mit dem Bit *I2CBC* lässt sich konfigurieren ob die genannten Eingänge über die I2C-Bits oder die Pins des ICs gesteuert werden sollen. Da in diesem Fall die Pins nicht angeschlossen ist muss dieses Bit also immer gesetzt sein. Durch die 2 *MODE* Bits lässt sich die Betriebsart einstellen. Diese müssen immer beide auf 0 gesetzt werden, damit der IC im 2-Phasen-Modus bleibt.

Damit sich der Schrittmotor aus seiner Position löst und in eine andere versetzt werden kann müssen die H-Brücken Eingänge in einer designierten Reihenfolge gesetzt und gelöscht werden. Die Reihenfolge für Halbschritte ist im DB des DRV8847 festgelegt (Abbildung 5.2.3).

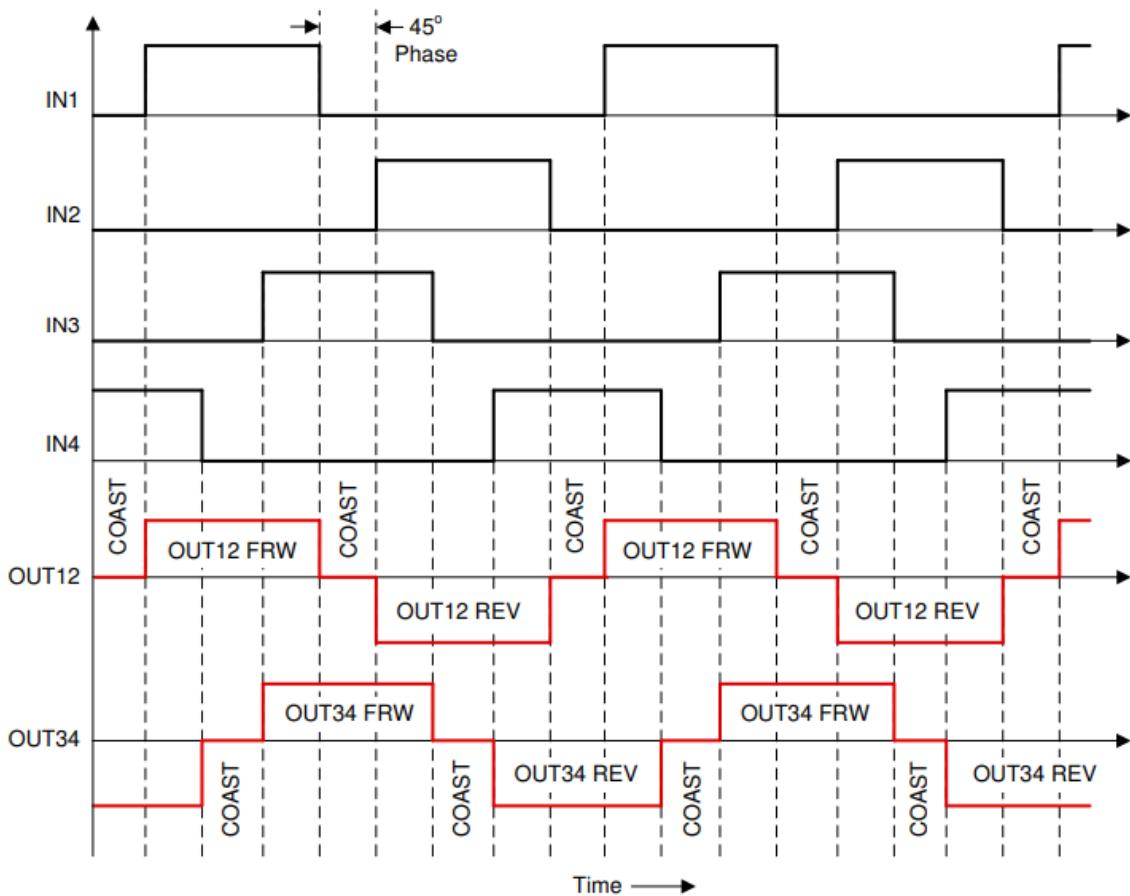


Abbildung 5.2.3: Auszug aus dem DB des DRV8847; Seite 33; Kapitel 7.3.7.3;
Figure 7-16

Steuerung der Sensoren Wenn der Drucksensor aufgerufen wird, wird der aktuelle Wasserdruck über den ADC eingelesen. Die Funktion `getWaterPressure()` liest den ADC-Wert und berechnet den Wasserdruck in PSI. Zum jetzigen Zeitpunkt wird mit diesem Wert noch nichts angefangen, aber bei Fertigstellung des Dokuments sollte, die Pumpe bei zu hohem Druck heruntergefahren und das Ventil zur Gänze geöffnet werden, damit sich der Druck abbauen kann.

5.3 Sensoren

Quellcode⁸²

Da beide von uns gekauften Sensoren auch von Adafruit vertrieben werden, gibt es

⁸² Joel05. *Joel05/klimakammer-firmware-sensor*. original-date: 2024-02-25T15:01:31Z. 25. Feb. 2024. URL: <https://github.com/Joe105/klimakammer-firmware-sensor> (besucht am 05.04.2024).

bereits Libraries im Arduino Framework, welche den Programmieraufwand für uns sehr gering halten. Das wichtigste hierbei ist die Konfiguration der Sensoren.

5.3.1 BME280

Der BME280 kann in drei verschiedenen Modi verwendet werden:

- **Sleep mode:** Hierbei sind alle Register des Sensors verfügbar, er führt aber keine Messungen durch. Kann zum stromsparen verwendet werden.
- **Forced mode:** Misst einmal und speichert die gelesenen Werte. Kehrt danach wieder in den Sleep mode zurück.
- **Normal mode:** Hierbei wird zyklisch gemessen und immer der aktuellste Wert gespeichert.

Von uns wurde der „Normal-mode“ verwendet, da der Stromverbrauch keine Rolle spielt, aber nicht auf die Daten gewartet werden kann.

Dann kann noch eingestellt werden, wievielfach der Sensor oversampeln soll. Dies bedeutet, wie viel Messungen der Sensor mitteln soll. Dabei kann ein Wert zwischen 1 und 16 eingestellt werden. Da wir die höchste Genauigkeit erreichen wollen, und nicht auf Schnelligkeit angewiesen sind, werden alle 3 Parameter mit 16-fachem Oversampling gemessen.

Außerdem ist auf dem Sensor noch ein IIR-Filter implementiert, welches kurzfristige Störung (Pusten auf den Sensor) ausgleichen soll. Dabei kann die Ordnung zwischen 0 (ausgeschalten) und 16 ausgewählt werden. Auch hier wurde die höchste Ordnung gewählt, um die höchste Genauigkeit zu erreichen.

Der Parameter $t_{standby}$ kann auch noch eingestellt werden. Dieser gibt an, wie oft der Sensor im „Normal-mode“ messen soll. Dieser wurde bei uns auf den niedrigsten Wert mit 0.5ms gestellt, um immer möglichst aktuelle Daten zu haben.

Diese Konfiguration geschieht über eine Funktion, wie in Abbildung 5.3.1 zu sehen ist.

```
bme.setSampling(Adafruit_BME280::MODE_NORMAL,
                  Adafruit_BME280::SAMPLING_X16, // temperature
                  Adafruit_BME280::SAMPLING_X16, // pressure
                  Adafruit_BME280::SAMPLING_X16, // humidity
                  Adafruit_BME280::FILTER_X16,
                  Adafruit_BME280::STANDBY_MS_0_5 );
```

Listing 5.3.1: Konfiguration des BME280

Dann können über die Funktionen `bme.readHumidity()`, `bme.readTemperature()` und `bme.readPressure()` die jeweiligen Werte ausgelesen werden.

5.3.2 LTR-390

Auch der LTR-390 kann über Funktionen konfiguriert werden. Dabei stehen aber weitaus weniger Einstellungen zur Verfügung:

- **Auflösung des Sensors:** Die Auflösung des Sensors kann zwischen 13 Bit und 20 Bit eingestellt werden. Dabei hängt die Zeit die für eine Messung benötigt wird, direkt von der Auflösung ab. Beim 20 Bit benötigt der Sensor 400ms für eine Messung, wohingegen er für 13 Bit nur 12.5ms benötigt.
- **Verstärkung des Sensors:** Die auf dem Sensor eingebaute Verstärkung des gemessenen Signals kann zwischen 1 und 18 eingestellt werden.

Da wir die höchste Genauigkeit wollten, haben wir eine Auflösung von 20 Bit gewählt. Die Verstärkung beließen wir bei 3, da wir hier den Dynamikumfang des Sensors gut ausnutzen.

Ein Problem bei unserer geplanten Benutzung des Sensors ist, dass er immer nur in einem Modus arbeiten kann, entweder als ALS oder als UV-Sensor. Das Problem hierbei ist, dass wenn der Sensor z.B. im ALS-Modus arbeitet und der Raspberry Pi die Intensität des UV-Lichts will, müsste er eine ganze Umwandlung warten, da der Sensor diese Messung erst dann starten kann. Es gibt zwar kein Limit beim I2C-Clockstretching, denn Bus 400ms lang zu blockieren ist aber keine gute Idee.

Die Lösung hierfür ist, dass wir den Sensor immer zwischen ALS- und UV-Mode togglen, und den gemessenen Wert auf dem Raspberry Pico speichern. Damit sind immer Werte für den Raspberry Pi verfügbar. Dafür wird die in Abbildung 5.3.2 gezeigte Funktion alle 500ms aufgerufen.

```
void readBrightness(){
    if(switchState){
        BrightnessValue = (ltr.readALS()*0.6)/(3*4);
        ltr.setMode(LTR390_MODE_UVS);
    }
    else{
        UVBrightnessValue = ltr.readUVS();
        ltr.setMode(LTR390_MODE_ALS);
    }
    switchState = !switchState;
    return;
}
```

Listing 5.3.2: Funktion um zwischen UV- und ALS-Modus zu rotieren

5.4 Sonnensteuerung

Auf dem Raspberry Pico wird ein ganz einfacher Code ausgeführt [Abbildung 5.4.1]. Sobald Daten vom Master an den Slave gesendet werden, wird durch die Wire Library eine Funktion namens „onRecieve“ ausgeführt. die Wire Library ermöglicht einfache Bedienung der IIC Schnittstelle am Raspberry Pi. in der onReceive() Funktion, werden die Daten mit Wire.read() ausgelesen, und anschließend zum nächsten Faktor von 20 gerundet. Dies ist unumgänglich, da unsere Lampen nur in 20er Schritten gesteuert werden können. Dieser gerundete Wert, wird in die setMUX Pins() Funktion weiter gegeben. Diese setzt je nach dem, die GPIO Pins 18,19 und 21 so, dass der Multiplexer auf der Steuerplatine den richtigen Widerstand dazuschaltet.

Wenn man nun die gesamte Steuerung als Blockschaltbild zeigt bekommt man eine Abbildung wie in 5.4.1

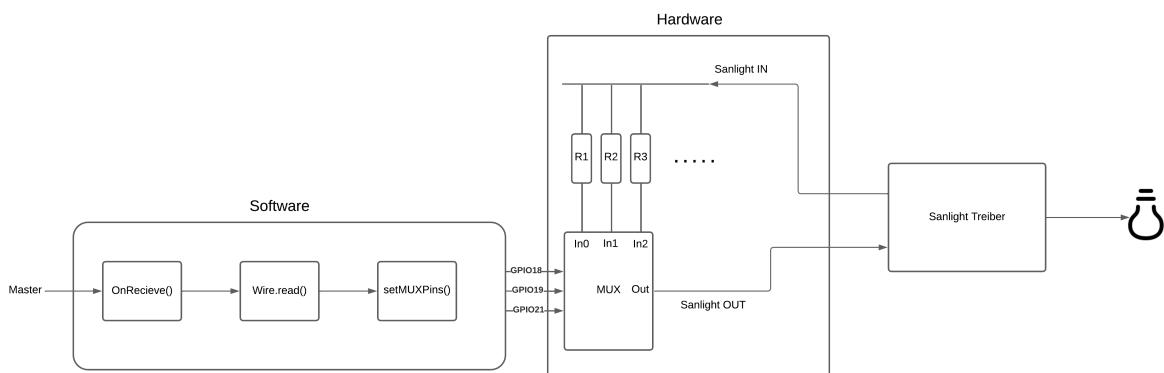


Abbildung 5.4.1: Blockschaltbild der gesamten Sonnensteuerung

5.5 Firmware für die Lüftersteuerung

Quellcode⁸³

Um die Windgeschwindigkeit innerhalb der Kammer präzise regeln zu können, wurde eine Art digitaler Regelkreis aufgebaut. Die Funktionsweise dieses Regelkreises ist simpel, und wird in Abbildung 5.5.1 gezeigt.

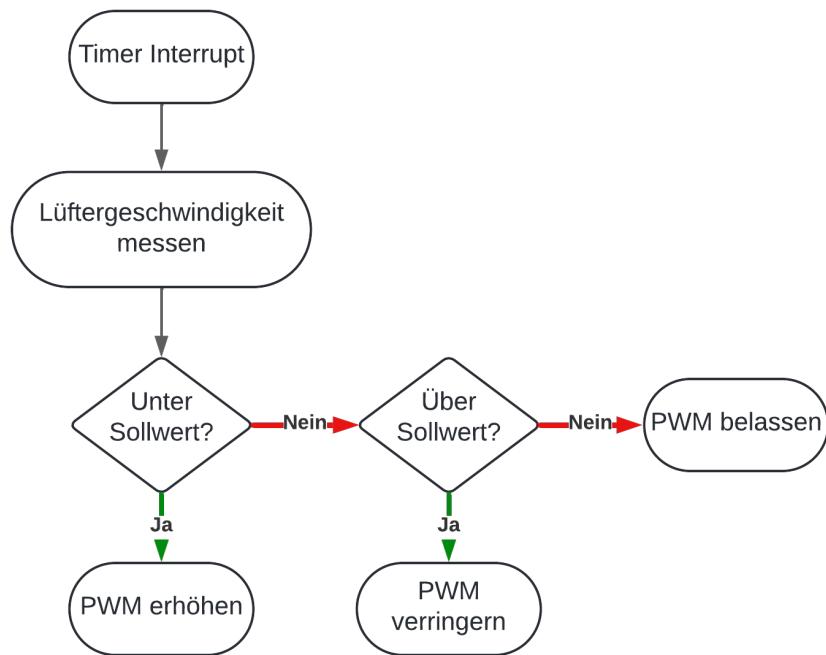


Abbildung 5.5.1: Ablaufdiagramm Lüfterregelung

Die Messung der Geschwindigkeit erfolgt über eine Messung der Zeit zwischen den Flanken des Tacho-Signals. Eine Flanke erfolgt jede halbe Umdrehung des Lüfters. Mit folgender Formel kann die Zeit zwischen den Flanken dann in Umdrehungen pro Minute umgewandelt werden.

$$fan_speed [rpm] = \frac{\frac{1}{Time[s]} * 60}{2}$$

Dieser Regler ist aber sehr langsam. Daher wird, wenn ein neuer Sollwert von Raspber-

⁸³ [JoeL05.JoeL05/klimakammer-firmware-fan](https://github.com/JoeL05/klimakammer-firmware-fan). original-date: 2024-02-19T14:59:28Z. 19. Feb. 2024.
URL: <https://github.com/JoeL05/klimakammer-firmware-fan> (besucht am 05.04.2024).

ry Pi gesetzt wird, mit folgender Formel ein ungefährer PWM-Sollwert berechnet:

$$PWM = \frac{Sollwert [rpm]}{2000}$$

Die Differenz zum Sollwert wird dann vom langsamen Regler ausgeglichen. Es wurde somit ein Regler von Typ PI-Regler gebaut.

6 Projektmanagement

6.1 Verwendete Software

Am Anfang der Projektplanung stand die Grundsatzfrage, welche Software dafür verwendet werden sollte. Dabei stand die Wahl zwischen „allgemeiner“ Software wie z.B. Excel, oder speziell für Projektmanagement entwickelten Softwares wie Monday.com⁸⁴, Asana⁸⁵ und Hive.com⁸⁶ (im folgenden nur als „Hive“ bezeichnet).

Angefangen hatten wir mit Asana, haben aber schnell bemerkt dass essentielle Features wie der Gantt-Chart und ein Time-Tracking darin fehlten, bzw. nicht in der Gratisversion enthalten sind. Die Wahl lag dann zwischen Monday.com und Hive.com. Da bei Monday.com die Teamgröße in der Gratisversion auf 2 Mitglieder begrenzt ist, sind wir dann auf Hive umgestiegen, wobei das komplette bereits erstellte Projekt darin übernommen werden konnte.

Meistens wurde in Hive die „Teamview“, zu sehen in Abbildung 6.1.1, verwendet. In dieser sind alle Tasks den jeweiligen Zuständigen zugewiesen. Dabei ist für jeden Task das geplante Enddatum, die Priorität, die Anzahl der Unteraufgaben (jedoch nur bis zur ersten Ebene) und die bereits investierten Stunden ersichtlich.

⁸⁴ *monday.com | A new way of working.* URL: <https://monday.com/> (besucht am 23.12.2023).

⁸⁵ *Asana. Verwalten Sie die Arbeit, Projekte und Aufgaben Ihres Teams online • Asana.* URL: <https://asana.com/de> (besucht am 23.12.2023).

⁸⁶ *Manage Projects Faster & Collaborate Better | Hive.* 26. Mai 2022. URL: <https://hive.com/> (besucht am 23.12.2023).

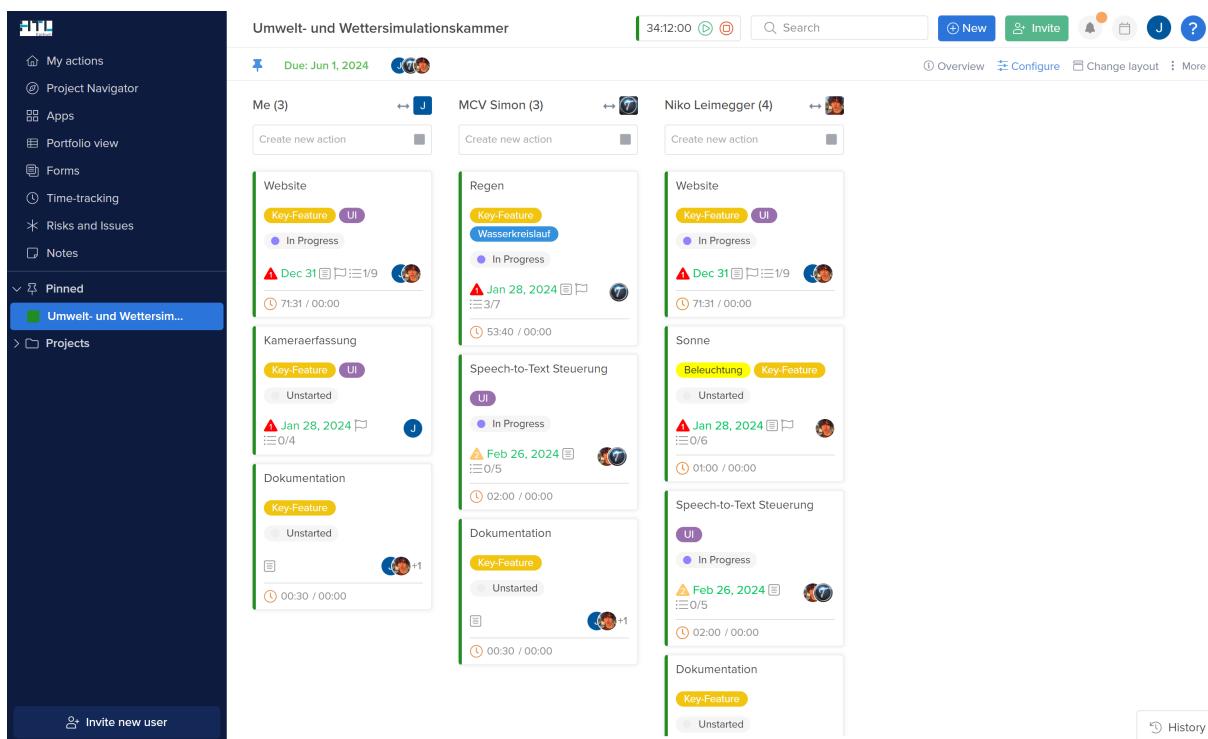


Abbildung 6.1.1: „Teamview in Hive“

6.2 Tracking der Stunden

In Hive können die bereits aufgewendeten Stunden zwar leicht eingetragen werden, jedoch gibt es keine befriedigende Möglichkeit zur Auswertung dieser. Die einzige Möglichkeit ist ein Export als CSV-Datei. Da es aber eine API gibt, wurde das Problem selber in die Hand genommen, und mit Grafana und der in Kapitel 4.2.1 beschriebenen API eine Auswertung selber programmiert. Grafana⁸⁷ ist ein Framework zur Visualisierung von Daten. Da dies nichts mit dem eigentlichen Projekt zu tun hat, wird auf diesen Code nicht weiter eingegangen. Das Ergebniss in Grafana ist in Abbildung 6.2.1 zu sehen.

⁸⁷ Grafana: The open observability platform. Grafana Labs. URL: <https://grafana.com/> (besucht am 19.03.2024).



Abbildung 6.2.1: Erfassung der gesamt getrackten Stunden in Grafana

6.3 Dokumentation der Zwischenergebnisse

Zur Dokumentation verschiedener Sachen, wie z.B. des aktuellen Stands, für Notizen oder als To-Do-Liste wurde Notion⁸⁸ verwendet. Bei Notion handelt es sich um eine sehr umfangreiche Anwendung, in welcher Notizen niedergeschrieben werden können, und in Echtzeit mit anderen geteilt werden können. Dabei können die verschiedenen Seiten der Notizen aufeinander aufbauen, was eine übersichtliche Struktur ermöglicht. Unsere „Landingpage“ in Notion ist in Abbildung 6.3.1 zu sehen.

⁸⁸ Notion. Your connected workspace for wiki, docs & projects. Notion. URL: <https://www.notion.so/product> (besucht am 18.03.2024).

Diplomarbeit

 Einzelne Module

 Wöchentliche Updates

 Bauteilliste

 Misc

 Meetings

 AI - Speech-to-Text Steuerung

 LaTeX Gliederung

Abgabe

1d to go

Abbildung 6.3.1: Startseite unseres Workspaces in Notion

7 Ausblick

Da von Grund auf festgelegt wurde, dass das gesamte System modular und leicht erweiterbar sein soll folgt ein Ausblick, welche Ziele noch denkbar wären, sollte sich jemand dazu entscheiden, an unserem Projekt weiter zu forschen.

Um das Erdenklima noch besser und akkurater darzustellen könnten noch weitere Module entworfen werden. Dazu zählen:

- **Temperaturregelung**

Die Temperatur ist ein wesentlicher Faktor wenn es um das Klima geht. Könnte man die Temperatur im Inneren der Kammer regeln, würden Endnutzer noch bessere Aussagen über die Auswirkung von klimatischen Bedingungen auf ihre Prüflinge tätigen können. Ein Ansatz für die Implementierung einer Temperaturregelung wäre zum Beispiel das Verwenden von Peltier-Elementen. Diese sind in der Lage die Umgebungstemperatur schnell und in einem großen Bereich zu verändern. Großer Nachteil von ihnen ist allerdings ihre sehr geringe Effizienz von meist kleiner als 30%.

- **Luftbe-/entfeuchtung**

Die Luftfeuchtigkeit wird bereits ansatzweise durch das Regen-Modul verändert. Das allerdings weder gesondert steuerbar noch genau. Dazu könnten wieder Peltier-Elemente verwendet werden, da eine Erhitzung bzw. Abkühlung von Luft die Luftfeuchte verändert. Allerdings ist es dafür, ebenso wie für die Temperaturregelung, notwendig, dass die Luft durch diese Elemente zirkuliert.

- **Tag/Nacht Zyklus**

Der Tag/Nacht Zyklus hat große Auswirkungen auf Pflanzen. Falls also pflanzliche Prüflinge angestrebt werden, wäre es von Vorteil diesen in das bestehende Sonnensystem zu implementieren. Die bereits bestehende Sonnesteuerung bietet bereits einen die Möglichkeit Tag und Nacht zu simulieren allerdings ohne Einwirkung von UV. Außerdem wäre das stufenlose Einstellen ein mögliches weiteres Ziel.

8 Abbildungsverzeichnis

1.1.1	Blockschaltbild des Gesamtsystems	15
2.1.1	Grundgerüst in MTpro64	19
2.1.2	Grundgerüst mit Mann mittlerer Größe	19
2.1.3	Fertiges Gehäuse in Solidworks	21
2.1.4	Innenkammer in Solidworks	23
2.1.5	Funktionsweise Gesenkbiegen	24
2.1.6	Kollision des Werkstückes mit der Gesenkbiegenmaschine	25
2.2.1	Anwendungen des Schlauch-Regners	27
2.2.2	Schlauch-Regner in Aktion	27
2.2.3	Perlschlauch	28
2.2.4	Hochdruck-Wasserpumpe der Marke Ejoyous	30
2.2.5	Filter der Marke LATRAT	31
2.2.6	10m Comfort Flex Schlauch 1/2"(13mm)	32
2.2.7	Gardena Schlauchkupplung mit Regulierventil	33
2.2.8	Gardena Schlauchverbinder für 1/2"(13mm) Schläuche	34
2.2.9	Gardena Hahnverbinder für G1/2"Gewinde	34
2.2.10	Blockschaltbild des Wasserkreises	36
3.1.1	Auszug aus „HPE Common Slot Power Supplies Quickspecs“	38
3.1.2	Pinout der T -Revision	39
3.1.3	Pinout der P -Revision	39
3.1.4	Anschlussdiagramm des Powersupply	40
3.1.5	12VHPWR-Stecker von Amphenol	42
3.1.6	XT60-Stecker	43
3.1.7	Aufgaben der Platine	44
3.1.8	Fertige Platine im Altium 3D-Viewer	44

3.1.9	Stromdichte der Platine laut Keysight-PDA (Legende im Bild)	46
3.1.10	Stromdichte der Platine an einer Stelle	47
3.1.11	Halterung der Spannungsversorgung	48
3.2.1	Blockdiagramm des Systems	49
3.2.2	Raspberry Pi 4 Model B inkl. Powersupply aufsatz	51
3.2.3	NVIDIA Jetson Nano DevKit	52
3.2.4	„Figure 4. Minimum Subsystem Diagram“ aus dem Datenblatt	52
3.2.5	Luftzirkulation im Gehäuse	53
3.2.6	CPU-Temperatur bei Vollauslastung bei unterschiedlichen Lüfterkonfigurationen	54
3.3.1	Raspberry Pico	56
3.3.2	Blockschaltbild der Steuerplatine	57
3.3.3	Fertige Steuerplatine mit Raspberry Pico	57
3.4.1	Drucksensor von Amazon	58
3.4.2	„Schaltung einer H-Brücke“	59
3.4.3	„gewählter Schrittmotor“	61
3.4.4	„Eckdaten des Schrittmotors“	61
3.4.5	„Schaltplan der Platine zur Steuerung des Wassersystems“	63
3.4.6	„Benötigte Anschlüsse am Pinheader“	64
3.4.7	„Spannungsregulierung über den MC7805“	65
3.4.8	„Auszug aus dem DB der MC78xx-Familie; Seite 1; Maximum Ratings“	65
3.4.9	„Auszug aus dem DB des DRV8874; Seite 9, Kapitel 7.2“	66
3.4.10	„Auszug aus dem DB des DRV8874; Seite 10, Kapitel 7.3.2, Tabelle 2“	67
3.4.11	„Auszug aus dem DB des DRV8874; Seite 3, Kapitel 5, Pin-Functions“	68
3.4.12	„Beschaltung des DRV8874“	68
3.4.13	„Auszug aus dem DB des DRV8874; Seite 3; Kapitel 5; Pin Functions“	69

3.4.14 „Auszug aus dem DB des DRV8874; Seite 3; Kapitel 5; Pin Functions“	69
3.4.15 „Auszug aus dem DB des DRV8874; Seite 12; Kapitel 7.3.3.1; Current Sensing“	70
3.4.16 „Auszug aus dem DB des DRV8874; Seite 4; Kapitel 6.3; Recommended Operating Conditions“	70
3.4.17 Auszug aus dem DB des DRV8847; Seite 16; Kapitel 7.2; Figure 7-2	72
3.4.18 Beschaltung des Schrittmotors inklusive Treiber IC	73
3.4.19 Auszug aus dem DB des DRV8847; Seite 5; Kapitel 5; Tabelle 5.1	73
3.4.20 Auszug aus dem Db des DRV8847; Seite 5; Kapitel 5; Tabelle 5.1	74
3.4.21 Freilaufdioden für die Anschlüsse des Schrittmotors	74
3.4.22 Auszug aus dem DB des DRV8847; Seite 28; Kapitel 7.3.4; Formel 1	75
3.4.23 Auszug aus dem DB des DRV8847; Seite 9; Kapitel 6.5; Electrical Characteristics	75
3.4.24 „Beschaltung des Drucksensors“	76
3.4.25 „Schraubklemme mit 2 Anschlüssen von Amphenol“	76
3.4.26 „Board-Layout der Platine “	77
3.4.27 „3D-Modell der finalen Platine“	78
3.5.1 Lichtspektrum des Sonnenlichts ⁸⁹	79
3.5.2 Sanlight Spektrum	80
3.5.3 Testung des Steuerungsprinzips	81
3.5.4 Interne Schaltung des 74HC151N Multiplexers	82
3.5.5 Schaltplan für die Steuerung der Q1W DIM	83
3.5.6 Interne Funktionsweise CD405B	83
3.5.7 Neue Sonnensteuerung mittels Mosfets	83
3.6.1 Überwachungskamera	84

⁸⁹ *Lichtspektrum - Spektralfarben*. URL: <https://www.lichtmikroskop.net/optik/lichtspektrum.php> (besucht am 05.01.2024).

3.7.1	Sensorboard	85
3.8.1	NF F12 industrial PPC von Noctua	87
4.1.1	Library und Framework Logos (vue-router hat kein Logo)	93
4.1.2	„Startmenu der Website “	93
4.1.3	„Startmenu der Website befüllt “	94
4.1.4	„Bedienoberfläche um Ablauf zu erstellen “	95
4.1.5	„PopUp für Sonnenparameter“	95
4.1.6	„Manuelle Steuerung Übersicht“	96
4.1.7	„Live Wetter Übersicht“	99
4.1.8	„Wetter Vorhersage inkl. Graphen“	99
4.1.9	„Übersicht mit gefülltem Ablauf“	100
4.1.10	Einstellungs Übersicht	101
4.1.11	neues Feature Initialisieren	105
4.1.12	neues Feature wurde hinzugefügt	105
4.1.13	StatusUpdate anpassen	106
4.1.14	API Endpunkt definieren	106
4.1.15	Getter und Setter definieren	107
4.1.16	Im ChartStore die Charts definieren	108
4.1.17	UpdateCharts() anpassen	109
4.1.18	StatusChart() anpassen	109
4.1.19	Cleanweather() Funktion anpassen	110
4.2.1	Dokumentation der API	116
4.2.2	Testen der API	117
4.2.3	Systemparameter des AWS-Servers	118
4.2.4	Statusseite des Servers bei BetterStack.com	119
4.2.5	„Grobes Blockschaltbild der Funktion des Whisper-Modells (quelle) “	121

4.2.6 gesprochener Befehl: „Stelle die Sonne auf 10 Prozent“	126
4.2.7 gesprochener Befehl: „Stelle die Sonne in 10 Minuten auf 50 Prozent“	126
4.2.8 gesprochener Befehl: „Bilde das Wetter in Wien ab“	126
4.2.9 Sprachsteuerung Ergebnisse	126
5.1.1 Flussdiagramm der „onReceive“Funktion	130
5.1.2 I2C-Read (Understanding the I2C Bus ⁹⁰ von Texas Instruments) . . .	131
5.1.3 Flussdiagramm der „onRequest“Funktion	131
5.2.1 „Raspberry Pi Pico Pinout “	133
5.2.2 Auszug aus dem DB des DRV8847; Seite 47; Kapitel 7.6.2; Tabelle 7-18	136
5.2.3 Auszug aus dem DB des DRV8847; Seite 33; Kapitel 7.3.7.3; Figure 7-16	137
5.4.1 Blockschaltbild der gesamten Sonnensteuerung	140
5.5.1 Ablaufdiagramm Lüfterregelung	141
6.1.1 „Teamview in Hive “	144
6.2.1 Erfassung der gesamt getrackten Stunden in Grafana	145
6.3.1 Startseite unseres Workspaces in Notion	146

⁹⁰ Jonathan Valdez und Jared Becker. *Understanding the I2C Bus*. 2015. URL: https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1710828980808&ref_url=https%253A%252F%252Fwww.google.com%252F.

9 Tabellenverzeichnis

1.3.1 Meilensteine	17
------------------------------	----

10 Codeverzeichnis

4.1.1 Vue Struktur	90
4.1.2 Beispiel eines Reaktiven counters in Vue	91
4.1.3 Antwort von Openweathermap um die Koordinaten eines Ortes zu bestimmen	97
4.1.4 Json welches die Routine für das Backend beinhaltet	102
4.1.5 Einbindung eines Components in Vue	103
4.1.6 Definierung eines Pinia Stores	104
4.1.7 Pinia Store verwenden	104
4.1.8 LineChart im StatusBox.vue einfügen	110
4.2.1 Initialisierung von FastAPI	111
4.2.2 CORS in FastAPI hinzufügen	112
4.2.3 Endpunkt in FastAPI	112
4.2.4 Klassen für API-Objekte	113
4.2.5 Code um Daten von den Modulen zu lesen	114
4.2.6 Systemd Service	115
4.2.7 Konfiguration von NGINX	116
4.2.8 Funktion zur Ausführung einer Spracherkennung	123
4.2.9 Funktionen für die Filtrierung der Parameter	123
4.2.10 Erstellung eines Json-Objektes (Python-Dictionary)	124
5.2.1 Firmware Wassersystem; Setup	134
5.2.2 Steuerung der einzelnen Module	135
5.3.1 Konfiguration des BME280	138
5.3.2 Funktion um zwischen UV- und ALS-Modus zu rotieren	139

11 Quellenverzeichnis

020910. URL: <https://www.ikz.de/ikz-praxis-archiv/p0209/020910.php> (besucht am 04.04.2024).
- 3D-CAD-Konstruktionssoftware | SOLIDWORKS. URL: <https://www.solidworks.com/de/home-page-2021> (besucht am 17.03.2024).
- Advanced Load Balancer, Web Server, & Reverse Proxy. NGINX. URL: <https://www.nginx.com/> (besucht am 19.03.2024).
- Aplin, Richard. *raplin/DPS-1200FB*. original-date: 2017-03-27T21:25:20Z. 13. Dez. 2023. URL: <https://github.com/raplin/DPS-1200FB> (besucht am 03.01.2024).
- Asana. *Verwalten Sie die Arbeit, Projekte und Aufgaben Ihres Teams online* • Asana. URL: <https://asana.com/de> (besucht am 23.12.2023).
- Banana Pi BPI-M4 Berry with Allwinner H618, 2G RAM, 8G eMMC onboard -Banana Pi open source hardware community, Single board computer, Router, IoT, STEM education. URL: <https://www.banana-pi.org/en/banana-pi-sbcs/167.html> (besucht am 09.03.2024).
- Chart.js | Chart.js. URL: <https://www.chartjs.org/docs/latest/> (besucht am 23.12.2023).
- Colintd. *Hacking HP Common Slot Power supplies - 12V@100A*. 14. Okt. 2016. URL: <http://colintd.blogspot.com/2016/10/hacking-hp-common-slot-power-supplies.html> (besucht am 04.01.2024).
- Comfort Flex Schlauch 13 mm (1/2"), 10 m | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/schlaeuche/comfort-flex-schlauch-13-mm-12-10-m/967243601.html> (besucht am 04.04.2024).
- Debian Org. *Debian – The Universal Operating System*. URL: <https://www.debian.org/> (besucht am 06.01.2024).
- drv8847.pdf. URL: https://www.ti.com/lit/ds/symlink/drv8847.pdf?ts=1712239200629&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV8847 (besucht am 05.04.2024).
- drv8874.pdf. URL: https://www.ti.com/lit/ds/symlink/drv8874.pdf?ts=1708861609787&ref_url=https%253A%252F%252Fwww.google.com%252F (besucht am 04.04.2024).
- Edelstahl Drucksensor, G1/4"5V Ausgang 0,5-4,5V/0-5V Druckmessumformer Sensorseingang für Öl Brennstoff Gas Luft Wasser(0-150PSI) : Amazon.de: Auto & Motorrad. URL: https://www.amazon.de/dp/B07YKZ773Z?_encoding=UTF8&ref_=cm_sw_r_cp_ud_dp_AXZY3ANT7NM832QEC63E&th=1 (besucht am 05.04.2024).
- Einen Gleichstrommotor mit einer H-Brücke steuern | Rule The Bots. URL: <https://spacehal.github.io/docs/motor/hBridge> (besucht am 04.04.2024).
- ESP32 Wi-Fi & Bluetooth SoC | Espressif Systems. URL: <https://www.espressif.com/en/products/socs/esp32> (besucht am 19.03.2024).
- FastAPI. URL: <https://fastapi.tiangolo.com/> (besucht am 19.03.2024).
- Gesenkbiegepresse. In: Wikipedia. Page Version ID: 228053128. 17. Nov. 2022. URL: <https://de.wikipedia.org/w/index.php?title=Gesenkbiegepresse&oldid=228053128> (besucht am 09.03.2024).
- Grafana: The open observability platform. Grafana Labs. URL: <https://grafana.com/> (besucht am 19.03.2024).

GY-BME280 barometric sensor for temperature, air humidity and air pressure. AZ-Delivery. URL: <https://www.az-delivery.de/en/products/gy-bme280> (besucht am 04.04.2024).

Hahnverbinder | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/ogs/hahnverbinder/970521301.html> (besucht am 04.04.2024).

Halogenlampen. watt24 GmbH. URL: <https://www.watt24.com/leuchtmittel/halogenlampen/> (besucht am 19.03.2024).

Hewlett Packard Enterprise. HPE Common Slot Power Supplies. URL: https://support.hpe.com/hpsc/public/docDisplay?docLocale=en_US&docId=c04111541 (besucht am 04.01.2024).

Humidity Sensor BME280. Bosch Sensortec. URL: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/> (besucht am 04.04.2024).

III, Earle F. Philhower. *earlephilhower/arduino-pico.* original-date: 2021-02-25T04:20:27Z. 19. März 2024. URL: <https://github.com/earlephilhower/arduino-pico> (besucht am 19.03.2024).

ISO 20653 - 2023-08 - Beuth.de. URL: <https://www.beuth.de/de/norm/iso-20653/372581287> (besucht am 08.03.2024).

Joel05. *Joel05/klimakammer-firmware.* original-date: 2024-02-12T21:21:55Z. 13. Feb. 2024. URL: <https://github.com/Joel05/klimakammer-firmware> (besucht am 05.04.2024).

— *Joel05/klimakammer-firmware-fan.* original-date: 2024-02-19T14:59:28Z. 19. Feb. 2024. URL: <https://github.com/Joel05/klimakammer-firmware-fan> (besucht am 05.04.2024).

— *Joel05/klimakammer-firmware-sensor.* original-date: 2024-02-25T15:01:31Z. 25. Feb. 2024. URL: <https://github.com/Joel05/klimakammer-firmware-sensor> (besucht am 05.04.2024).

Kikomitk. *Tee6/Diplomarbeit_Klimakammer.* original-date: 2023-10-13T11:24:04Z. 25. Jan. 2024. URL: https://github.com/Tee6/Diplomarbeit_Klimakammer (besucht am 04.04.2024).

— *Tee6/improved-Klima-SST.* original-date: 2024-03-19T16:30:39Z. 19. März 2024. URL: <https://github.com/Tee6/improved-Klima-SST> (besucht am 04.04.2024).

— *Tee6/klimakammer-firmware-SUN.* original-date: 2024-02-14T09:03:58Z. 14. Feb. 2024. URL: <https://github.com/Tee6/klimakammer-firmware-SUN> (besucht am 04.04.2024).

Koller, Simon. *InZ8ne/klimakammer-firmware-water-system.* original-date: 2024-02-19T15:27:24Z. 19. Feb. 2024. URL: <https://github.com/InZ8ne/klimakammer-firmware-water-system> (besucht am 05.04.2024).

— *InZ8ne/Speech-to-Klima.* original-date: 2024-03-04T07:15:28Z. 4. März 2024. URL: <https://github.com/InZ8ne/Speech-to-Klima> (besucht am 05.04.2024).

Kupplung mit Regulierventil | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/ogs/kupplung-mit-regulierventil/970625801.html> (besucht am 04.04.2024).

LATRAT Hochdruckreiniger Wasserpumpe Saugfilter für Waschmaschine, 1/2SSaugschlauch Filter mit Doppelschicht Verschlüsselungssieb für Sprühgeräte Ölpumpen Industrie.

- trieausrustung : Amazon.de: Garten.* URL: https://www.amazon.de/dp/B097ZX981?ref_=cm_sw_r_cp_ud_dp_VCTSFEVRNS6TWY8WEB (besucht am 04.04.2024).
- Lichtspektrum - Spektralfarben.* URL: <https://www.lichtmikroskop.net/optik/lichtspektrum.php> (besucht am 05.01.2024).
- LM2576.* URL: <https://www.onsemi.com/products/power-management/dc-dc-power-conversion/converters/lm2576> (besucht am 09.03.2024).
- Ltd, Raspberry Pi. Pico.* Raspberry Pi. URL: <https://www.raspberrypi.com/products/raspberry-pi-pico/> (besucht am 19.03.2024).
- *RP2040.* Raspberry Pi. URL: <https://www.raspberrypi.com/products/rp2040/> (besucht am 19.03.2024).
- LTR-390UV_Final_DS_V1.1.pdf.* URL: https://optoelectronics.liteon.com/upload/download/DS86-2015-0004/LTR-390UV_Final_%20DS_V1%201.pdf (besucht am 04.04.2024).
- Lundell, Simon. *slundell/dps_charger.* original-date: 2019-08-12T09:06:03Z. 7. Nov. 2023. URL: https://github.com/slundell/dps_charger (besucht am 23.12.2023).
- Manage Projects Faster & Collaborate Better | Hive.* 26. Mai 2022. URL: <https://hive.com/> (besucht am 23.12.2023).
- MC7800_D-2315963.pdf.* URL: https://www.mouser.at/datasheet/2/308/1/MC7800_D-2315963.pdf (besucht am 04.04.2024).
- Membership Pricing.* IPC International, Inc. 12. Mai 2022. URL: <https://www.ipc.org/ipc-membership-pricing> (besucht am 09.03.2024).
- MIL-STD-810H.* URL: <https://www.iest.org/Standards-RPs/MIL-STD-810H> (besucht am 28.01.2024).
- Minitek® Pwr Hybrid 3.0 Header.* URL: <https://www.amphenol-cs.com/product/101609201240100lf.html> (besucht am 09.03.2024).
- monday.com | A new way of working.* URL: <https://monday.com/> (besucht am 23.12.2023).
- Morote, Eduardo San Morote. *Everything Beyond State Management in Stores with Pinia by Eduardo San Martin Morote - GitNation.* URL: <https://portal.gitnation.org/contents/everything-beyond-state-management-in-stores-with-pinia> (besucht am 23.12.2023).
- MTpro Planungssoftware.* Bosch Rexroth Deutschland. URL: <https://www.boschrexroth.com/de/de/produkte/produktgruppen/montagetechnik/themen/mtpro-planungssoftware/> (besucht am 09.03.2024).
- Murata Power Solutions. *d1u86p-w-1600-12-hbxdc.pdf.* URL: <https://www.murata.com/products/productdata/8807027081246/d1u86p-w-1600-12-hbxdc.pdf?1583754811000> (besucht am 04.01.2024).
- *d1u86t-w-800-12-hxxc.pdf.* 2018. URL: <https://www.murata.com/products/productdata/8807027245086/d1u86t-w-800-12-hxxc.pdf?1583754811000> (besucht am 03.01.2024).
- Noctua. *NF-F12 industrialPPC-2000 IP67 PWM.* URL: <https://noctua.at/de/nf-f12-industrialppc-2000-ip67-pwm> (besucht am 05.04.2024).

Noctua. *Noctua_PWM_specifications_white_paper.pdf*. URL: https://noctua.at/pub/media/wysiwyg/Noctua_PWM_specifications_white_paper.pdf (besucht am 05.04.2024).

Notion. *Your connected workspace for wiki, docs & projects*. Notion. URL: <https://www.notion.so/product> (besucht am 18.03.2024).

NVIDIA Corporation. *NVIDIA Jetson Nano für KI-Anwendungen in der Peripherie und Bildung*. NVIDIA. URL: <https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-nano/> (besucht am 06.01.2024).

openai/whisper. original-date: 2022-09-16T20:02:54Z. 6. März 2024. URL: <https://github.com/openai/whisper> (besucht am 06.03.2024).

OpenAPI Specification - Version 3.1.0 | Swagger. URL: <https://swagger.io/specification/> (besucht am 19.03.2024).

openweathermapwebsite. URL: <https://openweathermap.org/> (besucht am 12.03.2024).

Orange Pi - OrangePi. URL: <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5-Pro.html> (besucht am 09.03.2024).

PCB Trace Width Calculator and Equations. URL: <https://www.smpe.us/pcb-calculator.html> (besucht am 09.03.2024).

Perl-Regner | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/regner/perl-regner/900969401.html> (besucht am 04.04.2024).

picosdk. URL: https://www.raspberrypi.com/documentation/pico-sdk/index_doxxygen.html (besucht am 19.03.2024).

PMBus. Specification Archives | PMBus. 9. Juni 2010. URL: <https://pmbus.org/specification-archives/> (besucht am 04.01.2024).

Raspberry Pi Ltd. Buy a Raspberry Pi 4 Model B. Raspberry Pi. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (besucht am 06.01.2024).

— Operating system images. Raspberry Pi. URL: <https://www.raspberrypi.com/software/operating-systems/> (besucht am 06.01.2024).

Schlauch-Regner | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/regner/schlauch-regner/900883801.html> (besucht am 04.04.2024).

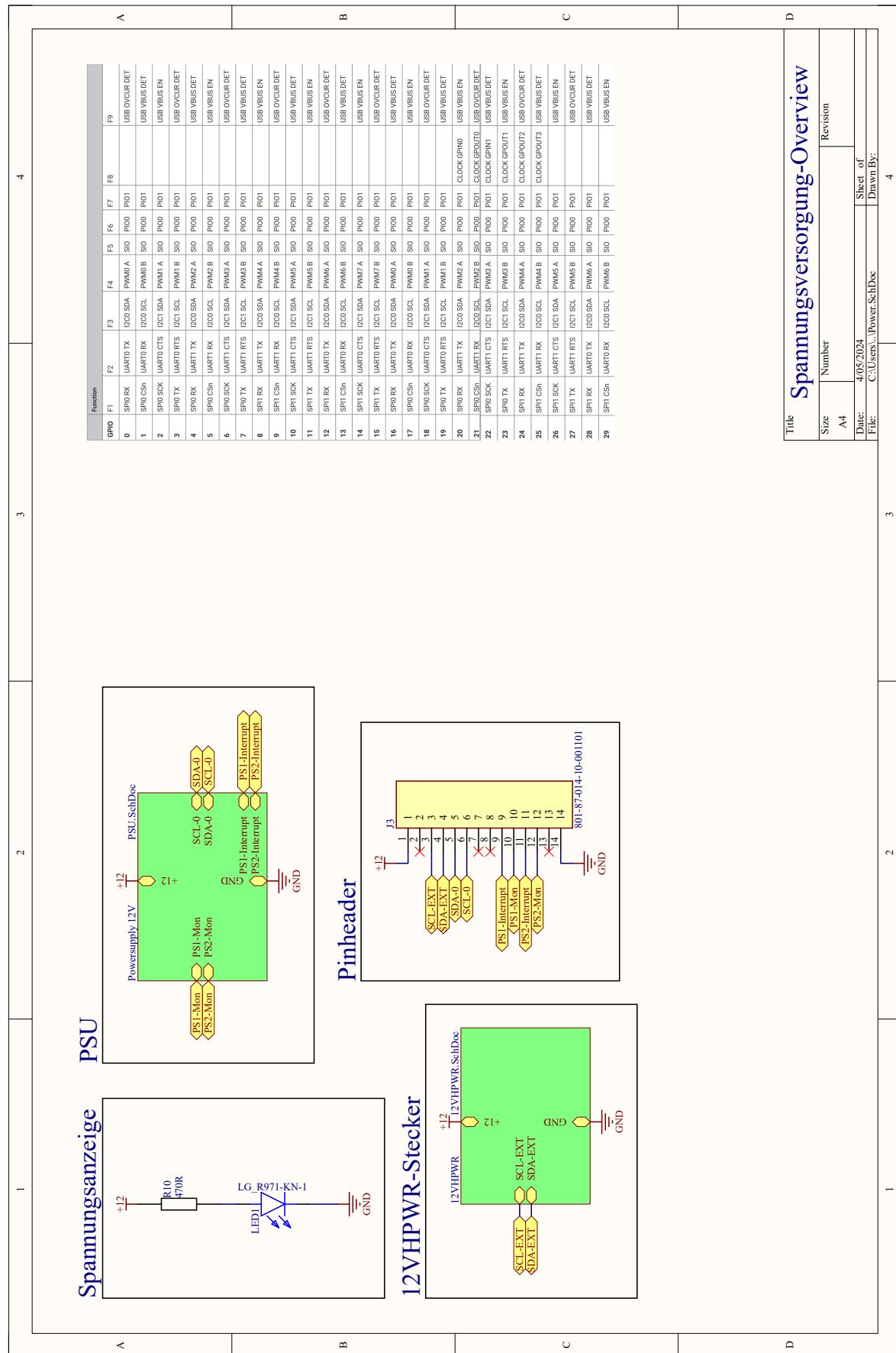
Schlauchverbinder 13 mm (1/2") - 15 mm (5/8") | GARDENA. URL: <https://www.gardena.com/at/produkte/bewaesserung/ogs/schlauchverbinder-13-mm-12---15-mm-58/967314001.html> (besucht am 04.04.2024).

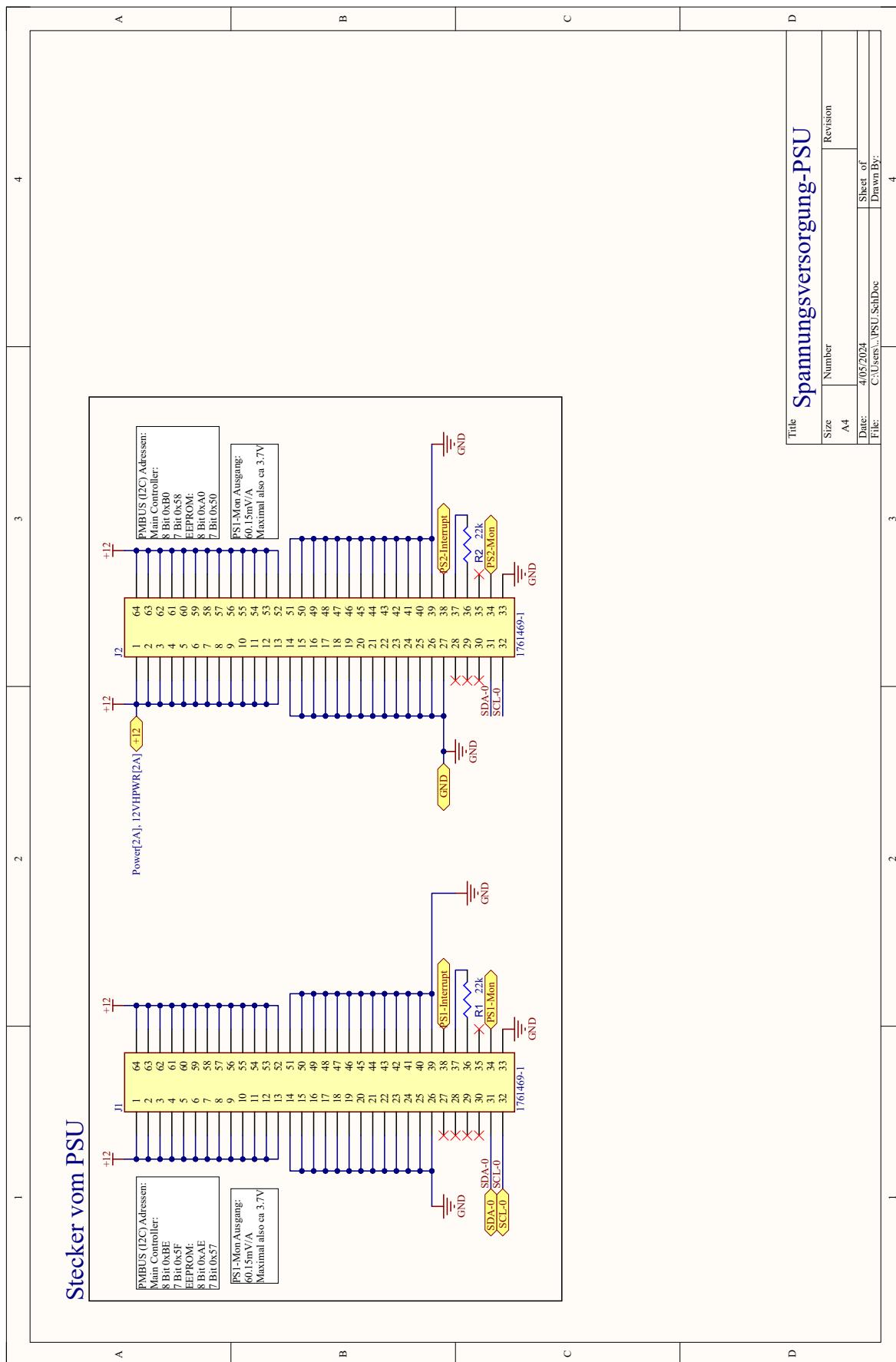
Schwenkbiegen. In: Wikipedia. Page Version ID: 238327141. 20. Okt. 2023. URL: <https://de.wikipedia.org/w/index.php?title=Schwenkbiegen&oldid=238327141> (besucht am 09.03.2024).

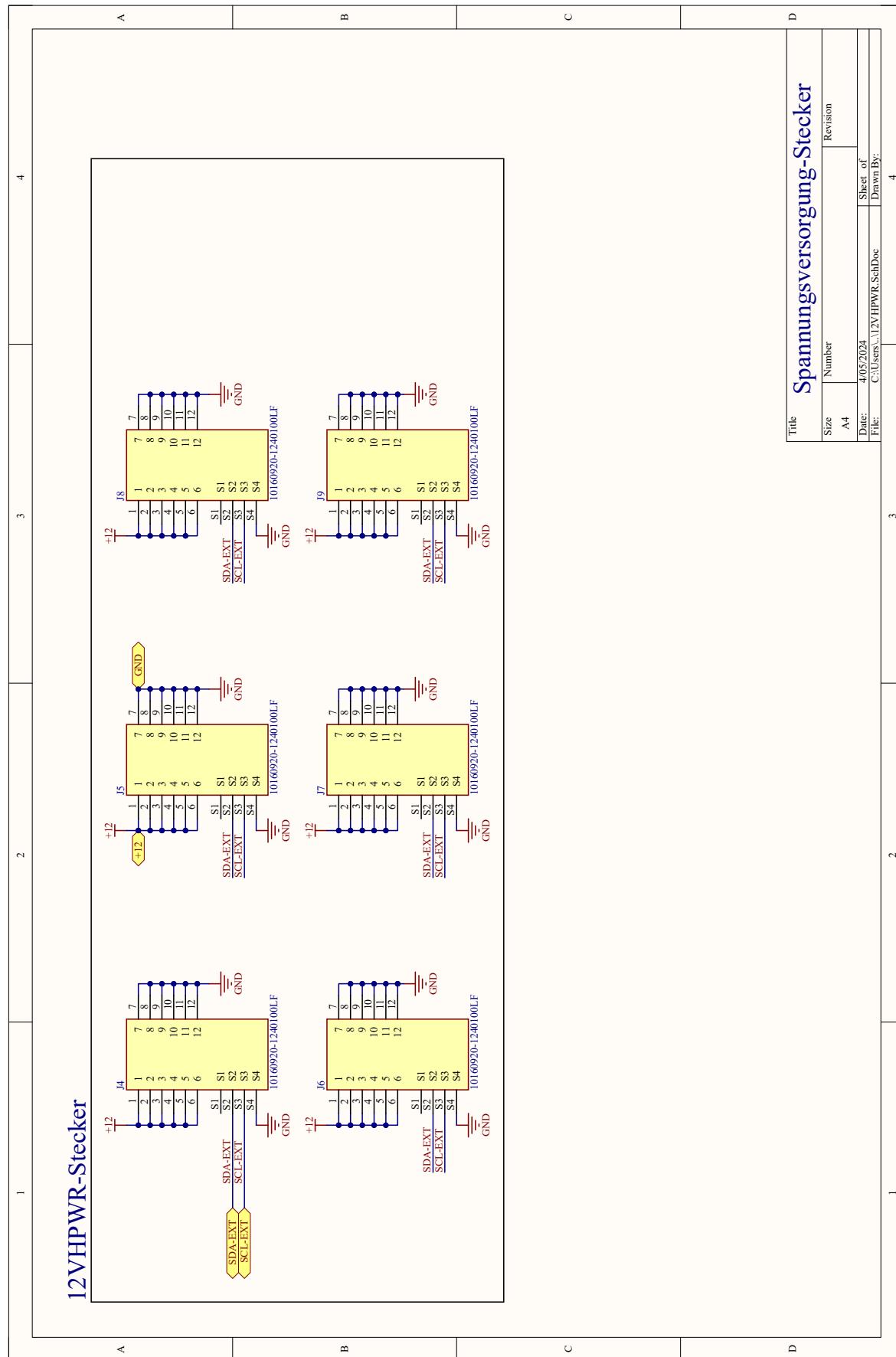
SYSTRAN/faster-whisper. original-date: 2023-02-11T09:17:27Z. 6. März 2024. URL: <https://github.com/SYSTRAN/faster-whisper> (besucht am 06.03.2024).

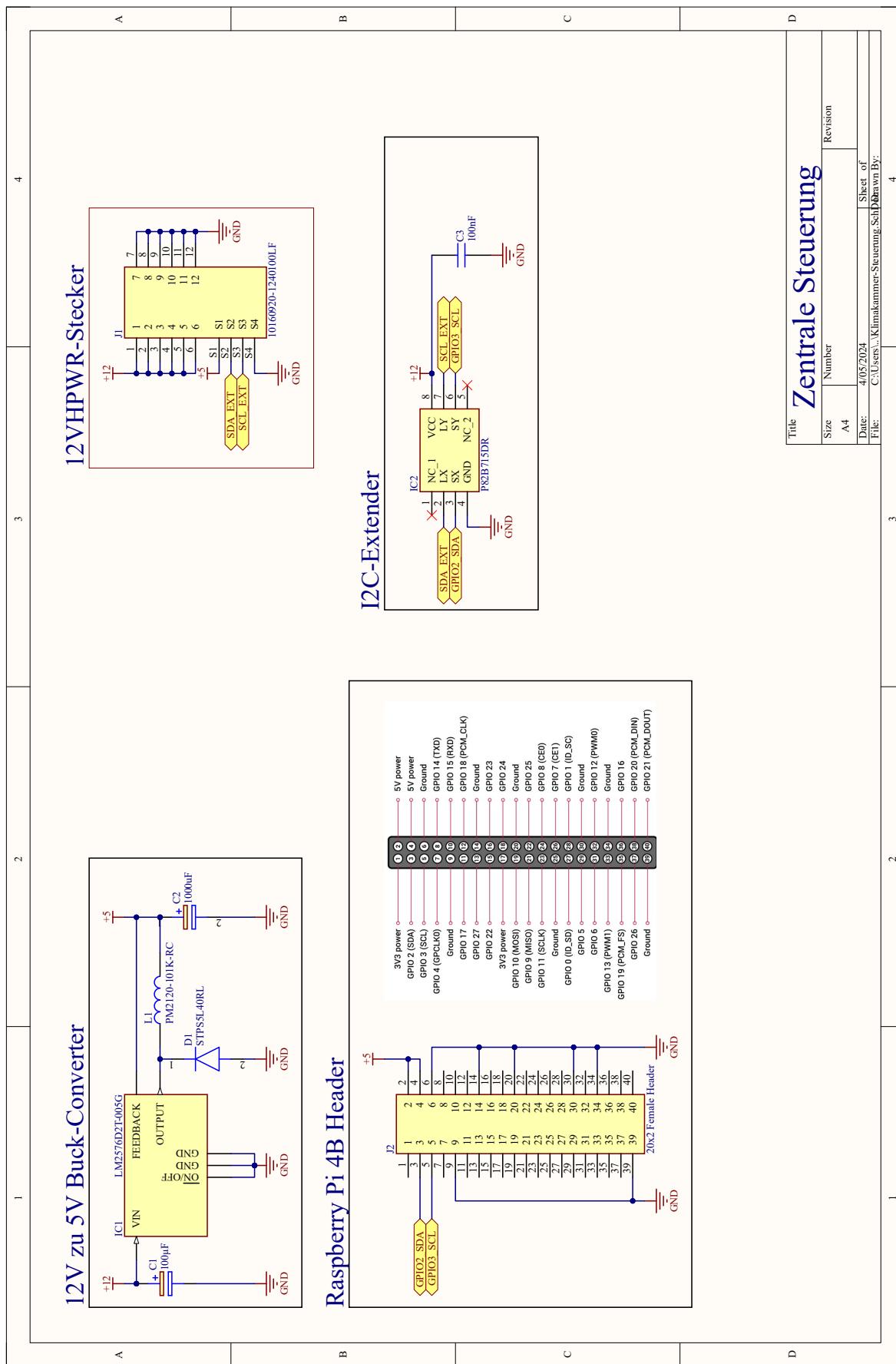
Texas Instruments. *p82b715.pdf*. Dez. 2007. URL: https://www.ti.com/lit/ds/symlink/p82b715.pdf?ts=1704579921411&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fde-de%252FP82B715%252Fpart-details%252FP82B715DR (besucht am 06.01.2024).

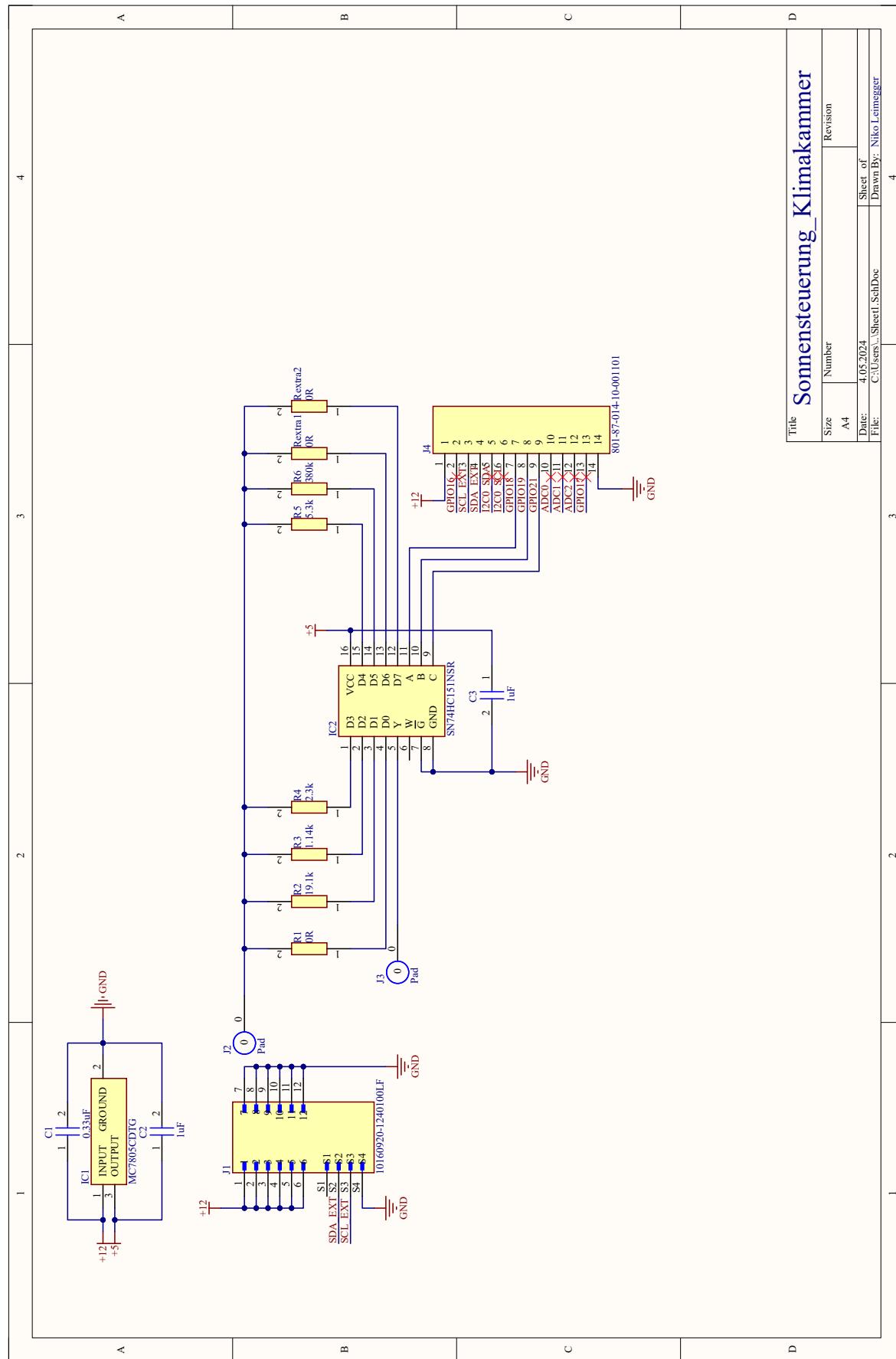
- Thingiverse.com. *Raspberry Pi 4 Case by anjanathisal*. Thingiverse. URL: <https://www.thingiverse.com/thing:3712499> (besucht am 11.03.2024).
- Uvicorn. URL: <https://www.uvicorn.org/> (besucht am 19.03.2024).
- Valdez, Jonathan und Jared Becker. *Understanding the I2C Bus*. 2015. URL: https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1710828980808&ref_url=https%253A%252F%252Fwww.google.com%252F.
- VN02A1500000G Amphenol Anytek | Mouser. Mouser Electronics. URL: <https://www.mouser.at/ProductDetail/649-VN02A15000J0G> (besucht am 04.04.2024).
- Wasserpumpe Wohnmobil 12V, Druckwasserpumpe Membranpumpe Selbstansaugende mit Druckschalter Automatik Hochdruckreinigerpumpen, für Camping Boot Autoreinigung : Amazon.de: Auto & Motorrad. URL: https://www.amazon.de/dp/B09KZVFVMPN?ref_=cm_sw_r_cp_ud_dp_9MYF29QS3533C3Z05ZR8 (besucht am 04.04.2024).
- XT60. URL: <https://docs.rs-online.com/4610/A700000008956683.pdf> (besucht am 09.03.2024).
- You, Evan. *Vue.js - The Progressive JavaScript Framework | Vue.js*. URL: <https://vuejs.org/> (besucht am 23.12.2023).
- You, Evan und Eduardo San Morote Morote. *Vue Router | The official Router for Vue.js*. URL: <https://router.vuejs.org> (besucht am 23.12.2023).

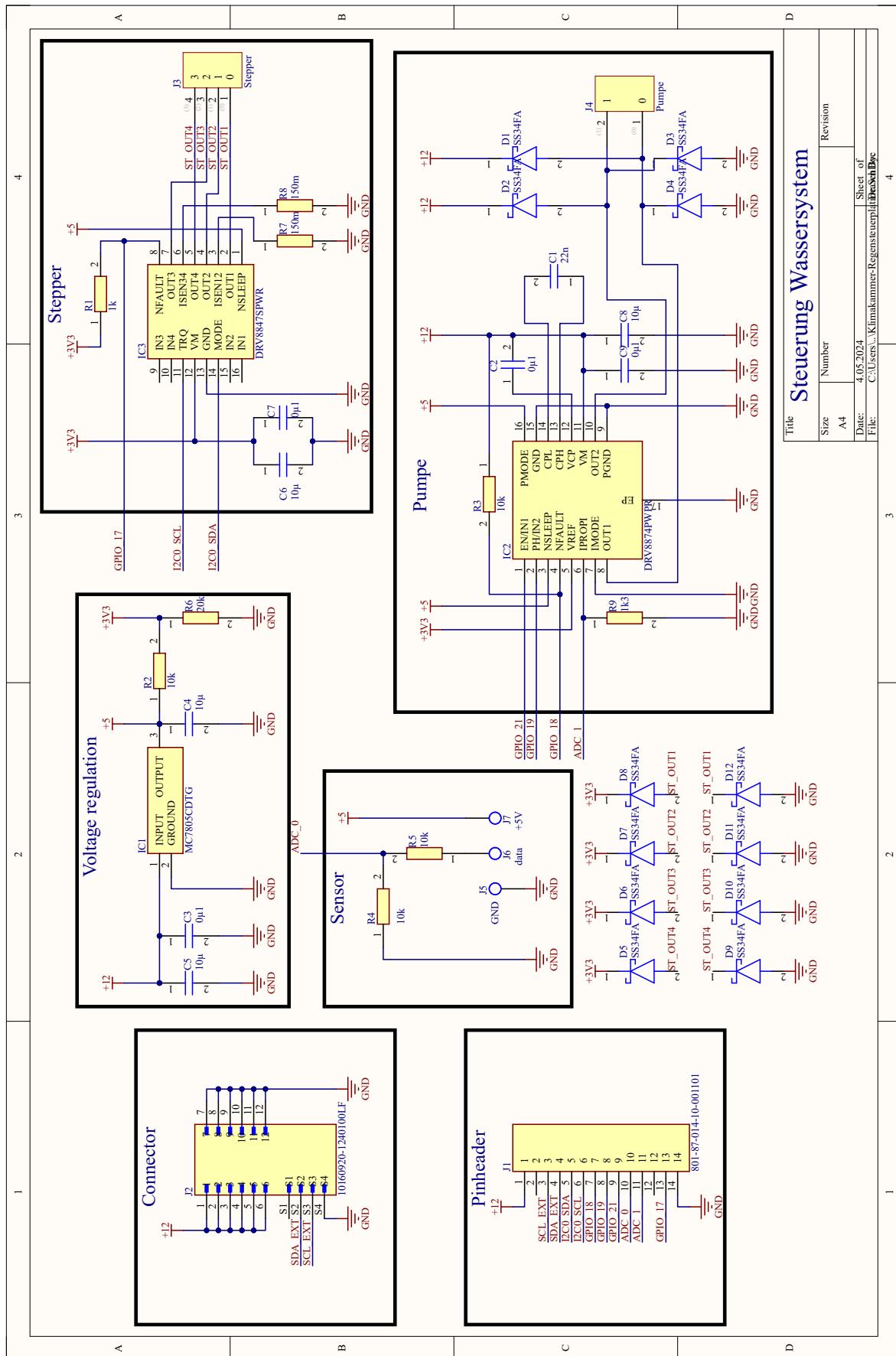










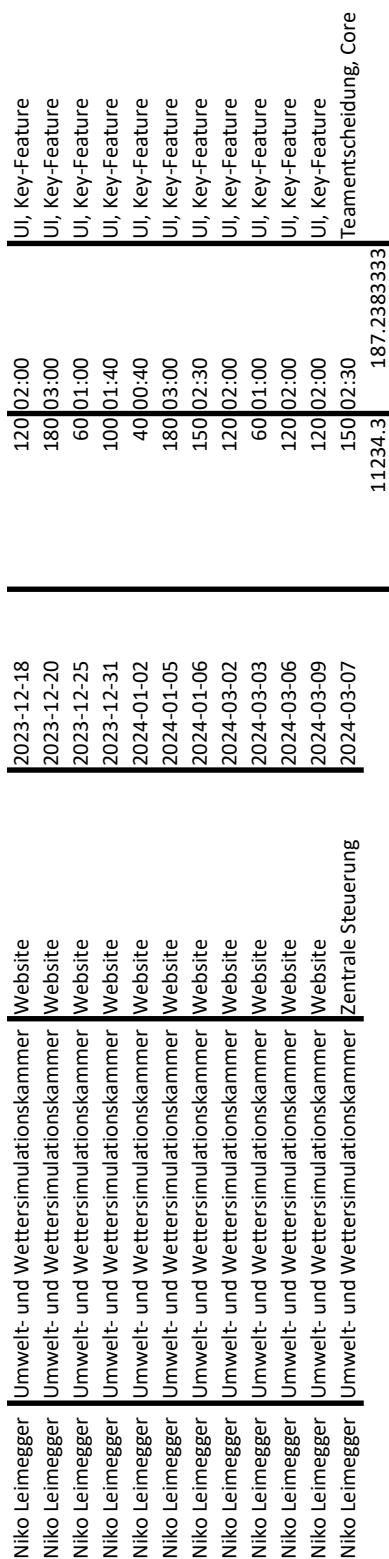


Time Tracked By	Project	Action Title	Tracked Date	Tracked (Minutes)	Labels
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2023-12-23	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2023-12-25	15	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-01-06	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-01-06	90	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-01-21	300	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-01-30	60	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-02-18	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-04	60	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-07	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-09	60	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-15	60	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-15	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-16	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-17	60	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-17	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-18	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-19	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-19	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-19	240	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-22	30	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-25	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-26	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-27	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-28	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-03-29	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-04-02	120	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-04-03	240	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Dokumentation	2024-04-04	180	Key-Feature
MCV Simon	Umwelt- und Wettersimulationskammer	Gehäuse	2023-11-13	210	Teamentscheidung, Core
MCV Simon	Umwelt- und Wettersimulationskammer	Gehäuse	2023-11-20	240	Teamentscheidung, Core
MCV Simon	Umwelt- und Wettersimulationskammer	Gehäuse	2024-01-17	240	Teamentscheidung, Core
MCV Simon	Umwelt- und Wettersimulationskammer	Gehäuse	2024-01-18	120	Teamentscheidung, Core
MCV Simon	Umwelt- und Wettersimulationskammer	Gehäuse	2024-01-26	60	Teamentscheidung, Core
MCV Simon	Umwelt- und Wettersimulationskammer	Gehäuse	2024-02-26	300	Teamentscheidung, Core

MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-02-27	135	02:15
MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-02-28	60	01:00
MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-03-06	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-03-03	60	01:00
MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-03-11	240	04:00
MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-03-27	240	04:00
MCV Simon	Umwelt- und Wittersimulationskammer	Regen	2024-03-18	240	04:00
MCV Simon	Umwelt- und Wittersimulationskammer	Sonne	2023-12-06	60	01:00
MCV Simon	Umwelt- und Wittersimulationskammer	Sonne	2024-01-09	180	03:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2023-12-12-18	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-01-11	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-01-25	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-01-25	60	01:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-01-18	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-02-15	30	00:30
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-02-20	135	02:15
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-02-22	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-02-26	15	00:15
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-02-26	60	01:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-02-29	180	03:00
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-03-07	90	01:30
MCV Simon	Umwelt- und Wittersimulationskammer	Speech-to-Text Steuerung	2024-03-15	120	02:00
MCV Simon	Umwelt- und Wittersimulationskammer	Temperaturregelung	2023-09-18	210	03:30
					10980
					183

Time Tracked By	Project	Action Title	Time Tracked Date	Tracked (Minutes)	Labels
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-26	90:01:30	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-23	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-24	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-25	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-28	10:00:10	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-28	39:00:39	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-28	20:00:20	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-30	40:00:40	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2023-12-31	30:00:30	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-01-05	150:02:30	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-01-06	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-01-22	300:05:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-01-25	90:01:30	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-01-25	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-03-06	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-03-07	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-03-09	90:01:30	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-03-16	150:02:30	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Dokumentation	2024-03-17	120:02:00	Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Feuchtigkeitregulierung	2023-09-25	83:93:01:23	Wasserkreislauf
Niko Leimegger	Umwelt- und Wetersimulationskammer	Feuchtigkeitregulierung	2023-09-25	0:02:00:00	Wasserkreislauf
Niko Leimegger	Umwelt- und Wetersimulationskammer	Gehäuse	2024-01-15	240:04:00	Teamentscheidung, Core
Niko Leimegger	Umwelt- und Wetersimulationskammer	Gehäuse	2024-01-18	180:03:00	Teamentscheidung, Core
Niko Leimegger	Umwelt- und Wetersimulationskammer	Gehäuse	2024-01-24	60:01:00	Teamentscheidung, Core
Niko Leimegger	Umwelt- und Wetersimulationskammer	Gehäuse	2024-02-01	240:04:00	Teamentscheidung, Core
Niko Leimegger	Umwelt- und Wetersimulationskammer	Gehäuse	2024-02-14	120:02:00	Teamentscheidung, Core
Niko Leimegger	Umwelt- und Wetersimulationskammer	Gehäuse	2024-02-26	240:04:00	Teamentscheidung, Core
Niko Leimegger	Umwelt- und Wetersimulationskammer	Kameraerfassung	2024-03-03	180:03:00	UI, Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Planung	2023-12-18	120:02:00	UI
Niko Leimegger	Umwelt- und Wetersimulationskammer	Sanlight kontaktieren	2023-12-06	60:01:00	Beleuchtung, Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Sonne	2024-01-02	40:00:40	Beleuchtung, Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Sonne	2024-01-08	240:04:00	Beleuchtung, Key-Feature
Niko Leimegger	Umwelt- und Wetersimulationskammer	Sonne	2024-02-19	270:04:30	Beleuchtung, Key-Feature

		Bedeutung, Key-Feature	
Niko Leimegger	Umwelt- und Wetersimulationskammer	Sonne	2024-03-11
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-01-06
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-01-11
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-01-04
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-01-14
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-01-18
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-01-25
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-02-01
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-02-29
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-03-04
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-03-04
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-03-06
Niko Leimegger	Umwelt- und Wetersimulationskammer	Speech-to-Text Steuerung	2024-03-15
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-10-09
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-10-12
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-10-02
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-10-13
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-10-15
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-10-16
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-11
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-13
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-15
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-19
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-23
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-03
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-26
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-27
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-11-20
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-06
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-07
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-09
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-10
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-11
Niko Leimegger	Umwelt- und Wetersimulationskammer	Website	2023-12-11
		300 05:00	120 02:00
		120 02:00	120 02:00
		120 02:00	120 02:00
		60 01:00	60 01:00
		120 02:00	120 02:00
		180 03:00	120 02:00
		120 02:00	120 02:00
		60 01:00	60 01:00
		420 07:00	120 02:00
		120 02:00	180 03:00
		240 04:00	150 02:30
		150 02:30	150 02:30
		30 00:30	30 00:30
		120 02:00	120 02:00
		270 04:30	300 05:00
		300 05:00	300 05:00
		150 02:30	150 02:30
		240 04:00	180 03:00
		120 02:00	120 02:00
		60 01:00	60 01:00
		124,47 02:04	56,88 00:56
		240 04:00	240 04:00



Time Tracked By	Project	Action Title	Time Tracked Date	Tracked (Minutes)	Labels
Joel Rupp	Umwelt- und Wittersimulationskammer	Backend entwickeln (Abstraction Layer)	2023-11-14	180	03:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2023-11-15	180	03:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2023-12-24	210	03:30
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-01-03	180	03:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-01-06	180	03:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-01-22	300	05:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-01-21	300	05:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-01-28	300	05:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-01-29	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Dokumentation	2024-03-04	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Fertigung	2024-02-26	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Firmware schreiben	2024-02-12	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Firmware schreiben	2024-02-14	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Firmware schreiben	2024-02-14	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Firmware schreiben	2024-02-17	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-08-16	180	03:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-09-25	360	06:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-10-02	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-10-15	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-10-17	100	01:40
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-11-09	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2023-11-27	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-01-29	480	08:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-02-20	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-02-20	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-02-22	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-02-26	210	03:30
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-02-28	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-02-28	60	01:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Gehäuse	2024-03-04	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	In Solidworks modellieren / Design	2023-08-23	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	In Solidworks modellieren / Design	2023-08-25	60	01:00
Joel Rupp	Umwelt- und Wittersimulationskammer	In Solidworks modellieren / Design	2023-10-03	120	02:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Innenkammer	2024-01-08	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Innenkammer	2024-01-10	240	04:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Innenkammer	2024-01-18	180	03:00
Joel Rupp	Umwelt- und Wittersimulationskammer	Innenkammer	2024-01-26	60	01:00

Joel Rupp	Umwelt- und Wittersimulationskammer	Verkleidung	2023-11-13
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-11-11
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-12-15
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-12-18
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-12-18
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-12-19
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-12-27
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2024-01-15
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2024-01-27
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2024-02-28
Joel Rupp	Umwelt- und Wittersimulationskammer	Website	2023-08-18
Joel Rupp	Umwelt- und Wittersimulationskammer	Zentrale Steuerung	2023-10-02
Joel Rupp	Umwelt- und Wittersimulationskammer	Zentrale Steuerung	2023-12-17
Joel Rupp	Umwelt- und Wittersimulationskammer	Zentrale Steuerung	2023-12-18
Joel Rupp	Umwelt- und Wittersimulationskammer	Zentrale Steuerung	2023-12-19
Joel Rupp	Umwelt- und Wittersimulationskammer	Zentrale Steuerung	2023-12-27
Teamentscheidung, Core			240 04:00
UI, Key-Feature			240 04:00
UI, Key-Feature			180 03:00
UI, Key-Feature			120 02:00
UI, Key-Feature			30 00:30
UI, Key-Feature			30 00:30
UI, Key-Feature			30 00:30
UI, Key-Feature			30 00:30
UI, Key-Feature			30 00:30
UI, Key-Feature			120 02:00
UI, Key-Feature			240 04:00
UI, Key-Feature			60 01:00
Teamentscheidung, Core			30 00:30
Teamentscheidung, Core			60 01:00
Teamentscheidung, Core			180 03:00
Teamentscheidung, Core			240 04:00
Teamentscheidung, Core			180 03:00
Teamentscheidung, Core			50 00:50
Teamentscheidung, Core			15942.03
			265.7005