

Projekt-Bericht

Isabell Albrecht, Erik Engelhardt, Oliver Kochan, Florian Steffens

Solarbetriebene, mobile Wetterstation

Betreuung durch: Prof. Dr. Franz Schubert
Eingereicht am: 14. Januar 2019

*Fakultät Technik und Informatik
Department Informations- und Elektrotechnik*

*Faculty of Computer Science and Engineering
Department Information and Electrical Engineering*

Inhaltsverzeichnis

1 Einführung	1
1.1 Die Wetterstation	1
1.2 Anforderungen	2
1.3 Vorüberlegungen	2
2 Spannungsversorgung	3
2.1 Anforderungen	3
2.2 Erzeugung benötigter Spannungen	4
2.2.1 3V3	4
2.2.2 5V	5
2.3 Spannungsabschaltung	6
2.4 Messung Strom/Spannung	7
3 Sensoren	8
3.1 Kompassmodul QMC5883L	8
3.1.1 Montage	8
3.2 GPS	9
3.2.1 Montage	9
3.3 Anemometer und Windfahne	9
3.3.1 Montage	9
3.4 Neigungssensor MPU6050	9
3.4.1 Montage	9
3.5 BME280	10
3.5.1 Montage	10
3.6 Anschlagssensor	10
3.6.1 Montage	10
4 C-Code	10
4.1 QMC5883L	10
4.2 GPS	10
4.3 Wind	10
4.4 MPU6050	10
4.5 BME280	10

Inhaltsverzeichnis

5 Ausrichtung des Solarpanels	10
5.1 Berechnung der Sonnenposition	11
6 Benutzeroberfläche	12
6.1 Funktionen	12
6.2 Entwicklung	18
7 3D gedruckte Komponenten	21
7.1 Hauptgehäuse	21
7.2 Nebengehäuse	23
7.3 Adaptor	25
7.4 Herstellung	27
8 Fazit	28
Literatur	29

Keywords: Leben, Universum, Alles

1 Einführung

Der folgende Bericht gibt einen Überblick über den Entwurf und die Umsetzung einer solarbetriebenen Wetterstation im Rahmen der Sensorikvorlesung. Hierbei sollen zunächst die Anforderungen und bereits vorhandene Konzepte/Komponenten kurz vorgestellt und dann sowohl auf die Soft- als auch die Hardwareseitige Umsetzung eingegangen werden.

1.1 Die Wetterstation



Abbildung 1: Gegebener Aufbau der Wetterstation

In Abbildung 1 ist der gegebene Aufbau der Wetterstation dargestellt. Dieser besteht aus einem kippbaren Solarpanel, das auf einer drehbaren Platte montiert ist.

Sowohl Dreh- als auch Kippbewegung wird über jeweils einen Getriebemotor ermöglicht. Des Weiteren ist bereits eine Windfahne zur Erfassung der Windrichtung und -geschwindigkeit montiert.

1.2 Anforderungen

Die Wetterstation soll folgende Werte erfassen:

- Temperatur
- Luftdruck
- Luftfeuchte
- Höhe über NN
- Windgeschwindigkeit
- Windrichtung
- Standort

Weiterhin soll die Wetterstation über das Solarpanel mit Strom versorgt werden, wobei aber dieses durch einen 12V Akku gepuffert werden soll, um z.B. die Nachtstunden ohne Ausfall der Spannungsversorgung überbrücken zu können. Für eine optimale Energieausbeute soll das Panel abhängig vom Sonnenstand über die beiden Getriebemotoren ausgerichtet werden. Außerdem soll der Akkuzustand erfasst und die drahtlose Kommunikation mit einem PC ermöglicht werden. Schließlich sollen die erfassten Sensordaten auf einer SD-Karte gespeichert werden.

1.3 Vorüberlegungen

Bevor die beschriebenen Anforderungen umgesetzt werden, sind einige Vorüberlegungen und die Auswahl der geeigneten Sensorik nötig. Letztere werden ausführlich in Abschnitt 3 dargestellt. Die Drahtloskommunikation soll über ein Bluetooth-Modul realisiert und um eine eigene GUI, die die Darstellung aktueller Sensordaten auf einem PC ermöglicht, erweitert werden. Des Weiteren wird die bisher vorhandene Windfahne ausgetauscht und ersetzt, da diese nur über eine unzureichende Dokumentation verfügt. Ebenfalls ist eine wasserdichte Ausführung der Wetterstation wünschenswert, die Umsetzbarkeit muss

überprüft werden. Um Energie zu sparen, soll die Ausrichtung der Wetterstation sowie die Abfrage der Sensordaten nicht kontinuierlich sondern in noch festzulegenden Zeitabständen erfolgen.

2 Spannungsversorgung

In diesem Abschnitt soll auf die hardwareseitige Umsetzung der Spannungsversorgung für die Wetterstation eingegangen werden. Ziel ist der Entwurf einer Platine, auf der sämtliche Anforderungen umgesetzt werden.

2.1 Anforderungen

Zunächst sollen in diesem Abschnitt die Anforderungen, die sich aus der Aufgabenstellung ableiten lassen, sowie solche, die sich aus den weiteren Überlegungen zur Umsetzung der Wetterstation ergeben.

- Messung des Ladestroms
- Messung der Batteriespannung (Ladezustand)
- Messung des Stromverbrauchs der Wetterstation

Des Weiteren soll der Stromverbrauch der Wetterstation so niedrig wie möglich sein, um die Puffer-Batterie zu schonen und sonnenarme Phasen bzw. die Nacht ohne Stromausfall überbrücken zu können. Die verwendete Batterie hat eine Ladeschlussspannung von 12 V. Da für den Mikrocontroller und die Sensoren allerdings Spannungsspegel von 3,3 V und 5 V benötigt werden, müssen diese auf der Platine erzeugt werden.

Aus den mechanischen Anforderungen, dass Mikrocontroller, Platine und Sensoren möglichst in einem Gehäuse untergebracht werden sollen, ergibt sich, dass die entworfene Platine auf die Pinheader des Mikrocontrollers gesteckt werden soll.

2.2 Erzeugung benötigter Spannungen

Wie in 2.1 beschrieben, werden sowohl 3,3V als auch 5V-Pegel für die Wetterstation benötigt. Angestrebt ist, dass die verwendeten Buck-Spannungsregler eine möglichst geringe Ruhestromaufnahme und einen guten Wirkungsgrad haben. Die Wahl fiel hierbei auf den *LTC3621*. Dieser hat einen Eingangsspannungsbereich von 2,7 V bis 17 V und eine Ausgangsspannung, die sich über einen Spannungsteiler am Feedback-Pin zwischen 0,6 V und der Eingangsspannung einstellen lässt. Der Ruhestrom beträgt laut Datenblatt $3,5 \mu\text{A}$ [6]. Der Regler kann einen maximalen Ausgangsstrom von 1 A liefern. Da bei der Wahl des Spannungsreglers noch keine Werte über die von der Peripherie benötigte Leistung vorlag, ist dieser Wert eventuell etwas überdimensioniert.

Die Beschaltung des Reglers entspricht den Empfehlungen des Datenblatts, wie sie in Abbildung 2 dargestellt ist:

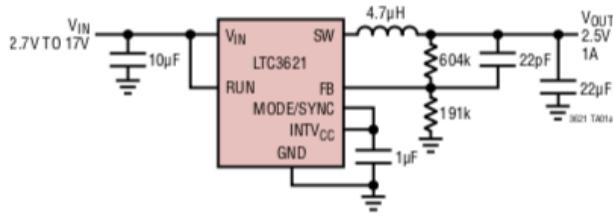


Abbildung 2: Darstellung des LTC3621 mit beispielhafter Beschaltung gemäß Datenblatt [6]

Zusätzlich verfügt dieser Regler über die Option, über einen Befehl des Mikrocontrollers manuell ein- bzw. ausgeschaltet zu werden, was weiterhin günstig für den Gesamtstromverbrauch ist.

2.2.1 3V3

Im folgenden soll hauptsächlich auf die Dimensionierung des Spannungsteilers zur Erzeugung von 3,3 V eingegangen werden. Der Regler hat dabei eine interne Referenzspannung am FB-Pin, die 0,6 V beträgt, über die die Ausgangsspannung geregelt wird.

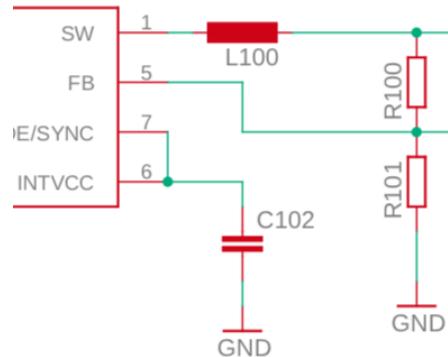


Abbildung 3: Spannungsteiler am FB-Pins des 3V3 Reglers

Es kann somit der folgende Spannungsteiler aufgestellt werden:

$$\frac{U_{FB}}{U_a} = \frac{R_{101}}{R_{101} + R_{100}} \quad (1)$$

Mit $U_{FB} = 0,6 \text{ V}$, $U_a = 3,3 \text{ V}$ und $R_{101} = 150 \text{ k}\Omega$ ergibt sich:

$$\frac{0,6 \text{ V}}{3,3 \text{ V}} = \frac{150 \text{ k}\Omega}{150 \text{ k}\Omega + R_{100}} \Rightarrow R_{100} = 680 \text{ k}\Omega \quad (2)$$

Der Spannungsteiler wurde hochohmig dimensioniert, um den Stromfluss möglichst klein zu halten.

2.2.2 5V

Die Bestimmung des Spannungsteilers für die Erzeugung von 5V erfolgt analog zu Abschnitt 2.2.1. Es lässt sich folgender Spannungsteiler aufstellen:

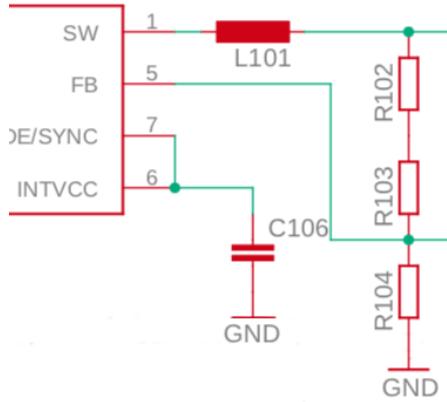


Abbildung 4: Spannungsteiler am FB-Pins des 5V Reglers

Dieser wird beschrieben durch:

$$\frac{U_{FB}}{U_a} = \frac{R_{104}}{R_{104} + R_{103} + R_{102}} \quad (3)$$

Mit $U_{FB} = 0,6 \text{ V}$, $U_a = 3,3 \text{ V}$ und $R_{104} = 100 \text{ k}\Omega$ ergibt sich:

$$\frac{0,6 \text{ V}}{3,3 \text{ V}} = \frac{100 \text{ k}\Omega}{10 \text{ k}\Omega + R_{103} + R_{102}} \Rightarrow R_{103} = 680 \text{ k}\Omega; R_{102} = 47 \text{ k}\Omega \quad (4)$$

2.3 Spannungsabschaltung

Im Zuge der Überlegungen bezüglich möglicher Energiesparmaßnahmen wurden sowohl das Bluetooth- als auch das GPS-Modul als große Verbraucher ermittelt. Da beide Module auch nicht dauerhaft benötigt werden – das GPS-Modul nur alle 15 Minuten zur Neuausrichtung des Panels und das Bluetooth-Modul nur nach Bedarf – ist es sinnvoll, die Betriebsspannungen beider Module schaltbar zu machen. Eine Möglichkeit dafür ist die Verwendung eines p-Kanal-Mosfets, der von einem n-Kanal-Mosfet getrieben wird. Diese Schaltung wird in Abbildung 5 dargestellt.

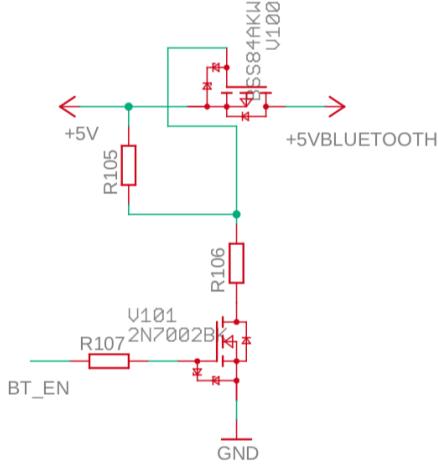


Abbildung 5: Spannungsabschaltung Bluetooth-Modul

Bei einem Einschaltsignal über BT_EN vom Mikrocontroller schaltet der n-Kanal-Mosfet V101 durch; der Spannungsteiler aus R_{105} und R_{106} ist aktiv. Es sei $R_{105} = 100 \text{ k}\Omega$ und $R_{106} = 1 \text{ k}\Omega$. Daraus ergibt sich für die Gate-Source-Spannung des p-Kanal-Transistors U_{GS} :

$$U_{GS} = 5 \text{ V} - 5 \text{ V} \cdot \frac{1 \text{ k}\Omega}{101 \text{ k}\Omega} = 4,95 \text{ V} \quad (5)$$

Damit schaltet der p-Kanal-Transistor mit $U_{GS,th} = 1,6 \text{ V}$ sicher durch. Die Schaltung für das GPS-Modul wird analog dazu entworfen.

2.4 Messung Strom/Spannung

Aus den Anforderungen abgeleitet wird auf der Platine die Messung von Strom und Spannung implementiert. Für die Messung der Stromaufnahme werden zwei Stromsensoren vom Typ ACS172 verwendet. Diese messen den Ladestrom der Batterie und die Stromaufnahme des Boards und geben diese Werte an den Mikrocontroller. Die Spannungsmessung wird mit einem einfachen Spannungsteiler umgesetzt, sodass bei $U_{Bat} = 12 \text{ V}$ 3,3 V nicht überschritten werden. Mit $R_{109} = 270 \text{ k}\Omega$ und $R_{110} = 100 \text{ k}\Omega$ ergibt sich eine maximale Spannung von 3,24 V.

3 Sensoren

3.1 Kompassmodul QMC5883L

Als Kompass dient uns der **QMC5883L**. Verwendet wird er, um die Ausrichtung der Wetterstation gen Norden zu messen. Diese Information wird benötigt, um das Solarpanel korrekt zur Sonne auszurichten zu können.

Mit einer Genauigkeit von 1° bis 2° ist er für unsere Zwecke ausreichend genau. Ebenfalls für den Sensor sprechen seine geringe Stromaufnahme von $75 \mu\text{A}$ und die Möglichkeit ihn mittels I²C anzusprechen.

3.1.1 Montage

Das Kompassmodul hat 5 Anschlüsse, von denen wir vier verwenden: V_{CC}, GND, SDA und SCL. Der fünfte Anschluss wird nur bei einer häufigen Datenübertragung benötigt, er sendet die Information, dass Daten zum Abruf bereit stehen. Dargestellt ist die Verschaltung in Abbildung 6.

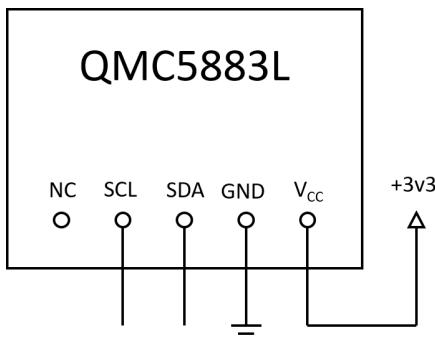


Abbildung 6: Verschaltung des QMC5883L

Da der Sensor das Magnetfeld misst, muss dieser entfernt von der Windfahne und dem Anemometer platziert werden, da beide mit magnetischen Bauteilen arbeiten. Andernfalls könnte es zu Fehlmessungen kommen. Umgesetzt wurde dies so, dass an der einen Stütze des Solarpanels das Kompassmodul und an der anderen die Windfahne, das Anemometer und das gegenüber magnetischen Beeinflussungen relativ unempfindliche GPS-Modul angebracht wurden.

3.2 GPS

Um den Standort des Geräts zu bestimmen und auch um die Berechnung der korrekten Auslenkung des Solarpanels auf Grund des Standorts zu ermöglichen, musste ein GPS-Modul genutzt werden. Wir haben uns für den

3.2.1 Montage

Da das GPS-Modul in Einheit mit einer Antenne funktioniert, musste das Modul entfernt von der Elektronik und damit außerhalb des Gehäuses platziert werden. Dazu bot sich an, das GPS-Modul an der zweiten Stütze des Solarpanels zu platzieren. Das Modul wurde dafür ebenso in einem Nebengehäuse untergebracht (s. Kap. 7.2). Die Antenne des Moduls wurde dabei mittels Kabelbinder an der Stütze angebracht. Dieses Vorgehen war einer dauerhaften Anbringung, etwa durch Kleber, vorzuziehen, da die Bauteile so leichter ausgetauscht werden können.

3.3 Anemometer und Windfahne

3.3.1 Montage

3.4 Neigungssensor MPU6050

3.4.1 Montage

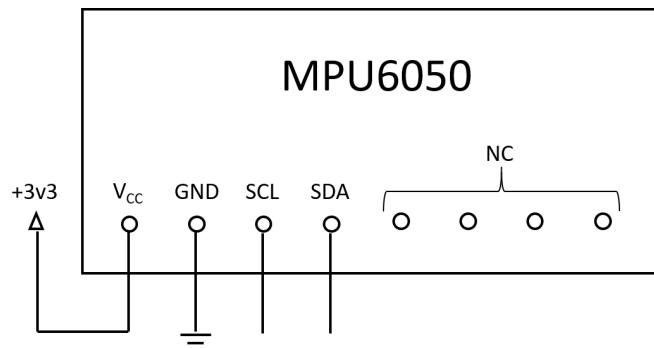


Abbildung 7: Verschaltung des MPU6050

3.5 BME280

3.5.1 Montage

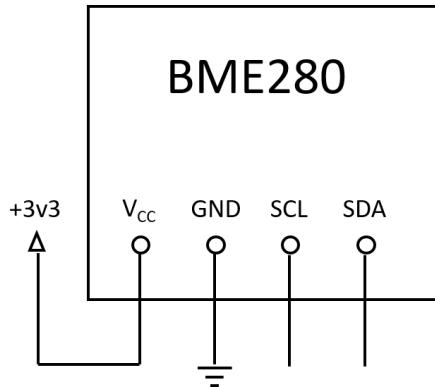


Abbildung 8: Verschaltung des BME280

3.6 Anschlagssensor

3.6.1 Montage

4 C-Code

4.1 QMC5883L

4.2 GPS

4.3 Wind

4.4 MPU6050

4.5 BME280

5 Ausrichtung des Solarpanels

Um die Leistungsaufnahme des Solarpanels zu optimieren ist es notwendig dieses direkt auf die Sonne auszurichten und diese Ausrichtung auch in geeigneten Zeitabständen zu

korrigieren. Im Vergleich mit einem fest ausgerichteten Solarpanel konnten J. Rizek *et al.* mit einem nachgeführten Solarpanel beispielsweise die Leistungsaufnahme um durchschnittlich 30% erhöhen [3].

Hierfür kommen grundsätzlich verschiedene Methoden in Frage. In diesen Fall soll die Position der Sonne relativ zur Wetterstation auf Grundlage des Längen- und Breitengrades, der Uhrzeit und des Kalendertages berechnet werden. Diese Information werden über das GPS-Modul bereitgestellt. Anschließend wird das Solarpanel mit Hilfe der Motoren, des Kompass-Moduls und des, am Panel befestigten Neigungssensors, auf die Sonne ausgerichtet.

5.1 Berechnung der Sonnenposition

Da die Formeln zur Berechnung der Sonnenposition in diesem Projekt lediglich benutzt werden wird an dieser Stelle auf eine Herleitung verzichtet. Die verwendeten Formeln können beispielsweise im *Astronomical Almanac* [1] gefunden werden. M. L. Roderick beschreibt die nötigen Berechnungen in seinem Report [5] und liefert zudem compilierbaren C-Code. Dieser wird im folgenden zur Berechnung der Sonnenposition verwendet.

Die Position der Sonne im Bezug auf einen Beobachter auf der Erde lässt sich durch die Werte *Zenith* und *Azimut* eindeutig beschreiben. Der Zenith beschreibt den Winkel zwischen einer Linie vom Beobachter zur Sonne und der Vertikalen. Der Azimut den Winkel zwischen der Horizontalen und Norden. Dabei stehen beispielsweise ein Azimut von 90° für Osten, 180° für Süden und 270° für Westen. Eine Veranschaulichung kann in Abbildung 9 gefunden werden.

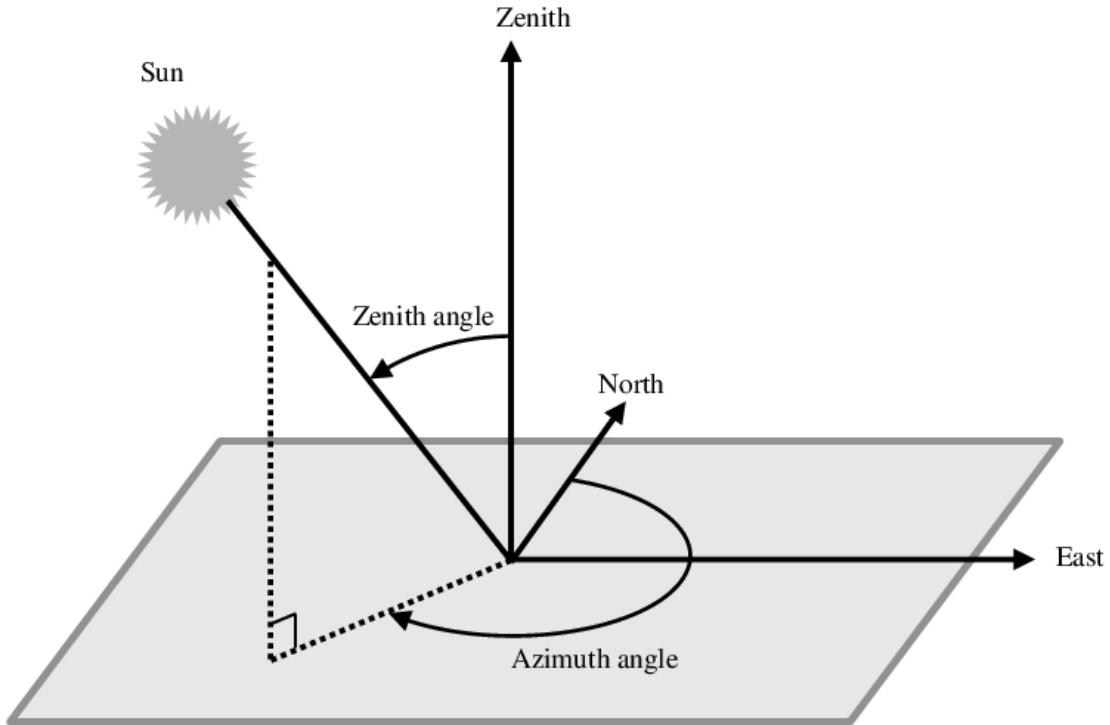


Abbildung 9: Beschreibung der Sonnenposition durch Zenith und Azimut [4]

Die Genauigkeit der verwendeten Formeln beträgt laut dem *Astronomical Almanac* 0.01° bezogen die auf die Position und 0.1 min bezogen auf die Zeit.

6 Benutzeroberfläche

Um eine leichte Handhabung der Wetterstation zu ermöglichen wird eine, auf dem PyQt5 [7] Framework basierende, Benutzeroberfläche verwendet. Für die Kommunikation mit der Wetterstation muss der Computer über eine Bluetooth-Schnittstelle verfügen

In diesem Kapitel wird die zur Verfügung stehenden Funktionen sowie die Entwicklung der Benutzeroberfläche beschrieben.

6.1 Funktionen

Die Benutzeroberfläche (s. Abb. 10) verfügt über zwei Hauptansichten: eine graphische (s. Abb. 11) und eine tabellarische (s. Abb. 12).

Diese unterscheiden sich jeweils nur durch die Anzeigeviante (s. Abb. 10 und 12 (1)).

Über das Konsolenfenster (s. Abb. 11 (2)) können die im Kapitel (HIER KAPITELLINK EINFÜGEN) beschrieben AT-Befehle an die Wetterstation gesendet werden. Sowohl die gesendeten als auch die empfangenen Daten werden im Konsolenfenster angezeigt. Über die Schaltfläche *Clear* lässt sich der Inhalt des Konsolenfensters löschen. Die Größe des Konsolenfensters und des Anzeigefensters kann beliebig verändert werden.

Das Menü (s. Abb. 11 (3)) enthält die zur Bedienung notwendigen Befehle. Es sind nicht alle AT-Befehle implementiert. Die AT-Befehle, welche nicht implementiert sind können manuell über das Konsolenfenster an die Wetterstation gesendet werden. Einige der Menübefehle wurden aufgrund von Zeitmangel nicht implementiert und sind daher ausgelaugt. Das Menü enthält die Folgenden Befehle:

- File
 - Save (nicht implementiert): Speichert die empfangenen Daten in einer CSV-Datei ab.
 - Open (nicht implementiert): Lädt die Daten aus einer CSV-Datei in das Programm.
- Control
 - Set Time
 - * UTC: Setzt die Uhrzeit und das Datum der Wetterstation auf die Koordinierte Weltzeit. Die Uhrzeit wird der Computeruhr entnommen.
 - * Custom (nicht implementiert): Öffnet ein Fenster in welches eine beliebige Uhrzeit und Datum eingegeben werden kann. Diese wird anschließend auf die Wetterstation gespielt.
 - Set Position
 - * Hamburg: Setzt die Position der Wetterstation auf (53.556354, 10.022650) (HAW Hamburg).
 - * Custom (nicht implementiert): Öffnet ein Fenster in welches beliebige Koordination eingegeben werden können. Diese werden anschließend auf die Wetterstation übertragen.

- Adjust Orientation (nicht implementiert): Gibt der Wetterstation den Befehl sich sofort neu auszurichten.
- Set Update-Interval: Bestimmt das Interval, in welchem eine Verbindung mit der Wetterstation aufgebaut wird. Nach dem Aufbau der Verbindung werden die neuen Messdaten angefordert, empfangen und dargestellt. Anschließend wird die Verbindung, zwecks Energiesparan, wieder geschlossen.
 - * 5 sec: fünf Sekunden
 - * 1 min: eine Minute
 - * 15 min: fünfzehn Minuten
 - * 1 h: eine Stunde
 - * Manuel: Kein Automatisches Anfordern der Messdaten. Zum Anfordern der Messdaten muss die *Update*-Schaltfläche betätigt werden.
- Set Measuring-Interval: Schickt einen Befehl an die Wetterstation, welcher vorgibt in welchem Interval Messungen durchzuführen sind.
 - * 5 sec: fünf Sekunden
 - * 15 sec: fünfzehn Sekunden
 - * 1 min: eine Minute
- View: Bestimmt das Zeitintervall, welchen in der graphischen Anzeige dargestellt wird.
 - * Last Minute: letzte Minute
 - * Last 15 Minutes: letzten fünfzehn Minuten
 - * Last Hour: letzte Stunde
 - * Last Day: letzter Tag
 - * Last Week: letzte Woche
 - * Last Month: letzter Monat

- * Custom (nicht implementiert): Der dargestellte Bereich wird über die Elemente zur Darstellung der Start- und Stopzeit (s. Abb. 11 (4)) eingestellt. Der Bereich wird im Gegensatz zu den anderen Optionen nicht automatisch mit voranschreiten der Zeit geupdated.
- Mode:
 - Enable Debug: Aktiviert den Debug-Modus. Debugnachrichten werden angezeigt.
 - Disable Debug: Deaktiviert den Debug-Modus. Debugnachrichten werden nicht angezeigt.
 - Enable Com: Baut manuell eine Bluetooth-Verbindung zur Wetterstation auf. Dies ist notwendig, wenn Befehle zur Wetterstation gesendet werden sollen. Für das automatische, periodische Einlesen der Messdaten ist dies nicht notwendig. Es gibt momentan keine Möglichkeit den für die Verbindung verwendeten Com-Port über die Benutzeroberfläche anzupassen. Stattdessen muss dieser manuell in der Datei *main_ctrl.py* geändert werden.
 - Disable Com: Schließt die Bluetooth-Verbindung zur Wetterstation.

Die Anzeigen (4) (s. Abb. 11) zeigen, wie bereits erwähnt, das Intervall an, welches dargestellt wird. Ist dieses Intervall nicht auf *Custom* gestellt, so ist die Endzeit immer die aktuelle Zeit. Die Startzeit ergibt entsprechend des eingestellten Zeitintervalls.

Über die Schaltflächen (5) (s. Abb. 11) können zwei Messkurven ausgewählt werden, welche dargestellt werden sollen. Die Schaltfläche *Update* ist für das manuelle Anfordern der Messdaten notwendig. Unter den Schaltflächen werden die, über das Fadenkreuz (6) (s. Abb. 11) ausgewählten, Messpunkte schriftlich dargestellt.

6 Benutzeroberfläche

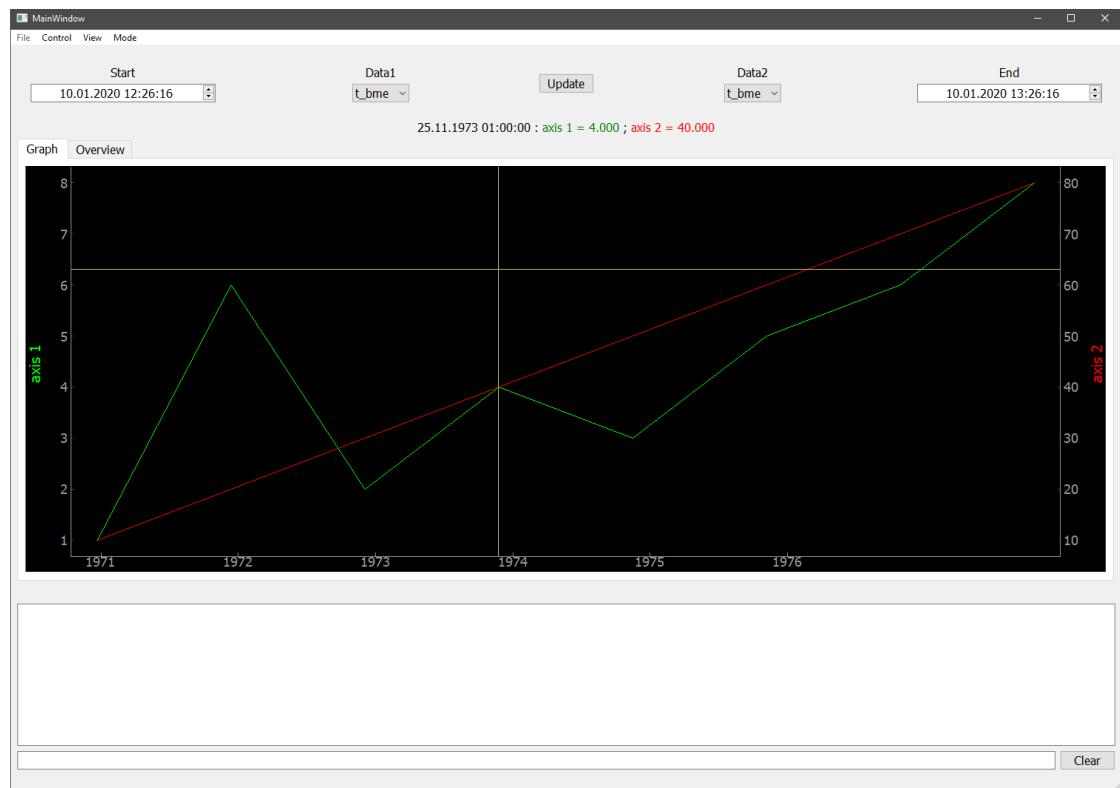


Abbildung 10: Benutzeroberfläche nach dem öffnen. Dagestellt sind zwei vordefinierte Testkurven, welche nach Empfang des ersten Messwertes gelöscht werden.

6 Benutzeroberfläche



Abbildung 11: Benutzeroberfläche: (1) Graphische Darstellung der Messwerte. (2) Konsolefenster. (3) Kontrollmenü mit den wichtigsten Befehlen. (4) Start- und Endzeit der graphischen Darstellung. (5) Auswahl der dargestellten Messwerte, Schaltfläche zum manuellen Update der Messwerte und Anzeige der Messwerte an der aktuellen Fadenkreuzposition. (6) Fadenkreuz zum Anzeigen spezifischer Messwerte.

6 Benutzeroberfläche

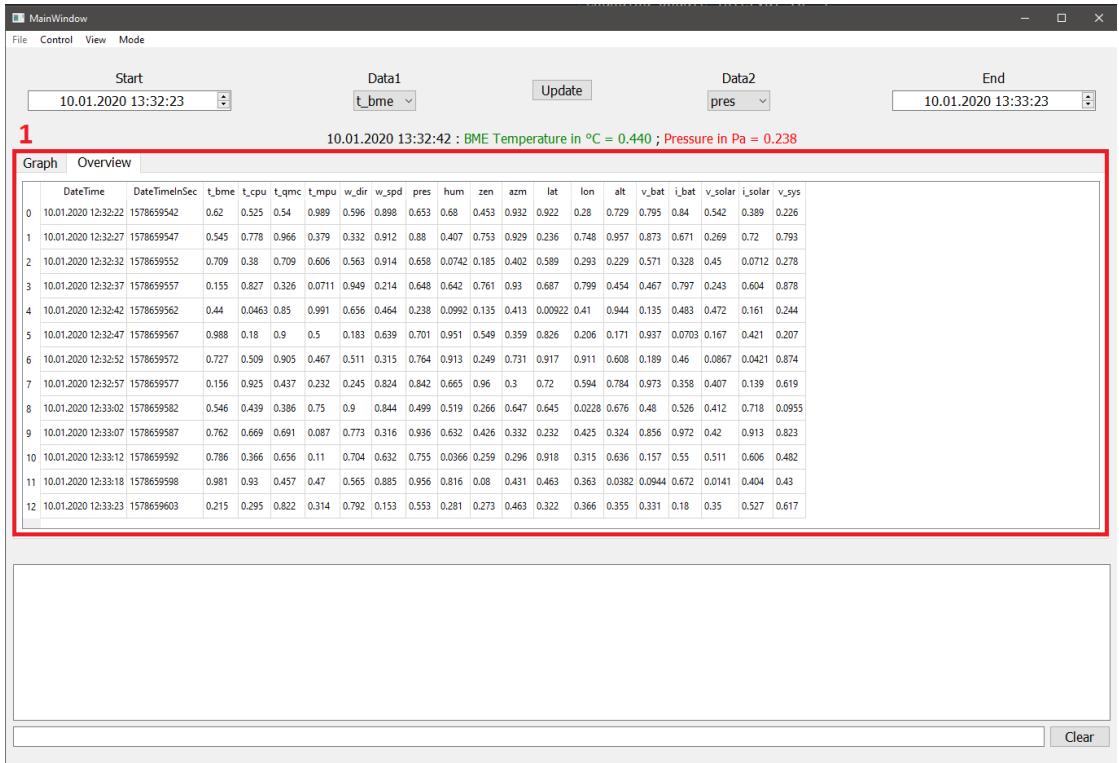


Abbildung 12: Benutzeroberfläche: (1) Tabellarische Darstellung von simulierten Werten.

6.2 Entwicklung

In diesem Kapitel werden die, für die Benutzeroberfläche implementierten Funktionen, *nicht* im Detail beschrieben. Es soll lediglich ein Überblick über die Verwendeten Technologien und den Aufbau der Projekts gegeben werden. Bei der Entwicklung wurde versucht der Code in einer Art und Weise zu schreiben, welche es anderen Personen ermöglicht ohne eine ausführliche Beschreibung des gesammten Codes, an diesem weiterzuarbeiten.

Als Entwicklungssprache wurde Python 3 verwendet. Der verwendete Quellcode befindet sich im Ordner UI. Die Oberfläche wurde nach dem Model-View-Controller Design-Muster (s. z.B. [2]) entwickelt. Für einige der Funktionalitäten wurde Code aus den, im PyQt5-Package enthaltenen, Beispielen als Vorlage verwendet.

Projektstruktur

Das Projekt ist wie folgt strukturiert:

```
UI
├── .git
├── controllers
│   └── main_ctrl.py
├── model
│   └── model.py
├── mywidgets
│   ├── dateaxis.py
│   ├── dateaxisitem.py
│   ├── mplwidget.py
│   └── mygraphicswidget.py
├── resources
│   ├── main_view.ui
│   └── mvc_app.qrc
└── views
    ├── main_view.py
    └── main_view_ui.py
├── .gitignore.txt
└── build.bat
└── mvc_app.py
└── requirements.txt
```

Die Hauptdatei des Projekts ist *mvc_app.py*. In ihr werden das Model, der Controller und das View initialisiert und verbunden. Das Model (*model.py*) beinhaltet alle Datenstrukturen, also sowohl die Messwerte als auch Variablen, welche den Programmablauf steuern. Das View (*main_view.py* und *main_view_ui.py*) beinhaltet die Benutzeroberfläche und bestimmt welche Funktionen beim Aktivieren welcher Schaltflächen aufgerufen werden. Diese Funktionen befinden sich im Controller (*main_ctrl.py*). Im Ordner *mywidgets* befinden sich die im View implementierten benutzerdefinierten graphischen Anzeigen.

Die Datei *requirements.txt* beinhaltet eine Liste aller benötigten Python-Packages. Mit dem Befehl „`pip install -r requirements.txt`“ können diese alle automatisch installiert werden.

6 Benutzeroberfläche

Im Ordner *resources* befinden die mit *PyQt5-Designer* erstellten UI-Dateien.

Erstellung der Oberfläche

Zum Erstellen der Oberfläche wurde das graphische Tool *PyQt5-Designer* (s. Abb. 13) verwendet. Das Übersetzen der mit diesem Tool erstellten *.ui*- und *.qrc*-Dateien in ausführbaren Python-Code erfolgt über das Tool *pyuic5*. Zum automatisieren dieses Prozesses wurde das Skript *build.bat* angelegt. Nach Bearbeitung der Oberfläche im *PyQt5-Designer* sollte dieses ausgeführt werden, um die Änderungen zu übernehmen.

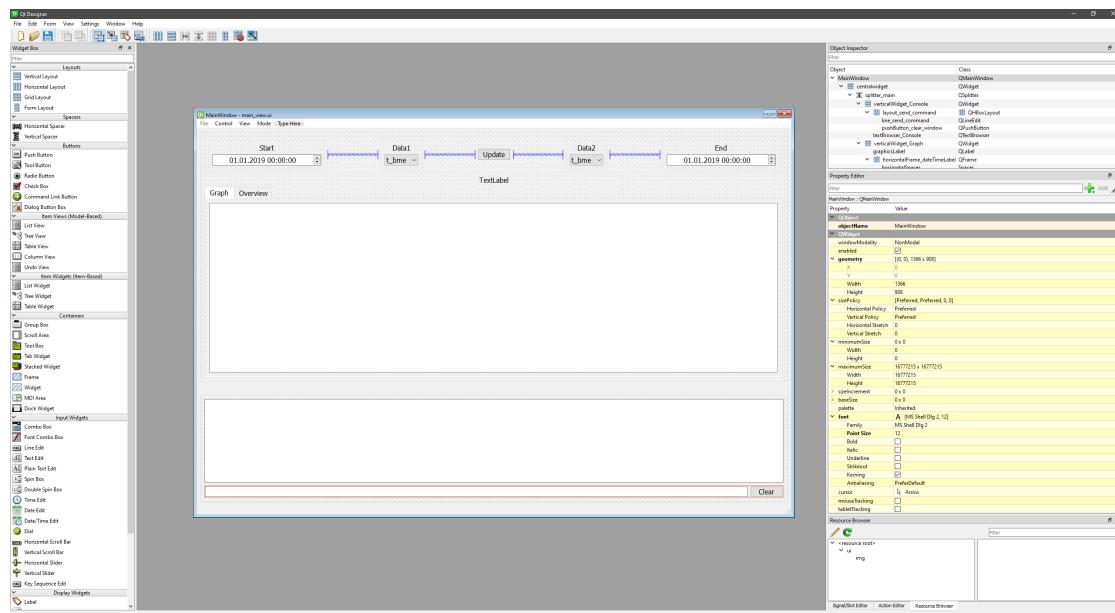


Abbildung 13: Erstellen der Oberfläche mit PyQt5-Designer.

Verwendete Python-Packages

Es werden die folgenden Python-Packages Programm verwendet:

- **PyQt5**: Als Framework für die Oberfläche.
- **serial**: Für die serielle Kommunikation, über Bluetooth, mit der Wetterstation.
- **pandas**: Für die Strukturierung der Messdaten.
- **pyqtgraph**: Für die graphische Darstellung der Messdaten.

- numpy: Für das Erstellen von Testdaten.

7 3D gedruckte Komponenten

Für das Unterbringen des Mikroprozessors und der Sensoren werden zwei verschiedene Gehäusevarianten verwendet. Dieses Kapitel beschäftigt sich mit deren Entwurf sowie dem Entwurf der Adaptors, mit welchem die Windfahne und das Anemometer an der Wetterstation befestigt sind.

Für den Entwurf der Komponenten wurde die Studentenversion von Autocad Fusion 360 verwendet.

7.1 Hauptgehäuse

Das Hauptgehäuse beinhaltet das Mikroprozessor-Board mit den zwei aufsteckbaren Platinen sowie den Sensoren. Drei der Sensoren befinden sich nicht im Hauptgehäuse (s. Kap. 7.2).

Das Gehäuse besteht aus zwei Teilen (s. Abb. 14), dem Hauptteil (links) und dem Deckel (rechts). Das STM8 Nucleo Board wird von drei Pins in Position gehalten (s. Abb. 14, 15). In der Vorderseite befindet sich zwei Aussparungen (s. Abb. 16). Über die Rechte ist der USB-Port des Mikroprozessor-Boards erreichbar. Hinter der Linken befindet sich der Motortreiber, welcher auf einer leicht erhöhten Platform plaziert wird (s. Abb. 15). Über die Aussparung an der rechten Seite kann die SD-Karte erreicht werden. Die Aussparung an der hinteren Seite wird verwendet um alle externen Komponenten mit dem Mikroprozessor-Board zu verbinden.

Im Deckel befinden sich drei Löcher, in welche Status-LEDs untergebracht werden.

Das Hauptteil und der Deckel werden lediglich aufeinander gesteckt. Bedingt durch die Geometrie entsteht ein relativ fester Halt. Optional können die Komponenten über die vier Aussparungen an den Seiten, mit beispielsweise Kabelbindern, zusätzlich gesichert werden.

7 3D gedruckte Komponenten

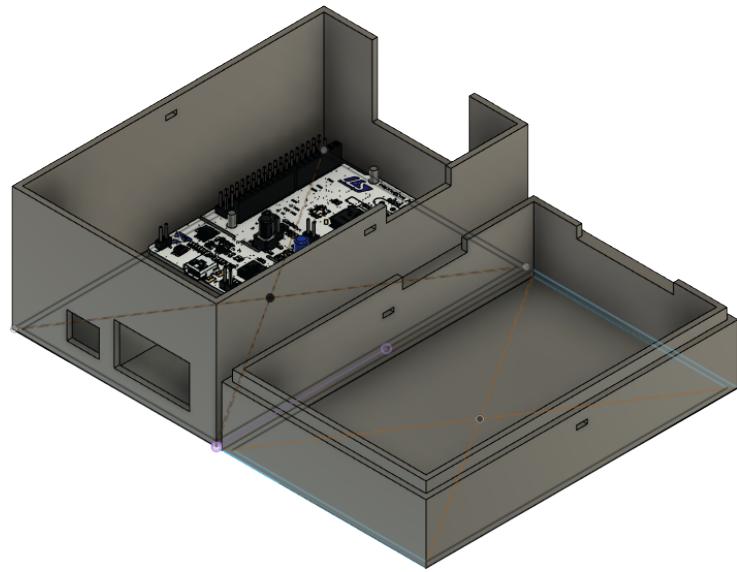


Abbildung 14: Hauptgehäuse

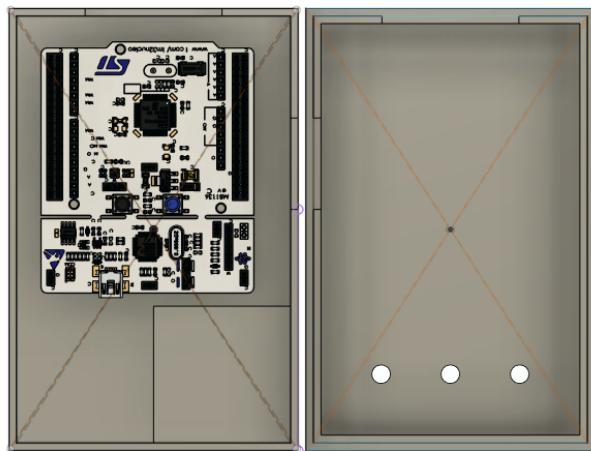


Abbildung 15: Hauptgehäuse, Ansicht von oben

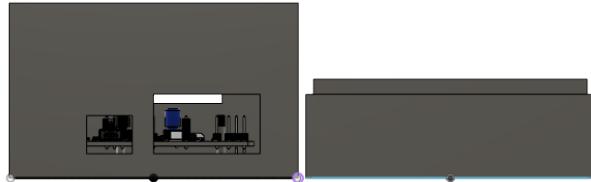


Abbildung 16: Hauptgehäuse, Ansicht von vorne

7.2 Nebengehäuse

Es wird jeweils ein Nebengehäuse für folgende Sensoren verwendet:

- der Neigungssensor, welcher an der Unterseite des Solarpanels befestigt werden muss
- der GPS-Empfänger, welcher für besseren Empfang am oberen Ende der Wetterstation befestigt wird
- das Kompass-Modul, welches empfindlich auf elektro-magnetische Störungen reagiert und aus diesem Grund auch am oberen Ende der Wetterstation untergebracht wird (auf der Seite gegenüber des GPS-Empfängers)

Nebengehäuse bestehen aus zwei Teilen, welche aufeinandergesteckt werden und zusätzlich, optional, über Kabelbinder, gesichert werden (s. Abb. 17, 19).

Die Befestigung der Nebengehäuse werden über eine Schraube und eine Mutter am Aluminium-Profil der Wetterstation befestigt (s. Abb. 18). Das Nebengehäuse, welches sich auf dem Solarpanel befindet kann nicht geschraubt werden und wird entsprechend mit Klebstoff befestigt.

7 3D gedruckte Komponenten

Die runden Aussparungen an den linken und rechten Seiten dienen dem Durchführen von benötigten Leitungen.

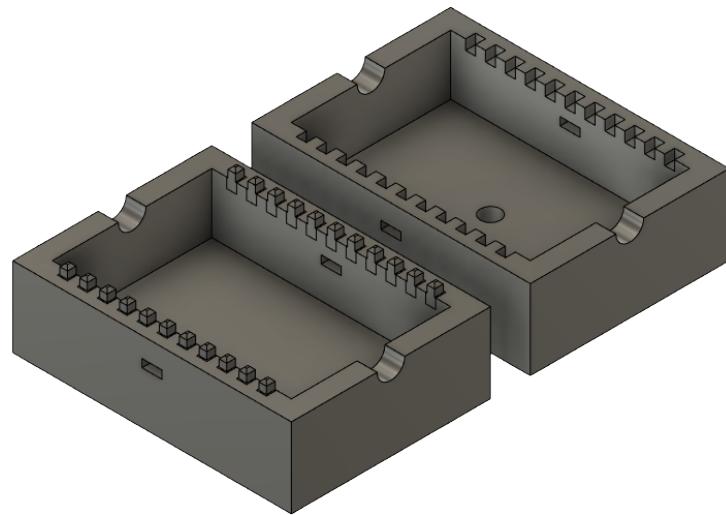


Abbildung 17: Nebengehäuse

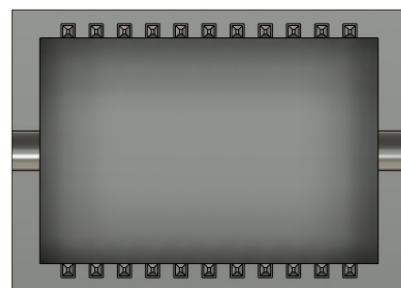
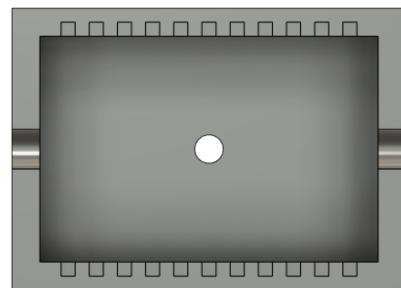


Abbildung 18: Nebengehäuse, Ansicht von oben

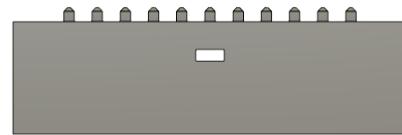


Abbildung 19: Nebengehäuse, Ansicht von vorne

7.3 Adaptor

Der Adaptor wird verwendet um eine feste Verbindung zwischem dem Alluminium-Profil der Wetterstaion und dem Mast, mit Windfahne und Anemometer, herzustellen.

Die Verbindung zur Wetterstation erfolgt durch Stecken der vier trapez-förmigen Steckkontakte am unteren Ende des Adaptors (s. Abb. 20, 22).

Der Mast wird von Oben auf die angepasste Aussparung gesteckt. Optional kann der über eine Schraube und Mutter fixiert werden (s. Abb. 20, 21).

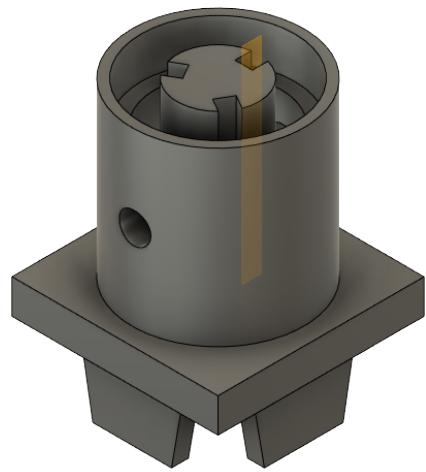


Abbildung 20: Adapter

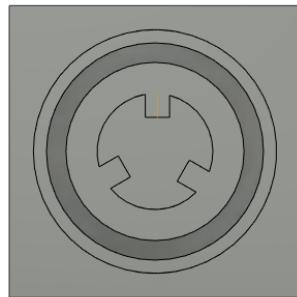


Abbildung 21: Adapter, Ansicht von oben

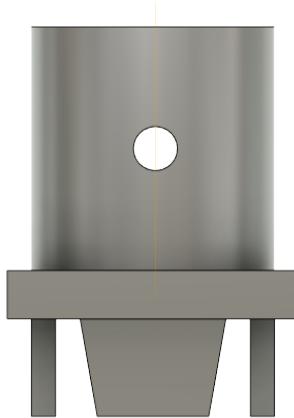


Abbildung 22: Adapter, Ansicht von vorne

7.4 Herstellung

Alle Komponenten wurden auf einem *Ender 5* in PLA mit einer Schichtdicke von 0.2 mm gedruckt. Es wurden drei Top-, Bottom- und Wandschichten gedruckt. Da die Komponenten keine großen Kräfte aushalten müssen wurden sie lediglich mit einem Infill von 20 % gedruckt.

8 Fazit

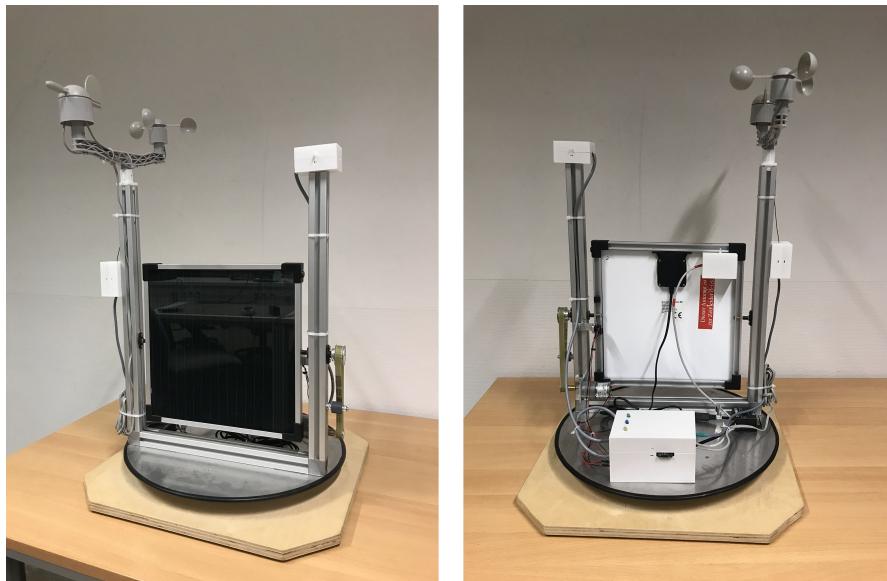


Abbildung 23: Gegebener Aufbau der Wetterstation

In Abbildung 23 wird der fertige Aufbau der Wetterstation dargestellt. Die Anforderung der Erfassung von Temperatur, Luftdruck und -feuchte wird mit dem BME280, der im Hauptgehäuse platziert ist, umgesetzt. Für die Erfassung von Windrichtung und -geschwindigkeit konnte ein neues Anemometer gefunden und implementiert werden.

Im Hauptgehäuse ließen sich Mikrocontroller, Platinen und Sensoren sinnvoll unterbringen (Abbildung 24). Die Wasserfestigkeit des Aufbaus konnte leider mit den verfügbaren Mitteln in der vorgegebenen Zeit nicht erreicht werden. Das verwendete 3D-Druck-Material ist nicht für einen abgedichteten Aufbau geeignet.

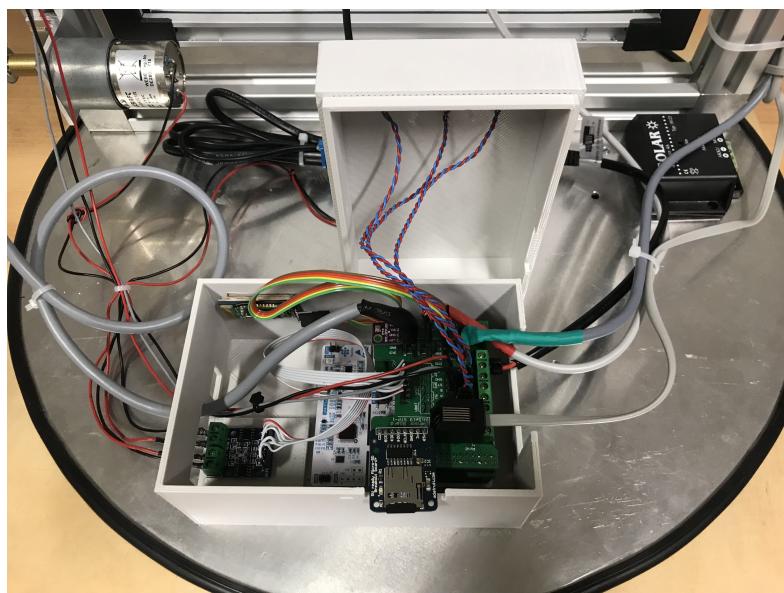


Abbildung 24: Blick in das Hauptgehäuse

Sowohl hardware- als auch softwareseitig wurden verschiedene Energiesparmaßnahmen implementiert. Die letztendliche Laufzeit müsste noch in einem Langzeittest ermittelt werden.

Grundlegend lässt sich sagen, dass alle gegebenen Anforderungen erfüllt wurden und sogar um eine zusätzliche Benutzeroberfläche zum Abrufen der gemessenen Werte erweitert.

Literatur

- [1] ANON.: *The Astronomical Almanac*
- [2] DEACON, John: *Model-View-Controller (MVC) Architecture*. 2009. – URL <https://www.rareparts.com/pdf/MVC.pdf>. – Zugriff: 10.01.2020
- [3] J., Rizek ; Y., Chaiko: Solar Tracking System: More Efficient Use of Solar Panels. In: *World Academy of Science, Engineering and Technology* 41 (2008). – URL <http://www.mugla.edu.tr/data/03060010/belgeler/solar%20tracking.pdf>. – Zugriff: 17.10.2019

- [4] NOU, Julien ; CHAUVIN, Rémi ; THIL, Stéphane ; GRIEU, Stéphane: A new approach to the real-time assessment of the clear-sky direct normal irradiance. In: *Applied Mathematical Modelling* 40 (2016), 03
- [5] RODERICK, M.L.: Methods for calculating solar position and day length including computer programs and subroutines. In: *Resource Management Technical Reports* (1992). – URL <https://researchlibrary.agric.wa.gov.au/cgi/viewcontent.cgi?article=1122&context=rmtr>. – Zugriff: 17.10.2019
- [6] TECHNOLOGY, Linear: *Datenblatt LTC3621*. – URL <https://www.analog.com/media/en/technical-documentation/data-sheets/3621fc.pdf>. – Zugriff: 18.11.2019
- [7] TOMPSON, Phil: *Python bindings for the Qt cross platform application toolkit*. 2019. – URL <https://pypi.org/project/PyQt5/>. – Zugriff: 28.12.2019