



HOCHSCHULE RUHR WEST  
UNIVERSITY OF APPLIED SCIENCES

# **Einsatz von Internet of Things im Bereich der Grünflächenbewässerung unter Einbindung der Anlieger**

## **Projektarbeit**

im Eingebettete Systeme 2  
Studiengang Angewandte Information  
der Hochschule Ruhr West

**Nils Milewski - 10010480**

**Lars Dubrau - 10011735**

**Alex Heiming - 10012086**

Erstprüfer\*in: Prof. Dr. Michael Schäfer

Betreuer: Lars Hofmann

Bottrop, September 2022

## Kurzfassung

Gegenstand der hier vorgestellten Arbeit ist ein Proof of Concept (PoC) welches sich mit der Informationstechnischen Seite der Thematik von Grünflächenbewässerung und Anwohnereinbindung befasst. Zunächst wird der aktuelle Stand des Projektes vorgestellt, woraus sich ein Konzept entwickelt wie Anlieger in die Bewässerung der Umgebung mit eingebunden werden können. Abschließend wird ein Konzept entwickelt, welches die Software und Hardwareempfehlungen für den Projektpartner bereitstellt. Kein Teil dieser Arbeit ist die genaue Realisierung der Umgebung, namentlich Aufbau der Pflanzbehälter

**Schlagwörter:** Grünflächenbewässerung, Internet of Things, Microcontroller,

## Abstract

This paper provides a proof of concept about the computer science part of a green space watering system. First, a general overview of the current project state is given. From this given state, a detailed draft will be developed. The draft contains how the residents can support this project. The final concept provides the software and hardware suggestions for the project partner. This paper does not include the realization of the environment, in particular the creation of a planter.

**Keywords:** greenspace watering, Internet of Things, Microcontroller

## Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>Kurzfassung .....</b>                                  | <b>2</b>  |
| <b>Abstract .....</b>                                     | <b>2</b>  |
| <b>Inhaltsverzeichnis .....</b>                           | <b>3</b>  |
| <b>Abbildungsverzeichnis .....</b>                        | <b>5</b>  |
| <b>Abkürzungsverzeichnis .....</b>                        | <b>6</b>  |
| <b>1 Einleitung.....</b>                                  | <b>7</b>  |
| <b>2 Ziele.....</b>                                       | <b>8</b>  |
| <b>3 Stand der Technik .....</b>                          | <b>9</b>  |
| 3.1 Pegelmessung.....                                     | 9         |
| 3.1.1 Elektronisch.....                                   | 9         |
| 3.1.2 Druck.....  | 9         |
| 3.1.3 Ultraschall .....                                   | 9         |
| 3.1.4 Optisch.....  | 10        |
| 3.1.5 Gewählte Methode .....                              | 10        |
| 3.2 Kommunikation.....                                    | 11        |
| 3.2.1 Wireless Local Area Network (WLAN) .....            | 11        |
| 3.2.2 Cellular – Long Term Evolution (LTE).....           | 11        |
| 3.2.3 Long Range Radio Wide Area Network (LoRa WAN) ..... | 11        |
| 3.2.4 Gewählte Technologie.....                           | 11        |
| <b>4 Komponenten .....</b>                                | <b>12</b> |
| 4.1 Kosten der Bauteile .....                             | 12        |
| 4.2 VI53L0X Time of Flight Sensor .....                   | 13        |
| 4.3 RFM95 LoRa Funk Transceiver.....                      | 14        |
| 4.4 Aufbau der Hardware .....                             | 15        |
| 4.5 GPS (Optional) .....                                  | 16        |
| <b>5 Kommunikation .....</b>                              | <b>17</b> |
| 5.1 Modul und Gateway .....                               | 17        |
| 5.2 Endnutzer und Gateway .....                           | 17        |
| <b>6 Website für Nutzer.....</b>                          | <b>18</b> |
| 6.1 Entwurf.....  | 18        |
| 6.2 Funktionen .....                                      | 19        |
| 6.3 Implementierung.....                                  | 21        |

|          |   |           |
|----------|---|-----------|
| <b>7</b> | <b>3D-Druck des Gehäuses .....</b>                | <b>23</b> |
| 7.1      | Entwurf.....                                      | 23        |
| 7.2      | Design.....                                       | 23        |
| 7.3      | Slicing .....                                     | 25        |
| <b>8</b> | <b>Messungen der einzelnen Module.....</b>        | <b>27</b> |
| 8.1      | Wasserpegel .....                                 | 27        |
| 8.2      | Batterie.....                                     | 27        |
| <b>9</b> | <b>Fazit.....</b>                                 | <b>28</b> |
|          | <b>Literaturverzeichnis .....</b>                 | <b>29</b> |
|          | <b>Anhang.....</b>                                | <b>30</b> |
|          | <b>Kenntlichmachung der Eigenleistungen .....</b> | <b>31</b> |
|          | <b>Erklärungen .....</b>                          | <b>32</b> |

## Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1 VL53L0X Time of Flight Modul und Anschlussplan .....           | 13 |
| Abbildung 2 RFM95 LoRa Funk Transceiver Komponente und Anschlussplan ..... | 14 |
| Abbildung 3 Schematischer Antennenaufbau .....                             | 14 |
| Abbildung 4 Aufbau und Verkabelung der Hardware .....                      | 15 |
| Abbildung 5 Fertiger Aufbau mithilfe eines Steckbrett .....                | 16 |
| Abbildung 6 Detailansicht eines Geräts .....                               | 18 |
| Abbildung 7 Website zur Darstellung der Geräte .....                       | 19 |
| Abbildung 8 Website Marker für ein Geräte (Administrator Ansicht) .....    | 19 |
| Abbildung 9 Erstellung eines Gerätes .....                                 | 20 |
| Abbildung 10 Bearbeiten eines Gerätes .....                                | 21 |
| Abbildung 11 Löschen eines Gerätes .....                                   | 21 |

## Abkürzungsverzeichnis

|                  |  |
|------------------|--|
| HRW              | Hochschule Ruhr West                             |
| PoC              | Proof of Concept                                 |
| IOT              | Internet of Things                               |
| WAN              | Wide Area Network                                |
| UUID             | Universally Unique Identifier                    |
| WLAN             | Wireless Local Area Network                      |
| LTE              | Long Term Evolution                              |
| LTE-M[TC]        | Long Term Evolution Machine [Type Communication] |
| LoRa             | Long Range Radio                                 |
| BLT              | Bluetooth Low Energy                             |
| LiPo             | Lithium Polymer                                  |
| GND              | Ground   |
| GPS              | Global Positioning System                        |
| GPIO             | General Purpose Input Output                     |
| SLK/SCK          | Serial Clock                                     |
| MOSI             | Master Output Slave Input                        |
| MISO             | Master Input Slave Output                        |
| NSS              | Slave Select                                     |
| DIO              | Digital Input Output                             |
| SPI              | Serial Peripheral Interface                      |
| I <sup>2</sup> C | Inter-Integrated Circuit                         |
| JSON             | Java Script Object Notation                      |
| TOF              | Time of Flight                                   |
| LiDAR            | Light detection and Ranging                      |
| STL              | STereoLithography                                |
| G-Code           | Geometric Code                                   |
| PLA              |  |

## 1 Einleitung

Diese Arbeit ist im Rahmen des Moduls „Eingebettete Systeme 2“ von Professor Michael Schäfer entstanden. Ziel des Moduls war die Unterstützung eines HRW-Startups bei ihrem Projekt. Es sollte mit Unterstützung dieser ein Prototyp entwickelt werden bei deren Entwicklung eigenständige Lösungen gefunden und implementiert werden mussten. Mehrere Startups standen zur Auswahl und man musste sich für eines entscheiden.

Unsere Wahl fiel auf das Startup „XiLab“, dessen Ziel es ist, die Bewässerung von städtischen Grünanlagen zu verbessern und Austrocknung vorzubeugen. Erreicht werden sollte dieses Ziel mit einer IOT-Lösung, die gebaut werden musste, sowie einer Softwarelösung, über die das System abrufbar ist

## 2 Ziele

In vielen Städten liegt der Fokus in der modernen Städteplanung auf der Begrünung und Bepflanzung. Bei der Planung von öffentlichen Plätzen und Wohngebieten, werden bereits Grünflächen, Wiesen und Gärten berücksichtigt. Ein Ziel ist es, die Flächenversiegelung zu minimieren. Flächenversiegelung meint den Überbau von Gras- oder Erdflächen mit Beton, Asphalt oder sonstigen Baumaterialien. Durch die versiegelten Flächen kann bei Starkregen das Wasser schlecht ablaufen und die Temperatur in der Stadt ist verglichen mit dem Umland um mehrere Grad höher. Das erhöht die Überflutungsgefahr und verringert auch die Lebensqualität.

Ein weiterer positiver Nebeneffekt ist, dass die vielen Pflanzen das CO<sup>2</sup> aus der Luft aufnehmen und so die Luftqualität verbessert wird, was gerade in Großstädten ein anhaltendes Problem ist.

Für die Bewässerung der vielen Grünanlagen ist in der Regel die Stadt verantwortlich. Dies funktioniert auch so weit gut, allerdings ist durch den massiven Anstieg von Grünanlagen sowie durch den erhöhten Wasserbedarf im Sommer die Arbeitslast gestiegen und die meisten Städten haben nicht genug Kapazitäten, um alle Grünflächen zu bewässern. Aufgrund dessen muss priorisiert werden und viele Grünflächen trocknen aus. Wiesen können sich teils von selbst wieder erholen, Fassadenbegrünungen allerdings nicht. Weiterhin leidet die Ästhetik von begrünten Fassaden, wenn alles vertrocknet und braun ist.

Hier kommt unser System ins Spiel. In jedes Beet, jeden Blumentopf oder Wiese kann eines unserer Systeme eingebaut werden. Die Geräte bestehen aus einem Behälter mit einer bestimmten Kapazität, einer Pumpe, Feuchtigkeitsmesser und einem Microcontroller. Der Microcontroller misst die aktuelle Feuchtigkeit des Bodens und lässt bei Erreichung eines Schwellenwertes eine bestimmte Menge Wasser über die Pumpe in die Erde geben. So ist die Bewässerung vollautomatisiert sichergestellt. Anwohner oder sonstige Interessenten können sich zusätzlich noch alle aktiven Systeme auf einer Karte anschauen und Informationen wie Batterieladung, Wasserstand oder die genaue Position anhand von Koordinaten abrufen.

Die Behälter lassen sich von Laien mit Wasser wieder auffüllen, so können Anwohner bequem die in ihrer Nachbarschaft befindlichen Systeme überwachen und bei Bedarf mit Wasser auffüllen.

Die Verantwortung der Bewässerung wird so automatisiert und die Befüllung der Behälter auf mehrere Schultern verteilt, da nun auch Anwohner sich darum kümmern können.



## **3 Stand der Technik**

### **3.1 Pegelmessung**

#### **3.1.1 Elektronisch**

Die einfachste elektronische Messung ist die Kapazitive. Bei der Methode werden Widerstandsveränderungen genutzt, um die Füllhöhe zu bestimmen. Hierfür wird ein Sensor der Metallische Leiterbahnen besitzt in dem Wasser platziert. Sobald sich der Wasserstand verändert, wird sich auch der Widerstand dementsprechend erhöhen oder verringern, ein hoher widerstand bedeutet das zwischen den Leiterbahnen Luft als Medium ist und dementsprechend kein Wasser vorhanden ist.

#### **3.1.2 Druck**

Der Druck kann mithilfe von zwei physikalischen Eigenschaften gemessen werden. Die erste Eigenschaft von Wasser ist der Niveaueausgleich bei verschiedenen Wasserhöhen. Zusätzlich besitzt eine ein Meter hohe Wassersäule den Druck von ein Bar. Diese Eigenschaften lassen sich durch ein senkrecht Rohr realisieren. In diesem Rohr muss ein spezieller Drucksensor platziert werden. Sobald Wasser verbraucht wird, gleicht sich der Wasserstand im Rohr dem des Kanisters an. Durch diese Wasseranpassung verändert sich der Druck in dem Rohr was dazu führt das der Sensor eine Veränderung an den angeschlossenen ESP meldet.

Eine alternative zu der Technischen Druckmessung wäre die Gewichtsmessung. Für diese Methode müssen Wägezellen unterhalb des Kanisters befestigt werden. Zunächst müssen dann Referenzwerte genommen werden, die dann in ein Prozentbereich abgebildet werden. Sobald sich der Wasserstand ändert, wird dementsprechend das Gewicht des Kanisters ändern und durch die Abbildung kann dann die aktuelle Füllhöhe in Prozente rekonstruiert werden.

#### **3.1.3 Ultraschall**

Ultraschal gehört zu der Familie der Time of Flight Sensoren.

Die Orientierung von Fledermäusen kann als eine Analogie für die Ultraschallmessung genutzt werden. Ein Audioquelle erzeugt ein Ton oberhalb von 20kHz, dieser Ton bewegt sich dann im Raum vorwärts und trifft nach einer Zeit auf ein Hindernis. Dieses Hindernis sorgt dafür das der Ton wieder zurückbewegt und auf einen Sensor trifft, die verstrichene Zeit wird dann mithilfe des Schalausbreitung in eine Distanz umgewandelt. Die Wasseroberfläche stellt in dem Projekt das Hindernis dar.

#### **3.1.4 Optisch**

Die optische Messung gehört, genauso wie die Ultraschalmessung, der ToF Familie an. Je nach eingesetzter Technik wird eine sichtbare oder nicht sichtbare Frequenz des Lichts gewählt, wobei die letztere weiterverbreitet ist.

#### **3.1.5 Gewählte Methode**

Die Kriterien für die Methode sind Platzverbrauch, Wasserdichtigkeit, Langlebigkeit und Preis. Aufgrund des hohen Preises fallen sowohl die Lidar Technik als auch die Wasserdichten Ultraschaltechnik raus aus der Wahl. Die Messung mithilfe von Wägezellen setzen eine absolut gerade und stabile Ebene voraus was aber in diesem Projekt nicht garantiert werden kann bzw. nicht gegeben ist, dieser Negative Punkt sortiert diese Technik raus. Die Druckmessung ist ein Klassischer Ansatz für die Pegelmessung aber der Aufbau benötigt zu viel Platz.

Die übrigbleibenden Techniken sind kapazitiv elektrisch und optisch mithilfe eines Speziellen Sensors, eine Kombination beider stellt ein robustes und zuverlässiges Messsystem dar. Aufgrund fehlender Sensoren wird die kapazitive Messung nicht genutzt daher bleibt nur die optische übrig, diese wird in einem späteren Artikel genauer beschrieben.

## **3.2 Kommunikation**

Da die Module mit einem Server kommunizieren müssen wird ein dementsprechend geeignetes Netzwerk aufgebaut. Aufgrund der Umgebung fallen alle Kabelgebundene Technologien weg weshalb WLAN, Cellular und LoRa zur Verfügung stehen.

### **3.2.1 Wireless Local Area Network (WLAN)**

Der in Frage kommende WLAN Standard wäre IEEE 802.11a, dieser besitzt eine Reichweite von 2000m was für die Umgebung absolut in Ordnung wäre. Jedoch wird der Standard nicht von der Eingesetzten Hardware unterstützt. Eine Lösung für das Problem mit der Reichweite wäre ein komplexes Netzwerk von WLAN-Zugriffspunkten.

### **3.2.2 Cellular – Long Term Evolution (LTE)**

Diese Technik setzt auf das Mobilfunknetz mithilfe von LTE-M. Mittels LTE können zuverlässig und konstant Datenraten bis zu 375kbit/s Download und 300kbit/s Upload erreicht werden. Der LTE-M Standard bietet zudem die Möglichkeit Energie zu sparen was gerade bei Batteriebetrieb ein Vorteil ist. Jedoch ist die Bindung an ein Mobilfunkanbieter ein größerer Nachteil.

### **3.2.3 Long Range Radio Wide Area Network (LoRa WAN)**

LoRa bietet die Möglichkeit gezielt kleine Datenpakete auszutauschen, wodurch der Energieverbrauch nur auf das Nötige reduziert wird. Die Reichweite betragen von 2000m (städtische Umgebung) bis hin zu 10000m (Ländliche Umgebung). Die Datenrate beträgt maximal 50kbit/s was für die zusendenden Daten absolut ausreichend ist.

### **3.2.4 Gewählte Technologie**

Die ausschlaggebenden Kriterien sind Reichweite und Energieeffizienz. Abgesehen von Kabelgebundenen und Bluetooth bieten alle Technologien irgendeine Möglichkeit längere Distanzen zu überwinden. Jedoch fällt hier die WLAN-Technik negativ auf da diese ein Komplexes Netzwerk von Zugriffspunkten benötigt, diese Komplexität würde den Kosten-Nutzen-Faktor enorm sprengen, weshalb WLAN nicht gewählt wurde. Die Cellular Technik bietet zwar viele Vorteile und stellt damit einen geeigneten Kandidaten dar, da die Verfügbaren Komponente alle einen Mobilfunkanbieter benötigen fällt diese Technik ebenfalls raus. Somit ist die Wahl auf LoRa gefallen da dies eine Ideale Technik für lokale IoT Netzwerke bildet.

## 4 Komponenten

### 4.1 Kosten der Bauteile

Die Hardwarerealisierung der einzelnen Module wurde recht einfach gehalten und besteht ausfolgenden Bauteilen.

| Komponente                    | Preis (Stand 30 Mai 2022) |
|-------------------------------|---------------------------|
| ESP WROOM-32                  | 10,49€                    |
| VL53L0X Time of Flight Sensor | 6,99€                     |
| RFM95 LORA Funk Transceiver   | 8,76€                     |
| LiPo Akkumulator              | 9,99€ - 14,99€            |
| <b>Summe (Ohne GPS)</b>       | <b>36.23€ - 41.23€</b>    |
| GPS Modul mit AT6558 Chip     | 9.99€                     |
| Summe (Mit GPS)               | 46,22€ - 51.22€           |

Das Gateway wird durch ein Raspberry PI realisiert

Diverse Kabel und eine Antenne welche aus etwas Stabileren Kabel erstellt werden kann.

## 4.2 VL53L0X Time of Flight Sensor

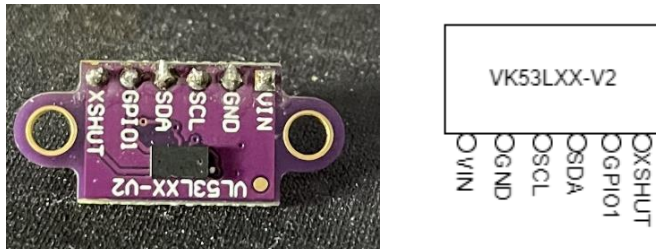


Abbildung 1 VL53L0X Time of Flight Modul und Anschlussplan

Für die Wasserstandmessung wird der VL53L0X ToF Sensor eingesetzt, dieser arbeitet auf Entfernungen bis zu 2m und bietet die Möglichkeit auf kurze Distanzen sehr hohe Genauigkeiten bei der Messung zu erzielen. Der eingesetzte Laser arbeitet mit einer für das Auge nicht Schädliche Wellenlänge ohne das sichtbare Licht emittiert wird.

Alle Komponenten des Sensors sind in einem sehr kleinen Gehäuse untergebracht was ein Idealen Einsatz in kleinen bzw. engen Umgebungen ermöglicht. Der VL53L0X ToF Sensor nutzt I<sup>2</sup>C für die Kommunikation mit anderen Geräten.

Zusätzlich zu dem I<sup>2</sup>C Anschlusspins besitzt der Sensor noch ein GPIO und XSHUT Anschluss. Wenn mehrere Sensoren, die dieselbe Strecke messen können unter Umständen Interferenzen auftreten. Diese Interferenzen können durch Deaktivieren der einzelnen Sensoren vermieden werden, hierfür müssen die nicht benutzten Sensoren mittel dem XSHUT Pin angesprochen werden.

Die hier gezeigte Abbildung ist ein VL53L0X ToF Sensor auf einer dafür bereits vorgesehenen Platine welche mithilfe von Schrauben an einem Gehäuse befestigt werden kann.

Die Funktionsweise ist sehr einfach erklärt, der Laser strahlt sein Licht punktuell in den Raum, nach einer gewissen Zeit wird das Licht zurück reflektiert und trifft auf einen Sensor. Die Zeit, die von Abstrahlen bis zu der Rückkehr vergeht wird gemessen woraus sich dann mithilfe der Lichtgeschwindigkeit die Distanz bestimmen lässt.

### 4.3 RFM95 LoRa Funk Transceiver

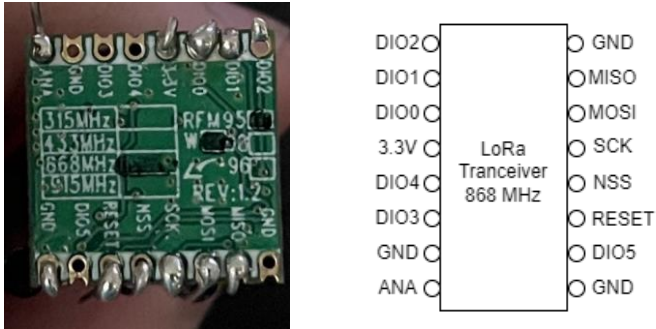


Abbildung 2 RFM95 LoRa Funk Transceiver Komponente und Anschlussplan

Damit der ESP LoRa unterstützt muss dieser um ein RFM95 Transceiver Modul erweitert werden. Der eingesetzte LoRa Chip ist in folgenden Frequenzen verfügbar, bei der Wahl sind die Lokalitäten zu beachten.

| Region                    | Frequenz | Antennenlänge (Selbstbau) |
|---------------------------|----------|---------------------------|
| 299792458                 | 315 MHz  | 95.172 cm                 |
| Europa / Asien            | 433 MHz  | 69.236 cm                 |
| Europa / Indien           | 868 MHz  | 34.538 cm                 |
| Australien / Nord Amerika | 915 MHz  | 32.762 cm                 |

In Deutschland wird die genutzte Frequenz von der Netzagentur verwaltet, diese hat den 868MHz Bereich für die Nutzung mit LoRa frei gegeben.

Zur Kommunikation mit anderen Komponenten nutzt der RFM95 das SPI. Der Chip besitzt zu den Typischen SPI Pins noch sechs IO Pins jedoch wird in dem hier vorgestellten Aufbau nur der DIO0 bin benutzt, die restlichen fünf werden nicht verbunden. Um zu senden wird eine externe Antenne am ANA Pin benötigt, für ein Prototyp reicht ein einfaches Kabel mit einem entsprechenden Durchmesser und Länge aus. Dennoch sollte in einem produktiven Einsatz eine richtige Antenne genutzt werden. Folgende Rechnung wurde gemacht, um eine selbstgebaute Dipolantenne zu erstellen.

$$C = 299792458 \frac{m}{s}$$

$$f = \{315MHz, 433MHz, 868MHz, 915MHz\}$$

$$\lambda = \frac{C}{f}$$

$$Länge (L) = \frac{\lambda}{2}$$

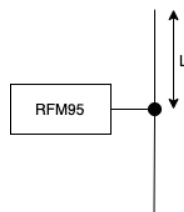


Abbildung 3 Schematischer Antennenaufbau



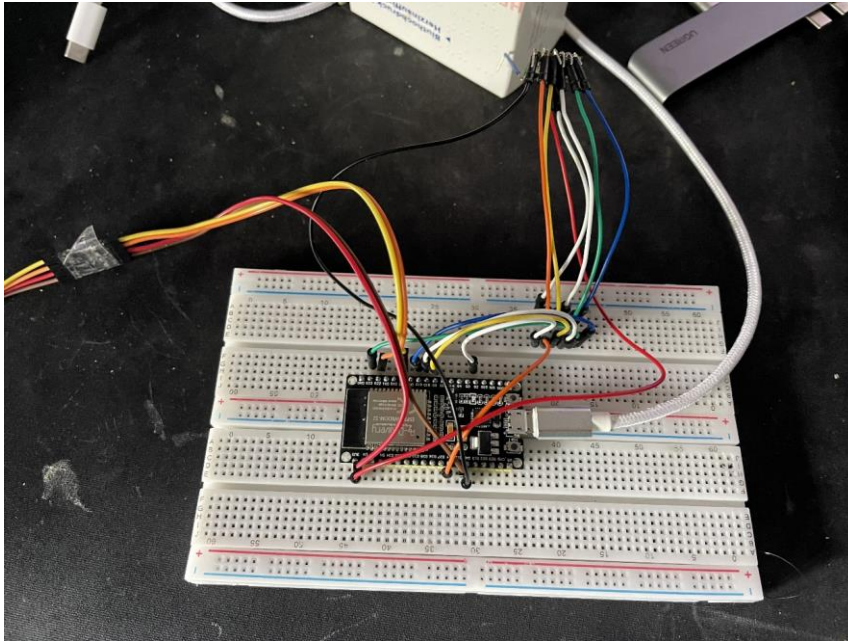


Abbildung 5 Fertiger Aufbau mithilfe eines Steckbrett

Um eine Spannungsüberwachung zu gewährleisten wird ein Spannungsteiler mit den Widerständen 47K an den GPIO Pin 34 angelegt.

#### 4.5 GPS (Optional)

Aufgrund der gegebenen Anforderungen kann eine GPS-Komponente als Optional bewertet werden weshalb diese hier gesondert in einem Abschnitt erörtert wird. Der große Vorteil an einem GPS basierten System ist die Möglichkeit das Messungsmodul dynamisch zu platzieren ohne genaue Koordinaten zu wissen, jedoch steigen damit die Kosten enorm.

Ein möglicher GPS-Chip ist der AT6558 welcher oftmals in GPS Modulen verbaut wird. Die meisten Module arbeiten mit einer seriellen Verbindung hierfür müssen Tx und Rx mit dem ESP verbunden werden.

Beispielhaft kann so ein Modul mithilfe der TinyGPS Bibliothek betrieben werden, indem die Anschlüsse Rx auf GPIO12 und Tx auf GPIO13 gelegt werden.



## 5 Kommunikation

### 5.1 Modul und Gateway

Die Kommunikation wird mithilfe der Adafruit VL530X Bibliotheken realisiert. Der zu schickende Datensatz ist ein JSON Dokument welches maximal 1024Byte groß ist.

Sobald sich entweder der Wasserstand oder Batterieentladestand verändert wird über das LoRa WAN ein Gateway benachrichtigt, das sich etwas verändert hat. Abgesehen von den zuvor genannten Veränderungen werden automatisch nach einer gewissen Zeitspanne eine Benachrichtigung an das Gateway geschickt. Die zu sendende Nachricht ist ein JSON Dokument. Nachfolgen ist das JSON Dokument kurz erläutert, genauere Informationen befinden sich im Anhang in Form eines JSON-Schemas.

1. **Device:** Meta Informationen bezüglich des Moduls
  1. **Name:** Name des Moduls
  2. **ID:** UUID des Modules
  3. **Battery:** Aktueller Entladestatus der Batterie
2. **Water:** Informationen über die Wassermessung
  1. **Max:** Maximales Fassungsvermögen
  2. **Min:** Minimales Volumen des Kanisters
  3. **Current:** Aktuelle Füllhöhe
3. **Location:** Koordinaten des Moduls
  1. **Long:** Längengrad
  2. **Lat:** Breitengrad

### 5.2 Endnutzer und Gateway

Die Kommunikation zwischen Endnutzer und Gateway ist eine Website, welche alle Informationen zu den einzelnen Modulen anzeigt, hierfür kommuniziert die Website via REST mit einem SpringBoot-Back-End. Dieses hat Zugriff auf die Datenbank und liefert die Daten als JSON-Objekt an die Website zurück. Die eingesetzte Datenbank beinhaltet ein Eintrag, basierenden auf dem JSON-Schema, zu jedem Modul.

## 6 Website für Nutzer

### 6.1 Entwurf

Zur Kommunikation der Endnutzer mit dem System wurde eine Website entwickelt. Anspruch war eine leicht verständliche und sich auf Open-Source Komponenten verlassene Website.

Entschieden wurde sich für „React“<sup>1</sup>. Hierbei handelt es sich um eine moderne JavaScript Bibliothek die 2013 von Facebook veröffentlicht wurde. Von Beginn an wurde React als Open-Source Projekt betrieben. In React lassen sich einzelnen Komponenten bauen, aus denen dann die Website besteht. Ändert sich etwas in einer Komponente, wird nur diese Komponente neu im Browser geladen, was zu hoher Performanz führt.

Die Website besteht aktuell aus einer Weltkarte, auf der mit Markern die Positionen der Geräte markiert sind. Klickt man einen der Marker an, sind weitere Informationen wie der Wasserstand, der Name, die Koordinaten sowie den Ladezustand.

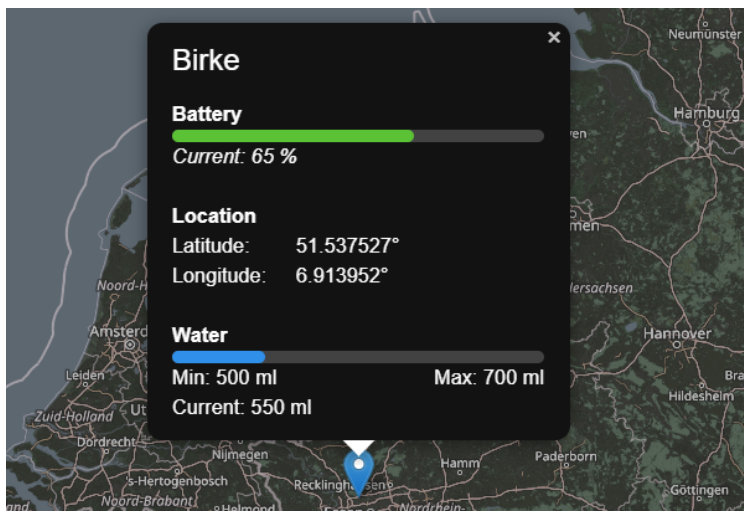


Abbildung 6 Detailsansicht eines Geräts

Weiterhin sollte es möglich sein, Geräte hinzufügen, editieren und löschen zu können. Wir haben uns dafür entschieden diese Funktionen hinter einem Log-In zu verstecken, damit nicht jeder Nutzer Geräte manipulieren kann. Ebenfalls ist es möglich die Geräte

<sup>1</sup> <https://reactjs.org/>

auf der Seite nach Batteriestand und leerem Wassertank zu filtern. So kann man schnell die Anzahl der Geräte, die in einem kritischen Zustand sind, auf einen kleinen Kreis reduzieren.

## 6.2 Funktionen

Im Folgenden werden anhand von Bildern die verschiedenen Funktionen der Website vorgestellt. Die Bilder sind alle aus Sicht eines Administrators entstanden.

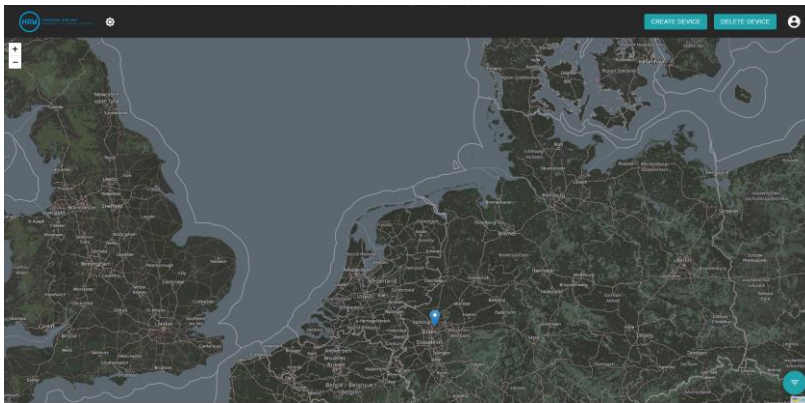


Abbildung 7 Website zur Darstellung der Geräte

Der Marker ist bereits oben kurz vorgestellt worden, in der Administrator-Sicht enthält dieser noch zusätzlich die UUID des Gerätes. Sichtbar ist zusätzlich noch ein Stift-Symbol, über welchen sich das Gerät editieren lässt.

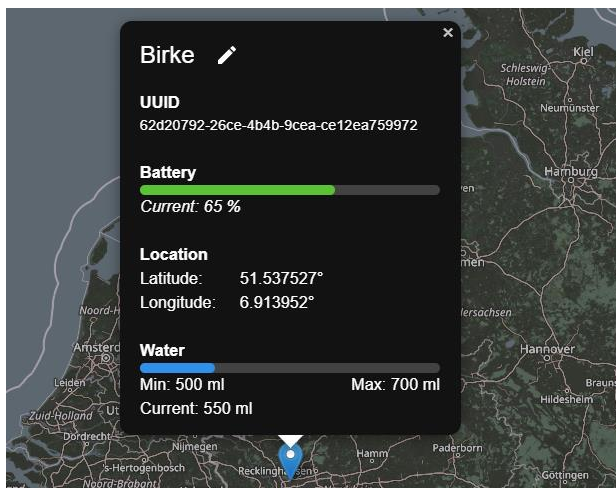


Abbildung 8 Website Marker für ein Geräte (Administrator Ansicht)

Über das Banner am Kopf der Website lässt sich ein neues Gerät erstellen. Der Nutzer wird hier zur Eingabe eines Namens, einer Mindest- sowie Maximalmenge an Wasser gebeten. Der Batteriestand ist nicht pflegbar sondern wird vom Gerät selbst gemeldet, genauso wie der aktuelle Wasserstand. Bei den Koordinaten haben wir uns schlussendlich ebenfalls dafür entschieden, die Pflegefunktion nicht anzubieten. Das Gerät meldet auch hier selbstständig seinen Standort an das Back-End. Der Grund hierfür ist, dass nicht jeder die Aufteilung der Koordinaten in Längen- und Breitengraden kennt, wir aber eine leicht verständliche Benutzeroberfläche entwickeln wollten. Weiterhin führt das Vertauschen von Längen- und Breitengraden zu gravierenden Fehlern. Ebenfalls wird so der Manipulation von Daten vorgebeugt. Nachdem das Gerät erfolgreich angelegt wurde, wird eine „main.ino“-Datei zurückgegeben. Diese enthält den notwendigen Quellcode, um den Microcontroller in das System anzuschließen. Der Quellcode muss also vom Nutzer auf den Microcontroller übertragen werden und kann dann anschließend an seine angedachte Position gebracht werden.

Abbildung 9 Erstellung eines Gerätes

Editieren lassen sich dann ebenfalls nur der Name, die Mindest- und Maximalkapazität.

Abbildung 10 Bearbeiten eines Gerätes

Löschen lassen sich die Geräte über deren UUID. Aus diesem Grund ist auch die UUID in der Administrator Ansicht sichtbar, da sonst natürlich nicht herauszufinden ist, welche UUID die Geräte haben

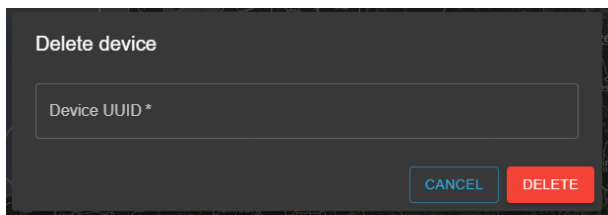


Abbildung 11 Löschen eines Gerätes

### 6.3 Implementierung

Dieser Abschnitt beschreibt die Implementierung der Website, das heißt mit welchen Standards, Bibliotheken ist aus welchem Grund gearbeitet worden. Die Struktur des Codes wird grundlegend beschrieben.

Im Kern der Anwendung steht die Karte, auf der die Geräte dargestellt werden können. Unsere erste Wahl fiel hier auf Google Maps. Wir haben uns dann aber dagegen entschieden, da Google Maps nicht Open-Source nutzbar ist und daher nicht als praktikabel erschien. Unsere abschließende Wahl fiel dann auf „React Leaflet“<sup>2</sup> einen Wrapper für React von der JavaScript Bibliothek „Leaflet“<sup>3</sup>. Bei Leaflet handelt es sich um eine Open-Source entwickelte Bibliothek die eine weltweite Karte sowie einigen Funktionen wie zum Beispiel Marker enthält. Das Kartenmaterial stammt von „OpenStreetMap“<sup>4</sup>, einem weltweiten Projekt zur Bereitstellung einer freien Weltkarte.

Für das Styling der Komponenten haben wir ebenfalls eine Bibliothek mit fertigen Komponenten verwendet. Hier haben wir uns für das Open-Source verfügbare „Material UI“ entschieden. Material UI enthält eine Vielzahl von fertigen Komponenten, die an das Design von Google angelehnt sind.

Zur Implementierung wurde die Programmiersprache TypeScript verwendet, die auf JavaScript aufbaut und diese mit Typisierung erweitert. Die Anwendung enthält eine Datei „api.ts“, welche die Kommunikation mit dem SpringBoot-Back-End übernimmt. Die Anwendung kommuniziert mit dem Back-End über eine REST-Schnittstelle. Verwendet wird hierzu JSON, um die Objekte zwischen den beiden Anwendungen zu

---

<sup>2</sup> <https://react-leaflet.js.org/>

<sup>3</sup> <https://leafletjs.com/>

<sup>4</sup> <https://www.openstreetmap.org/>

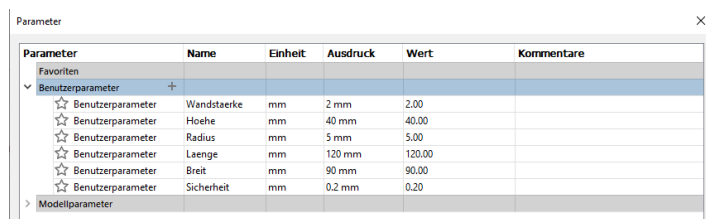
übertragen. Eine besondere Funktion übernehmen die drei Dateien im Ordner „contexts“. Die Kommunikation zwischen den Schichten erfolgt in React normalerweise nur zwischen Kindern und ihren Eltern-Komponenten. Um Daten nicht durch die gesamte Anwendung durchreichen zu müssen, lassen sich „Contexts“ verwenden. Diese sind von allen Komponenten verwendbar und ermöglichen eine schlankere und vereinfachte Kommunikation. In dieser Anwendung sind von diesen insgesamt drei verbaut, einer für das Design der Anwendung. Darüber wird der Wechsel zwischen einem hellen und dunklen Modus gesteuert. Ein weiterer verwaltet die gerade geladenen Geräte und bietet Funktionen zur Filterung und neu laden an. Ein dritter verwaltet den aktuellen Status des Nutzers, das bedeutet, ob dieser angemeldet ist oder nicht. Über diesen ist auch die Funktion zum Log-In und Log-Out zugänglich.

## 7 3D-Druck des Gehäuses

### 7.1 Entwurf

Zur Befestigung und Anbringung der Komponenten wurde ein Gehäuse konstruiert und mittels 3D-Druck hergestellt. Das Gehäuse hat die Aufgabe, die technischen Komponenten vor Wasser zu schützen und auch den Sensor zur Messung des Füllstandes zu fixieren. So ist sichergestellt, dass der Sensor immer eine feste Position innerhalb des Tankes hat und konstante Messergebnisse geliefert werden können.

Zur Konstruktion des Gehäuses wurde die Software „Fusion 360“ verwendet. Diese wurde 2013 von Autodesk vorgestellt und löst damit die Software Inventor Fusion ab und ist unter anderem auf additive Fertigungsmethoden wie den 3D-Druck optimiert. Ein großer Vorteil sind die Abhängigkeiten, welche während der Konstruktion angelegt werden können. Zusammen mit der Möglichkeit eigene Parameter mit Werten anzulegen, ermöglichen sie es ein möglichst flexibles Design zu erstellen. So können einfach der Wert von bestimmten Parametern geändert werden und das gesamte Design passt sich automatisch an, sodass keine händischen Korrekturen gemacht werden müssen.



| Parameter         | Name       | Einheit | Ausdruck | Wert   | Kommentare |
|-------------------|------------|---------|----------|--------|------------|
| Favoriten         |            |         |          |        |            |
| Benutzerparameter |            |         |          |        |            |
| Benutzerparameter | Wandstärke | mm      | 2 mm     | 2.00   |            |
| Benutzerparameter | Höhe       | mm      | 40 mm    | 40.00  |            |
| Benutzerparameter | Radius     | mm      | 5 mm     | 5.00   |            |
| Benutzerparameter | Laenge     | mm      | 120 mm   | 120.00 |            |
| Benutzerparameter | Breit      | mm      | 90 mm    | 90.00  |            |
| Benutzerparameter | Sicherheit | mm      | 0.2 mm   | 0.20   |            |
| Modellparameter   |            |         |          |        |            |

Abbildung 12 Benutzerdefinierte Parameter in Fusion 360

### 7.2 Design

Das Design des Gehäuses wurde in zwei Teile unterteilt: die Halterung sowie das eigentliche Gehäuse. Damit die technischen Komponenten gut erreicht werden können, wird das eigentliche Gehäuse an einer Halterung, welche fest am Tank befestigt ist, angebracht. So können technische Komponenten besser erreicht oder ausgetauscht werden.

Das Gehäuse hat eine Halterung für den Lasersensor, welcher den Füllstand messen soll. Um diesen zu fixieren, wurden an die Positionen der Schraublöcher des Sensors entsprechende Gewinde gedruckt. Ebenfalls hat das Gehäuse eine Aussparung nach unten, damit der Sensor arbeiten kann.

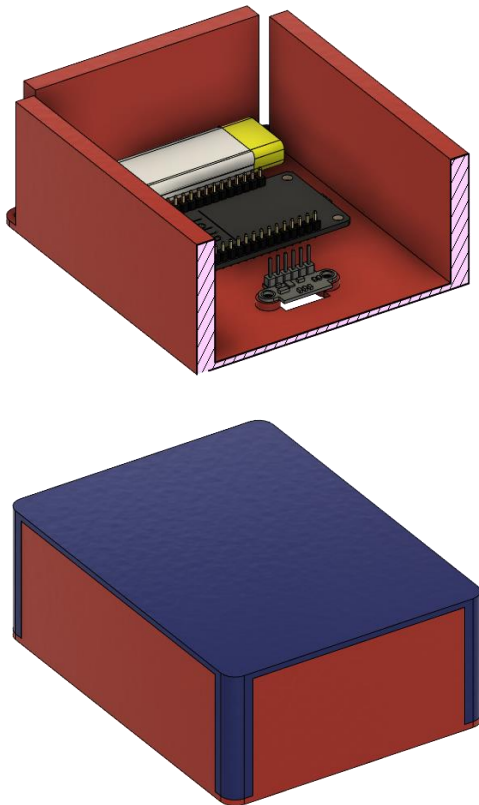


Abbildung 13 Schnittanalyse des Gehäuses mit schemenhafter Darstellung der Komponenten

Der Halter wird mit Schrauben am Tank befestigt, und hat vier Beine, auf die das Gehäuse gesteckt werden kann. Zur Fixierung dient jeweils eine Schraube an jedem Bein.

Abbildung 14 Gehäuse (rot dargestellt) mit Halter (blau dargestellt)



Für diesen Prototypen wurden die Bemaßung noch nicht optimiert, damit die Auswahl der Komponenten sowie die Verkabelung innerhalb des Gehäuses nicht eingeschränkt wird. Das Gehäuse des Prototyps ist 120mm lang, 90mm breit und 40mm hoch. Hier kann durch die Auswahl von entsprechend kleineren Bauteilen die Größe noch optimiert werden.

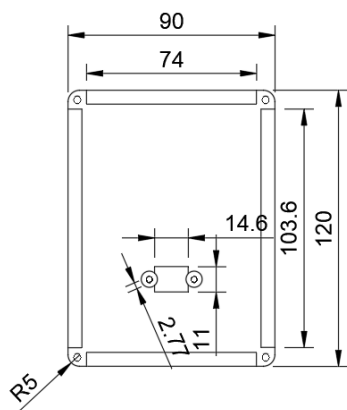


Abbildung 15 Bemaßung des Gehäuses

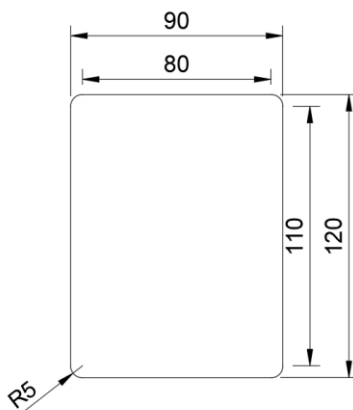


Abbildung 16 Bemaßung des Halters

### 7.3 Slicing

Beim Slicing wird aus der von Fusion 360 generierten 3D-Datei eine sogenannte G-Code Datei erzeugt. Eine G-Code Datei enthält Befehle für den 3D-Drucker wie beispielsweise Koordinaten, welche der Druckkopf als nächstes anfahren soll.

Da es sich bei dem Drucker um einen „Prusa Mini Plus“ handelt, kam die Slicer-Software „Prusa Slicer“ zum Einsatz. Hier wurden größtenteils die Voreinstellungen verwendet. Eine Ausnahme bildet dabei die Einstellung „Infill“. Sie gibt an, wie hohl oder massiv ein Körper gedruckt wird. Da das Gehäuse möglicherweise mit Spritzwasser in Kontakt kommen kann, wurde der Infill für das Gehäuse auf 100% gesetzt. So soll verhindert werden, dass Wasser in das Gehäuse eindringt und technische Komponenten beschädigt.

Folgende Einstellungen wurden verwendet:

| <b>Einstellung</b>  | <b>Halter</b>            | <b>Gehäuse</b>            |
|---------------------|--------------------------|---------------------------|
| Düsendurchmesser    | 0,4mm                    | 0,4mm                     |
| Stützen             | Keine                    | Keine                     |
| Infill              | 15%                      | 100%                      |
| Filament            | PLA (Durchmesser 1,75mm) | PLA (Durchmesser 1,75 mm) |
| Schichthöhe         | 0,2mm                    | 0,2mm                     |
| Düsentemperatur     | 205° C                   | 205° C                    |
| Druckbetttemperatur | 60° C                    | 60° C                     |



Abbildung 17 Gedruckter Halter



Abbildung 18 Gedrucktes Gehäuse

## 8 Messungen der einzelnen Module

### 8.1 Wasserpegel

Das genutzte Messsystem sieht eine gewisse Anzahl von Einzelmessung vor, jede Einzelmessung wird zehn Millisekunden nach der letzten durchgeführt. Nachdem eine fest definierte Anzahl Messungen durchgeführt wurde, wird der Mittelwert gebildet, dieser Mittelwert wird dann als realer Abstand zwischen Sensor und Wasserstand genommen.

Der schwerste Teil der Pegelmessung bildet das Abbilden zwischen Messwert und Wasseraufbewahrungsbehälter. Allgemein betrachtet sind sowohl die Höhe als auch Form des Kanisters bekannt, mithilfe dieser Werte kann das maximale Volumen berechnet werden. Wenn das Maximum bekannt ist, kann eine Abbildung in Prozente realisiert werden. Für die Abbildung kann die map Funktion genutzt werden, als Argumente werden aktuelle Füllhöhe, minimaler sowie maximaler Wasserstand und die Prozentgrenzen übergeben.

Das bereitgestellte Programm sieht einen rechteckigen Kanister vor. Für spezielle Anforderungen an das Modell muss eine Messung der Grenzen ausgeführt werden, ansonsten gibt es fest definierte Maße. Die gemessenen Grenzen bilden dann die minimale sowie maximale Füllhöhe.

### 8.2 Batterie

Der anliegende LiPo Akku wird mithilfe eines Spannungsteilers gemessen werden.

Da das Messergebnis zwischen 0 und 4096 liegt wird dieses mithilfe der map funktion in ein Bereich von 0 bis 210 abgebildet, 210 ist hier die Hälfte der maximalen Spannung des LiPo Akku von 4.2V. Da das neu abgebildete nur die Hälfte Tatsächlichen Spannung entspricht muss diese in einem Zwischenschritt umgewandelt werden. Der abschließende Schritt ist es die Spannung in ein Prozentbereich abzubilden, hierfür wird ein einfacher Dreisatz angewandt der die maximale Spannung als hundert Prozent betrachtet.

## 9 Fazit

Abschließend wollen wir noch ein Fazit ziehen, ob unser definiertes Ziel erreicht worden ist und einen Ausblick über mögliche Verbesserungen oder Erweiterungen geben.

Abschließend ist es uns gelungen, sowohl einen funktionierenden Hardwareprototypen als auch eine fertige Website zu entwickeln. Unser Ziel die Anwendung für die Nutzer zugänglich zu machen und diesen Zugang aus so einfach wie möglich zu gestalten ist uns damit gut gelungen. Auch das übergeordnete Ziel, die Entwicklung des Prototyps konnten wir erfolgreich abschließen.

Was in Zukunft noch verbessert werden kann ist zum Beispiel das Anlegen eines neuen Geräts über die Website. Das Herunterladen einer Datei mit Quellcode ist etwas unglücklich und lässt sich vielleicht mit einer angegebenen drahtlosen Verbindung oder auch einer USB-Verbindung zum Computer einfacher und nutzerfreundlicher lösen. Weiterhin ist die Website noch um neue Funktionen erweiterbar, die den Komfort erhöhen können, wie das Einlesen von XML-Dateien zu Massenanlegung von Geräten, oder zum Löschen von einer Vielzahl von Geräten.

Die einzelnen Teile des Prototyps können nach einigen Testläufen in realitätsnahen nochmals einer genauen Prüfung unterzogen werden, ob diese sich so für den produktiven Gebrauch eignen. Module wie das GPS Modul sollten in einer genauen Kosten-Nutzen-Analyse geprüft werden. Je nach Verkaufspreis ist dies eventuell nicht rentabel und man schwenkt um auf eine genaue Angabe der Koordinaten bei Erstellung des Gerätes.

Im Nachhinein sind wir der Meinung, dass das Projekt erfolgreich gelaufen ist und das Startup mit dem von uns entwickelten Prototyp viel anfangen und darauf aufbauen kann.

## Literaturverzeichnis

- az-delivery. (05 2022). *www.az-delivery.de*. Von <https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/vl53l0x-time-of-flight-tof-laser-abstandssensor-teil-2> abgerufen
- az-delivery. (05 2022). *www.az-delivery.de*. Von <https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/vl53l0x-time-of-flight-tof-laser-abstandssensor-teil-1> abgerufen
- blues.io. (30. 05 2022). *blues.io*. Von <https://blues.io/blog/network-connectivity/> abgerufen
- Müller, I. (22. August 2022). *Fassadenbegrünung an Gladbecker Straße in Essen vertrocknet*. Von <https://www.waz.de/staedte/essen/essen-fassadenbegrueunung-an-gladbecker-strasse-vertrocknet-id236197969.html> abgerufen
- Ovcharov, A. (05 2022). *en.ovcharov.me*. Von <https://en.ovcharov.me/2020/02/29/how-to-measure-battery-level-with-esp32-microcontroller/> abgerufen
- pangodream. (05 2022). *www.pangodream.es*. Von <https://www.pangodream.es/esp32-getting-battery-charging-level> abgerufen
- randomnerdtutorials.com. (05 2022). Von <https://randomnerdtutorials.com/esp32-lora-rfm95-transceiver-arduino-ide/> abgerufen
- mtlab.com. (05 2022). Von <https://mtlab.com/question/is-it-possible-to-power-the-esp32-directly-with-a-cr123-battery/> abgerufen
- Sparkfun. (05 2022). *cdn.sparkfun.com*. Von [https://cdn.sparkfun.com/assets/learn\\_tutorials/8/0/4/RFM95\\_96\\_97\\_98W.pdf](https://cdn.sparkfun.com/assets/learn_tutorials/8/0/4/RFM95_96_97_98W.pdf) abgerufen
- STMicroelectronics. (05 2022). *www.st.com*. Von [https://www.st.com/resource/en/user\\_manual/um2039-world-smallest-timeofflight-ranging-and-gesture-detection-sensor-application-programming-interface-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um2039-world-smallest-timeofflight-ranging-and-gesture-detection-sensor-application-programming-interface-stmicroelectronics.pdf) abgerufen

Kommentiert [NM1]:

## Anhang

Anhang A: *Sourcecode ESP32*

Anhang B: *JSON Kommunikationsschema*

Anhang C: *Sourcecode Website*

Anhang D: *GPS Beispielcode*

## **Kenntlichmachung der Eigenleistungen**

Hier ist aufgeführt, wer welche Seiten gestaltet und geschrieben hat, um Eigenleistungen kenntlich zu machen.

Alex Heiming

Kapitel 1: Einleitung (Seite 7)

Kapitel 2: Ziele (Seite 8)

Kapitel 6: Website für Nutzer (Seiten 18 bis 22)

Kapitel 8: Fazit (Seite 24)

Lars Dubrau

Kapitel 7: 3D-Druck des Gehäuses (Seiten 23 bis 26)

Nils Milewski

Kapitel 3: Stand der Technik (Seiten 9 bis 11)

Kapitel 4: Komponenten (Seiten 12 bis 16)

Kapitel 5: Kommunikation (Seite 17)

Kapitel 8: Messung der einzelnen Module (Seite 27)

## Erklärungen

Hiermit erkläre ich, dass ich die vorliegende Abschlussarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Die vorgelegte Arbeit hat weder in der gegenwärtigen noch in einer anderen Fassung schon einem anderen Fachbereich der Hochschule Ruhr West oder einer anderen wissenschaftlichen Hochschule vorgelegen.

Bottrop, 28.09.2022

Ort, Datum



Unterschrift

Hiermit erkläre ich, dass ich die vorliegende Abschlussarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Die vorgelegte Arbeit hat weder in der gegenwärtigen noch in einer anderen Fassung schon einem anderen Fachbereich der Hochschule Ruhr West oder einer anderen wissenschaftlichen Hochschule vorgelegen.

Wesel, 29.09.2022

Ort, Datum



Unterschrift

Hiermit erkläre ich, dass ich die vorliegende Abschlussarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Die vorgelegte Arbeit hat weder in der gegenwärtigen noch in einer anderen Fassung schon einem anderen Fachbereich der Hochschule Ruhr West oder einer anderen wissenschaftlichen Hochschule vorgelegen.

Essen, 29.09.2022

Ort, Datum



Unterschrift