

Technische Spezifikation

Smarte Gartenbewässerung über LoRaWAN

eine insgesamt sehr gute technische Spezifikation. Die Pin-Belegung im Schaltplan sowie die Bemaßung des Gehäuses für den Ultraschallsensor fehlen.
weitere kleine Anmerkungen siehe unten.

Wertung:

TechSpec : 5 Punkte

Projektplan : 5 Punkte

Mitarbeiter und Autoren:

- Rami Hammouda
- Khac Hoa Le
- Jaro Machnow

Letzte Änderung: 21.07.2021

Version: 1.5

Inhaltsverzeichnis

| | |
|--|-----------|
| 1. Einleitung | 4 |
| 1.1 Überblick | 4 |
| 1.2 Definitionen und Abkürzungen | 4 |
| 1.3 Vorhandene Dokumente | 4 |
| 2. Prozessüberblick | 4 |
| 2.1 Realisierungsprozess | 5 |
| 2.2 Fachlicher Workflow | 7 |
| 2.3 Checkliste | 8 |
| 3. Systemarchitektur und Infrastruktur | 10 |
| 3.1 Systemarchitektur | 10 |
| 3.2 Beschreibung der Komponenten | 11 |
| 4. Technische Spezifikation der Software | 12 |
| 4.1 Überblick Software-Komponenten | 12 |
| 4.2 Schnittstellen zwischen den Komponenten | 14 |
| 4.3 Technologiestack | 15 |
| 4.4 Anmeldedaten für TTN | 16 |
| 4.5 Erste Testphase - Senden und Empfangen von Daten über TTN | 17 |
| 4.6 Entwicklung der User Interface (UI) | 23 |
| 4.7 Integration von Telegram | 25 |
| 4.8 Lora32-Komponente: Klassendiagramm | 26 |
| 4.9 Autonome Steuerung | 27 |
| 4.10 Fehlererkennung und -behandlung | 29 |
| 5. Spezifikation der Hardware | 30 |
| 5.1 Einzelteile | 30 |
| 5.2 Schaltplan | 35 |
| 5.3 Halterung für den Ultraschallsensor | 37 |
| 5.4 Gesamtaufbau | 39 |

Versionshistorie

| Version | Datum | Verantwortlich | Änderung |
|---------|------------|--------------------|------------------------------|
| 1.0 | 29.05.2021 | Jaro Machnow | Dokumenterstellung |
| 1.1 | 06.06.2021 | Jaro, Hoa Le, Rami | Ergänzungen |
| 1.2 | 15.06.2021 | Jaro, Hoa Le, Rami | Ergänzungen |
| 1.3 | 16.06.2021 | Rami, Hoa, Jaro | Vervollständigung Sprint 1 |
| 1.4 | 15.07.2021 | Jaro, Hoa Le, Rami | Ergänzungen |
| 1.5 | 21.07.2021 | Jaro, Hoa Le, Rami | Vervollständigungen Sprint 3 |

1. Einleitung

1.1 Überblick

Es wird eine smarte und möglichst preisgünstige und überwachte Bewässerung von Beeten per Netzwerksteuerung über das *The-Things-Network* (TTN) gebaut. Im Urban Garden kontrolliert ein Mikrocontroller entsprechend der über das Netzwerk gesendeten Nutzereingaben oder eines autonomen Workflows verschiedene Sensoren und Aktoren. Mit Hilfe einer LoRaWAN-Antenne werden die Daten der Sensoren vom Urban Garden aus in das Netzwerk gesendet und können auf einem entfernten Computer ausgelesen werden und als Visualisierung auf *OpenSenseMap.org* angesehen werden.

Zu den Mechanismen der autonomen Steuerung gehören vor allem das Abschalten des Wasserflusses im Falle eines Lecks und das eigenständige Starten der Bewässerung beim Erreichen einer bestimmten Bodenfeuchte.

Für alle eingebauten Komponenten und Funktionen gilt der Grundsatz, dass das System in Zukunft leicht modular erweitert werden kann.

1.2 Definitionen und Abkürzungen

LoRaWan -Long Range Wide Area Network

TTN - The Things Network

1.3 Vorhandene Dokumente

Tabelle 1: Vohandene Dokumente

| Dokument | Autor(en) | Datum |
|----------------------------|--|------------|
| Lastenheft | Rami Hammouda, Khac Hoa Le, Jaro Machnow | 28.04.2021 |
| Lastenheft + Kommentare | + Prof. Dr. Mohammad Abuosba | 30.04.2021 |
| Anforderung-Email | Holger Martin | 10.04.2021 |
| Pflichtenheft | Rami Hammouda, Khac Hoa Le, Jaro Machnow | 19.05.2021 |
| Pflichtenheft + Kommentare | + Prof. Dr. Mohammad Abuosba | 25.05.2021 |

2. Prozessüberblick

2.1 Realisierungsprozess

Nachfolgend ist eine Darstellung aller Funktionen und deren Unterteilung im Projekt:

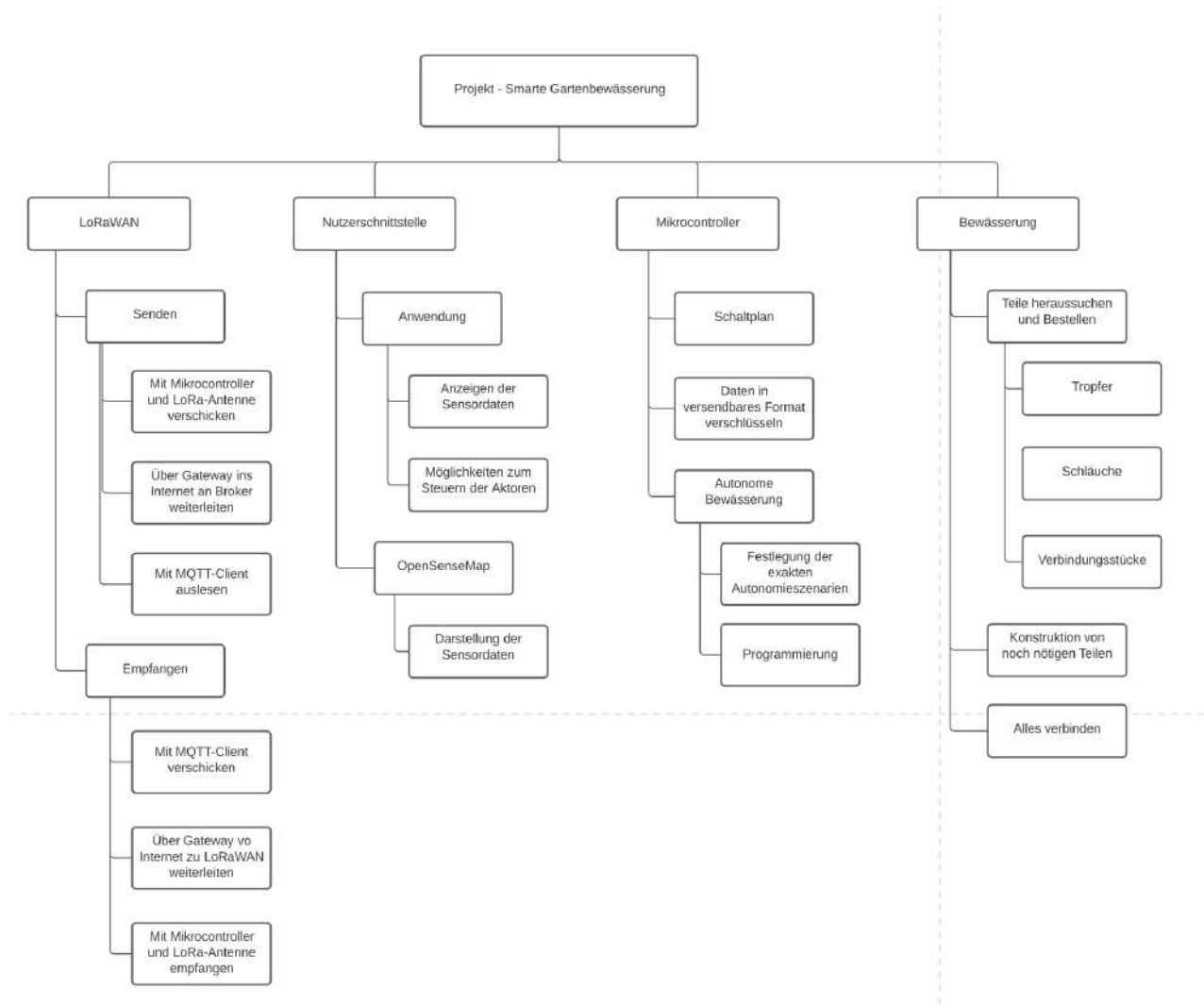


Abbildung 1: Übersicht über die Aufgabenstruktur

Für die Realisierung des Projektes ist es zuerst nötig, alle notwendigen Teile herauszusuchen und zu bestellen. Dann kann damit begonnen werden die Funktionsfähigkeit des Hauptziels des Projektes sicherzustellen: Das Senden und Empfangen von Daten vom Urban Garden über das LoRaWAN-Netzwerk. Danach können parallel die Nutzerschnittstellen mit TTN verbunden und programmiert werden und der Mikrocontroller mit den zugehörigen

Technische Spezifikation - Smarte Gartenbewässerung

Komponenten eingerichtet werden. Zum Schluss wird das gesamte Bewässerungssystem mit allen Komponenten zusammengebaut und im Urban Garden installiert.

Der Realisierungsprozess ist in folgendem Diagramm dargestellt:

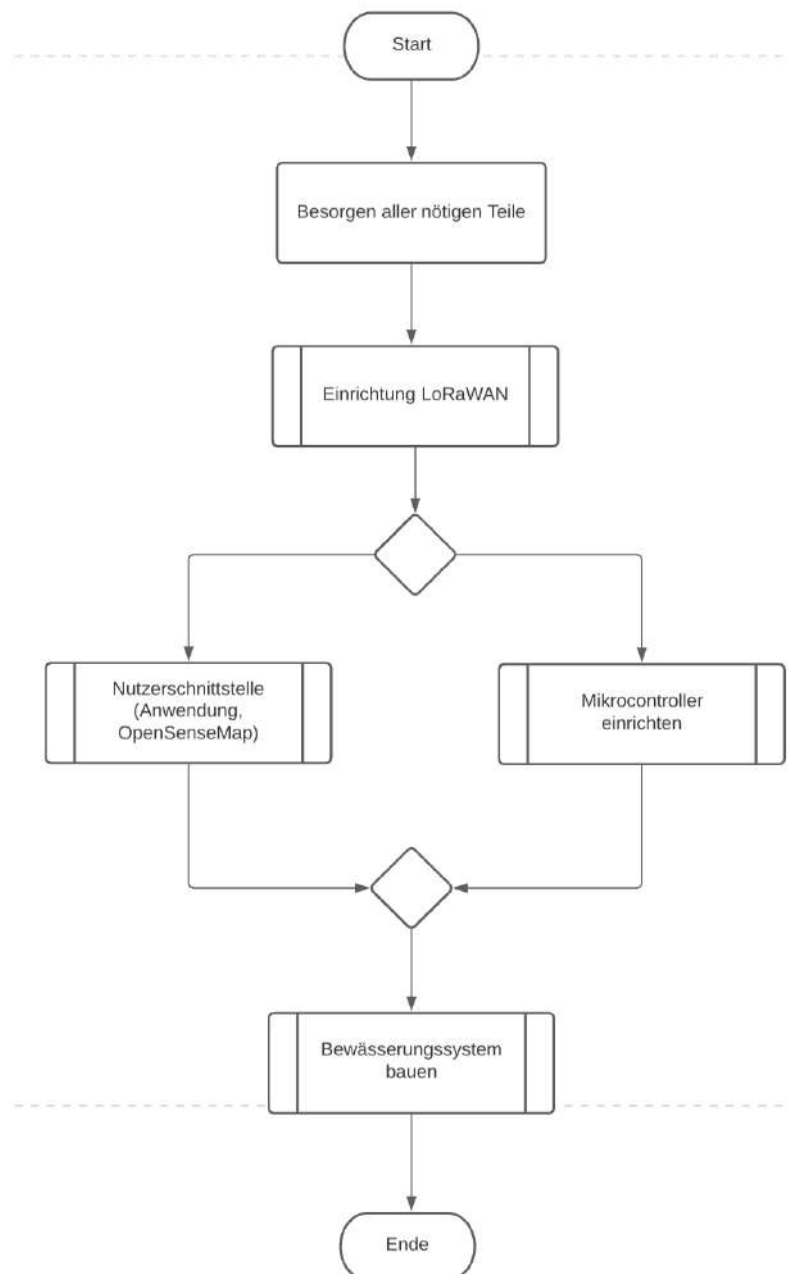


Abbildung 2: Diagramm des Realisierungsprozesses

2.2 Fachlicher Workflow

Nachfolgen ist das Diagramm des fachlichen Workflows zu sehen. Hier kann die Zusammenarbeit der Software und Hardware gesehen werden und der Ablauf bei Benutzung, Überwachung und Steuerung des Systems.

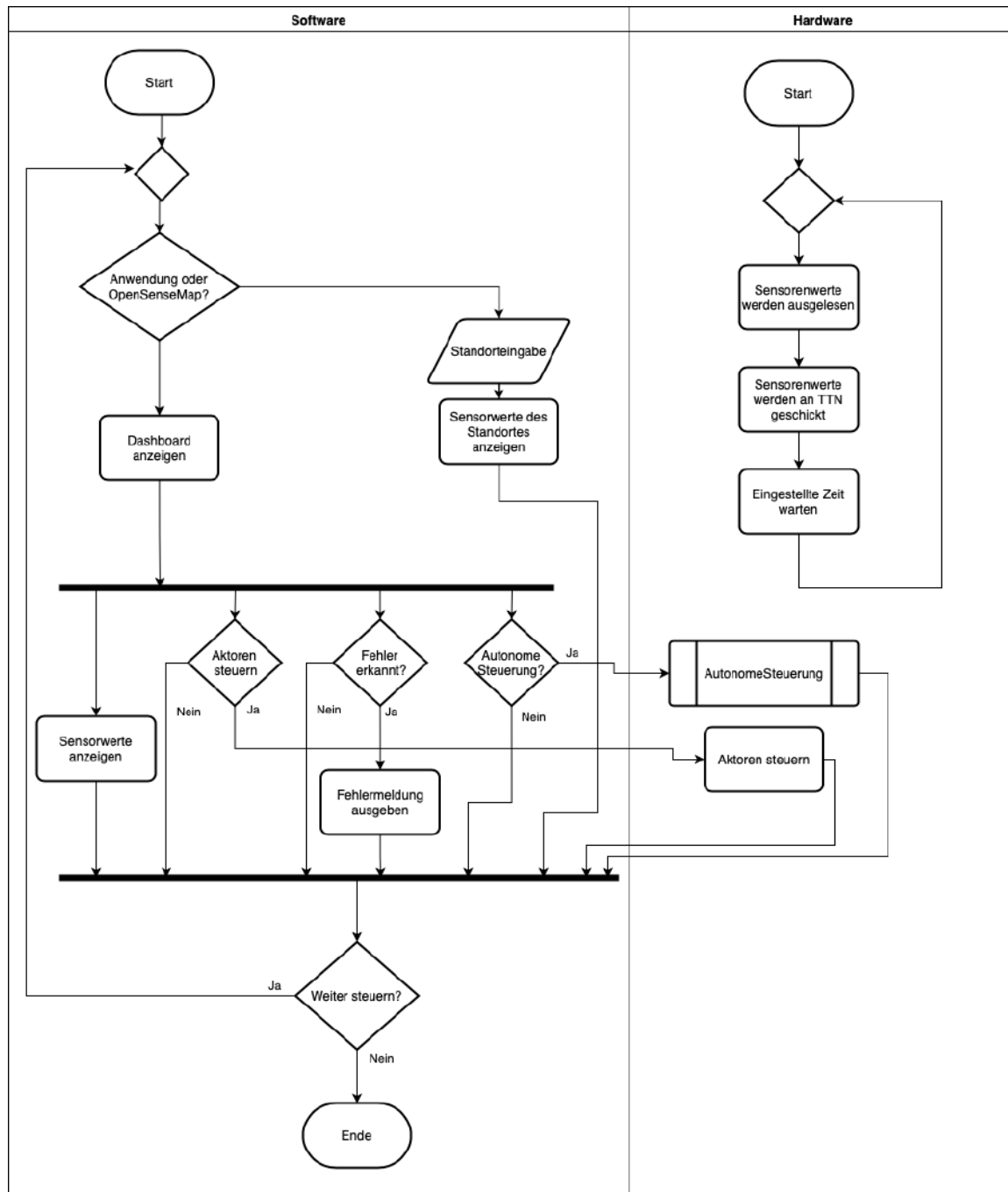


Abbildung 3: Fachlicher Workflow

2.3 Checkliste

Zur Strukturierung unserer Arbeit und um zu wissen, was schon gemacht wurde und was noch zu erledigen ist, haben wir folgende Checkliste erstellt. Die Checkliste beinhaltet aktuell nur Punkte, die mit der Software im Zusammenhang stehen (für den 1. Sprint).

Controlling Sensors, Actors:

- ☒ ~~Collect current ambient parameters (Temperatur sensor, Humidity sensor, Air pressure sensor..)~~
- ☒ ~~Collect current water level in water tank (Ultrasound sensor)~~
- ☒ ~~Collect current water hose parameters (Water pressure sensor, Water flow sensor)~~
- ☒ ~~Control Actors (Water pump, Magnetic valve)~~

On LoraWan Server:

- ☒ ~~Send collected sensor data to TTN Network~~
- ☒ ~~Operate with ABP Mode or OTAA Mode (prefer OTAA)~~
- ☒ ~~Support float decode payload~~
- ☒ ~~Support int decode payload~~
- ☒ ~~Binding Server with opensensemap.org~~

Operate Actors:

- ☒ ~~Remote control actors manually~~
- ☒ ~~Actors operates itself automatically based on pre-configuration~~
- ☒ ~~Auto Mode and Manual Mode should be selectable~~

Sensor Data Visualization:

- ☒ ~~Simple sensor data are visualized by opensensemap~~
- ☒ ~~Complex sensor data will be visualized by a self development UI~~

UI Development:

- ☒ ~~Create a UI to get all current information of system~~
- ☒ ~~Control actors through simple user-friendly UI like button, not on JSON UI (comply to requirement from Mr. Prof. Dr. Ing. Abuosba)~~
- ☒ ~~Pre-configuration parameters should be changeable (e.g through input fields) (comply to suggestion from Mr. Holger Martin)~~

Options:

- ☒ ~~An UI on mobile for convenient monitoring~~
- ☒ ~~Send Notification to user for alert~~

3. Systemarchitektur und Infrastruktur

3.1 Systemarchitektur

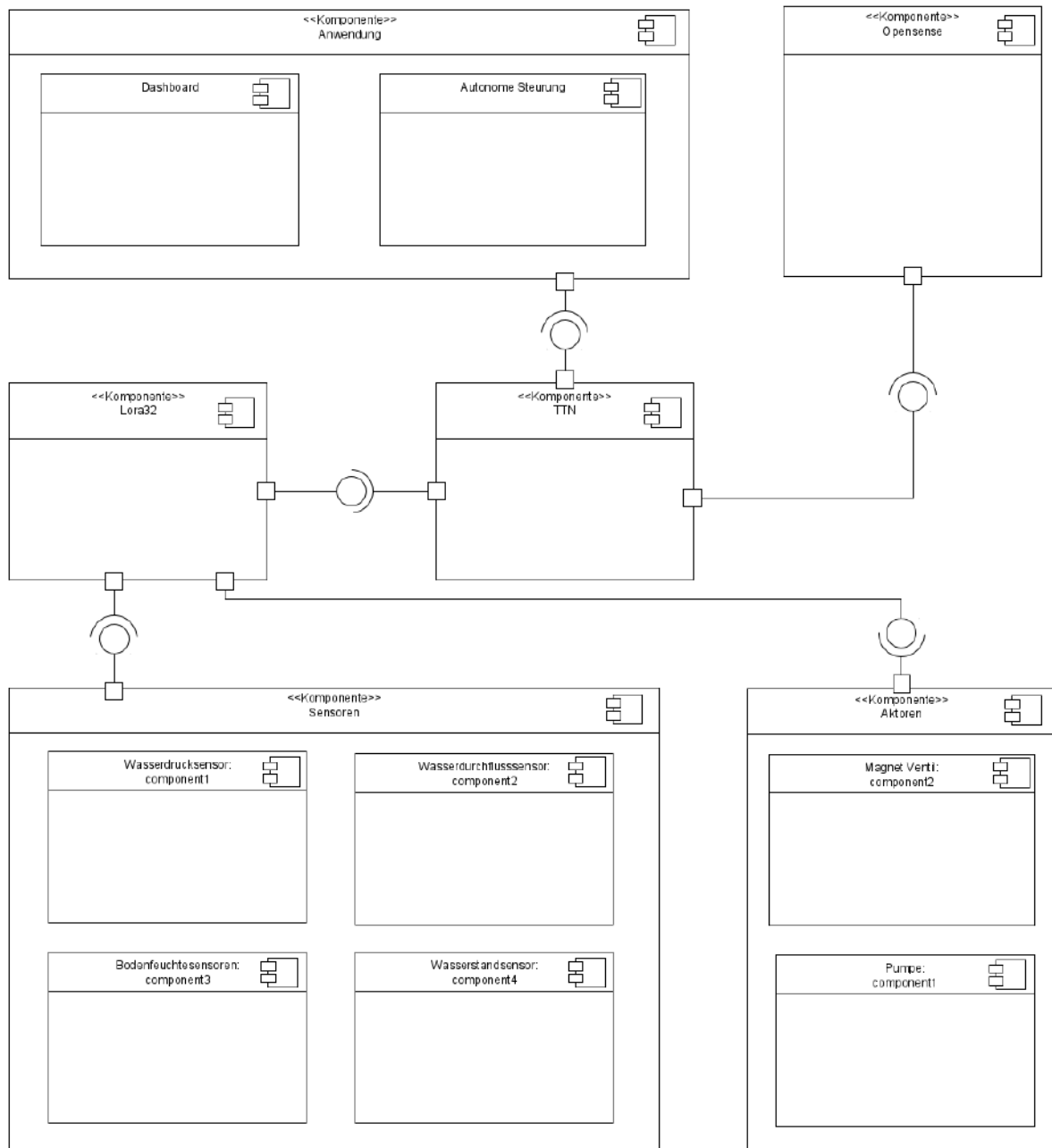


Abbildung 4: Komponentendiagramm

3.2 Beschreibung der Komponenten

Das Komponentendiagramm zeigt die verschiedenen Schichten des Bewässerungssystems:

- Ganz oben sind die Komponenten, mit denen der Nutzer interagieren kann. Dazu gehören die Anwendung und die OpenSenseMap. Bei der Anwendung sieht der Nutzer das Dashboard, in dem Sensordaten angezeigt werden können und Aktoren per Knopfdruck aktiviert oder deaktiviert werden können. Zudem kann die autonome Steuerung aktiviert werden. Die OpenSenseMap zeigt eine Karte und die Sensordaten für den Standort des Urban Gardens.
- In der Mitte befinden sich Komponenten, die die obere Anwendungsschicht mit den Sensoren und Aktoren im Urban Garden verbinden. Dazu gehören das TheThingsNetwork und der Lora32-Mikrocontroller. Über diese beiden Daten kommen die Sensordaten zur Nutzeranwendung und die Nutzereingaben zu den Aktoren. Die Kommunikation bzw. Übertragung der Daten erfolgt hier über das LoRaWan-Netzwerk und über das Internet. Nähere Informationen sind dazu in Punkt 4.2 nachzulesen.
- Unten befinden sich die Komponenten, die direkt im Urban Garden agieren, also alle Sensoren und Aktoren. Dazu gehören: Wasserdrucksensor, Wasserdurchflusssensor, Wasserstandsensor, Bodenfeuchtigkeitssensor, Pumpe und Magnetventil. Diese Komponenten sind mit Lora32 verbunden und werden dadurch überwacht und gesteuert.

4. Technische Spezifikation der Software

4.1 Überblick Software-Komponenten

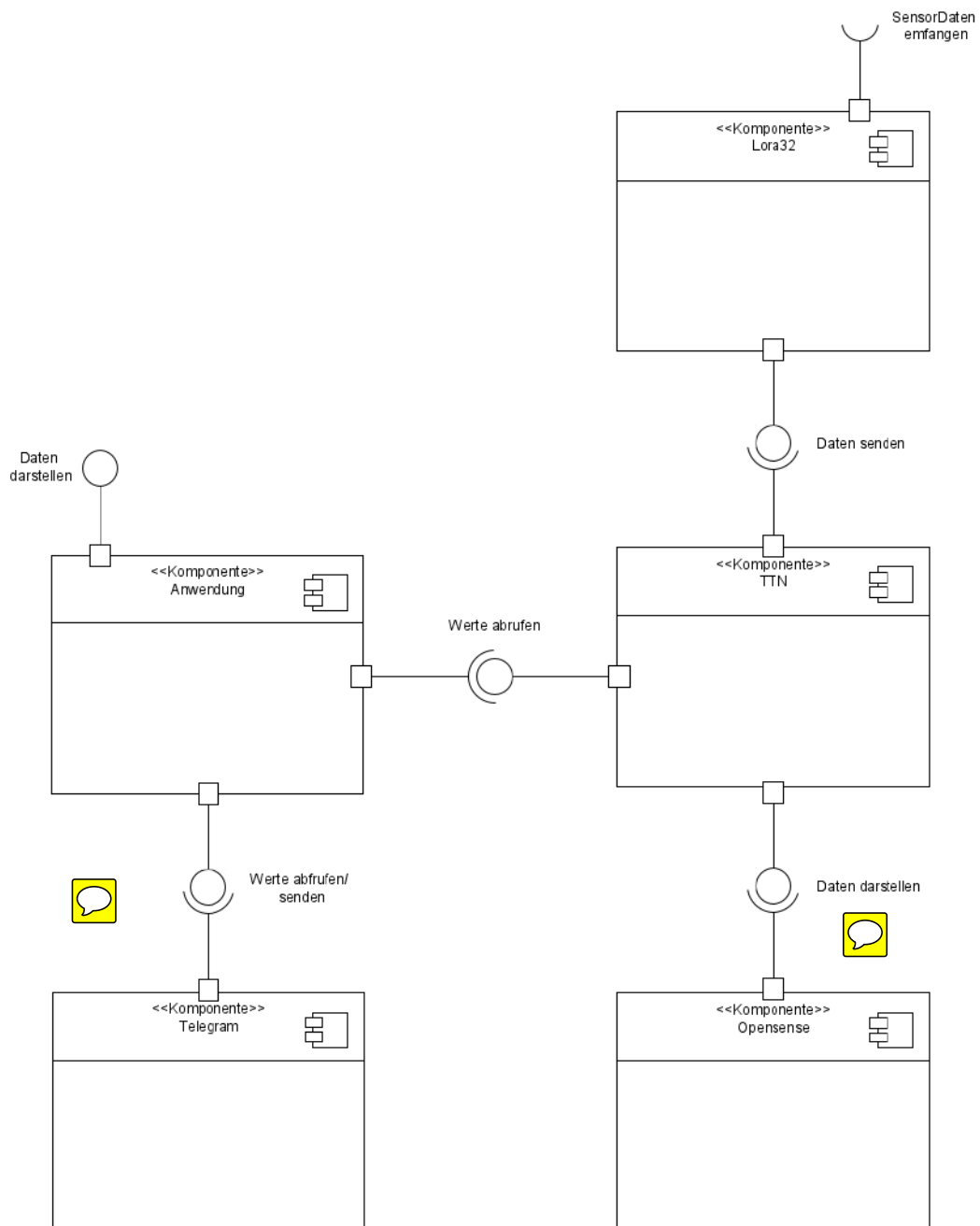


Abbildung 5: Komponentendiagramm Software

Technische Spezifikation - Smarte Gartenbewässerung

Beschreibung der Software-Komponenten:

Tabelle 2: Software-Komponenten

| SW Komponente | Funktionen | Sprache/Typ |
|---|---|----------------------------|
| Anwendung (geplant)  | <ul style="list-style-type: none"> - Daten empfangen und anzeigen - Daten auswerten - Daten senden - Alles wird in einem Dashboard angezeigt | JavaScript (Node Js) |
| Opensensemap (geplant) | <ul style="list-style-type: none"> - Anzeigen der Messdaten auf einer Karte | Website Service als Client |
| Telegram (geplant) | <ul style="list-style-type: none"> - Schnell und aktuell die Daten und den Status vom System abrufen | API Service als Client |
| Lora32 (Software) | <ul style="list-style-type: none"> - Sensorendaten abrufen - Aktoren steuern - Signal schicken/empfangen. | Programming language C++ |
| TTN | <ul style="list-style-type: none"> - Zentraler Server für LoRaWAN - Schnittstelle zwischen LoRaWAN und Client. - Datenbank: Automatische Speicherung der Daten einer Woche | Website Server |

4.2 Schnittstellen zwischen den Komponenten

Die Datenübermittlung zwischen dem LoRa32-Mikrocontroller und der Anwendung erfolgt in mehreren Schritten:

1. LoRa32 sendet die Daten an TTN-Gateways mit Hilfe der LoRaWan-Antenne.
2. Das Gateway schickt die Daten weiter ins Internet an die TTN-Cloud durch HTTP.
3. Die Anwendung ist mit der TTN-Cloud verbunden und erhält so die Daten.

Die TTN-Cloud dient dabei zugleich als Datenbank und speichert automatisch die Daten einer Woche. Die Anwendung dient als grafische Benutzeroberfläche, die es dem Nutzer ermöglicht die Daten aus dem System auszulesen und das System fernzusteuern.

Umgekehrt, also von der Anwendung zum LoRa32, funktioniert der Vorgang analog:

1. Der Nutzer gibt Daten in der Anwendung ein. Die Daten werden an die TTN-Cloud übermittelt.
2. Die TTN-Cloud übermittelt die Daten über HTTP an das TTN-Gateway
3. Mit Hilfe der LoRaWan-Antenne werden die Daten vom Gateway an den LoRa32-Mikrocontroller gesendet.

Die OpenSenseMap auch mit der TTN-Cloud verknüpft, um Daten anzuzeigen. Das funktioniert auf die gleiche Weise, wie bei der Anwendung. OpenSenseMap dient als weitere Benutzerschnittstelle.

Zudem wird Telegram benutzt, um eine schnelle und unkomplizierte Kommunikation des Nutzers mit dem System zu ermöglichen. Telegram ist direkt mit der Anwendung verknüpft.

4.3 Technologiestack

Folgender Technologiestack wird für die Erstellung der Software für die Kommunikation über LoRaWan verwendet:

Tabelle 3: Technologiestack

| | |
|---|---|
| Operation System: | - Windows 10/(Linux) |
| IDE: | - Visual Studio Code - Platform IO |
| Framework: | - Arduino |
| Abhängigkeiten: | - MCCI LoraWAN LMIC Library v3.3 - Adafruit Unified Sensor v1.1.4 - DHT sensor library v1.4.2 |
| Protokolle: | - MQTT - HTTP |
| Drahtlose Verbindung/ Kommunikation: | - LoRaWAN (Long Range Wide Area Network) |
| Tools: | - MQTT Mosquito - MQTT Explorer |
| Server: | - TTN (thethingsnetwork) |
| UI Programmierung: | - JavaScript - NodeJS |
| Hardware Programmierung: | - C++ |
| Hardware: | - Lora32 (TTGO) |

4.4 Anmeldedaten für TTN

Mit folgenden Daten kann man sich beim TTN-Account unseres Teams anmelden, Einstellungen vornehmen und empfangene Sensordaten einsehen und Aktoren steuern:

THETHINGSNETWORK.ORG (Legacy V2 Console)

User name: htwgardenproject

Email address: htwgardenss21@gmail.com

Password: htwgarden2021

4.5 Erste Testphase - Senden und Empfangen von Daten über TTN

Bevor die Verbindung der TTN-Server mit der Anwendung, OpenSenseMap und Telegram erstellt wird, wird zunächst die Kommunikation über LoRaWan mit MQTT-Mosquito, MQTT.Fx und MQTT-Explorer getestet.

Wenn die Daten von LoRa32 sicher erzeugt werden, wird LoRa32 mit dem TTN-Server verbunden. Bei der TTN-Cloud können Uplinks (Signal von Sensoren) gelesen werden und die Downlinks (Signal zur Sensoren) geschickt werden. Das Empfangen von Sensordaten und das Schicken von Daten zur Steuerung von Aktoren werden getestet.

In der Testschaltung werden die Daten des DHT22 Temperatur- und Luftfeuchtigkeitssensors vom Lora32 erfasst. Dann werden die Daten weiter an TNN gesendet. Bei jedem Senden leuchtet die grüne LED. Zusätzlich kann ein Signal von TTN aus gesendet werden um die rote LED an- bzw. auszuschalten.

Bei dem ersten Test stand uns ein Lora32 v1 zur Verfügung. Ein Bild dieser Schaltung ist in nachfolgender Abbildung zu sehen:

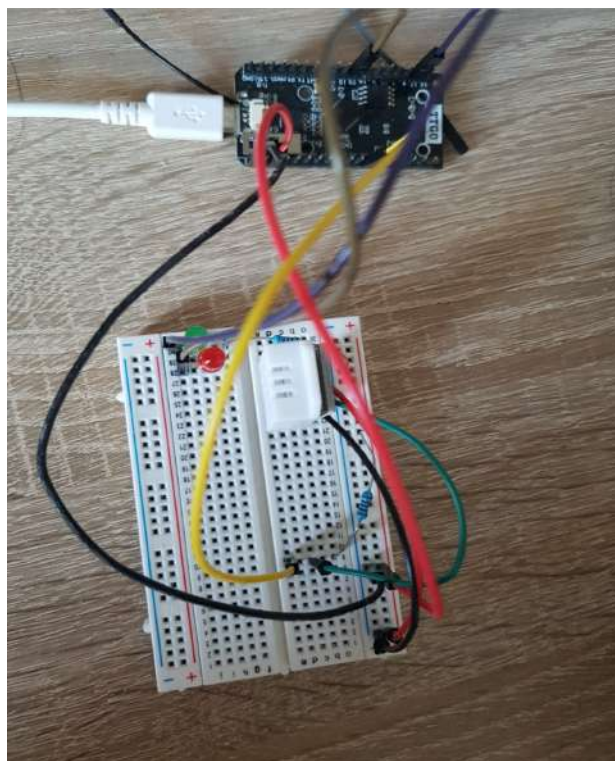


Abbildung 6: Aufbau der Schaltung für den Test der LoRaWan-Verbindung (Lora32 V1 ohn Oled)

Technische Spezifikation - Smarte Gartenbewässerung

Wenig später stand die neuere Version des TTGO Lora32 zur Verfügung: Lora32 OLED V2.1.6. Auf dem Display werden direkt die gemessenen Sensordaten angezeigt. So kann geprüft werden, ob die Daten korrekt an TTN übertragen werden und gleiche Werte angezeigt werden.

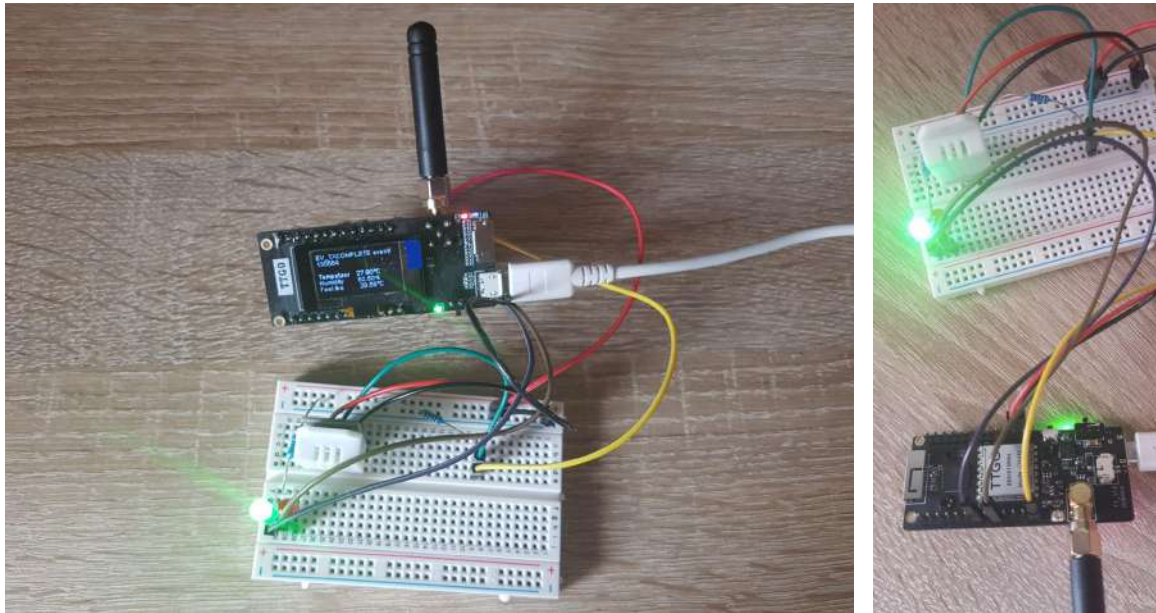


Abbildung7: Aufbau der Schaltung für den Test der LoRaWan-Verbindung (Lora32 OLED V2.1.6)

In Gitlab → Software → Node-Lora32 → src → main.cpp ist der C++ Code zu finden, der auf dem Lora32 installiert ist. Grundlegend gibt es eine Funktion zum Auslesen der Sensordaten des DHT22, zum Steuern von einer grünen und einer roten LED, zur Umwandlung der Daten in Byte mit dem Encoder, um sie versenden zu können und zum Senden der Daten über die Lora-Antenne.

Die Daten werden von einem TTN-Gateway empfangen und ins TTN-Netzwerk zu einem TTN-Server weitergeleitet. Mit dem Tool MQTT-Mosquitto oder MQTT-Explorer oder mit der Website von von TTN können wir auf die Daten in TTN zugreifen und die Sensordaten auslesen.

Technische Spezifikation - Smarte Gartenbewässerung

1. MQTT-Mosquito (als Client betrachtet)

Folgende Daten werden zur Verbindung über MQTT benötigt (um auf die Daten auf dem Daten TTN-Server zuzugreifen):

App ID on TTN: mygardenproject

Access key on TTN: ttn-account-v2.60jnFj-pF6rapK8BtiWsr2CQXM8TufQspWzjreeI2Zc

Folgende Befehle müssen in die Konsole eingegeben werden :

Active Mosquito service:

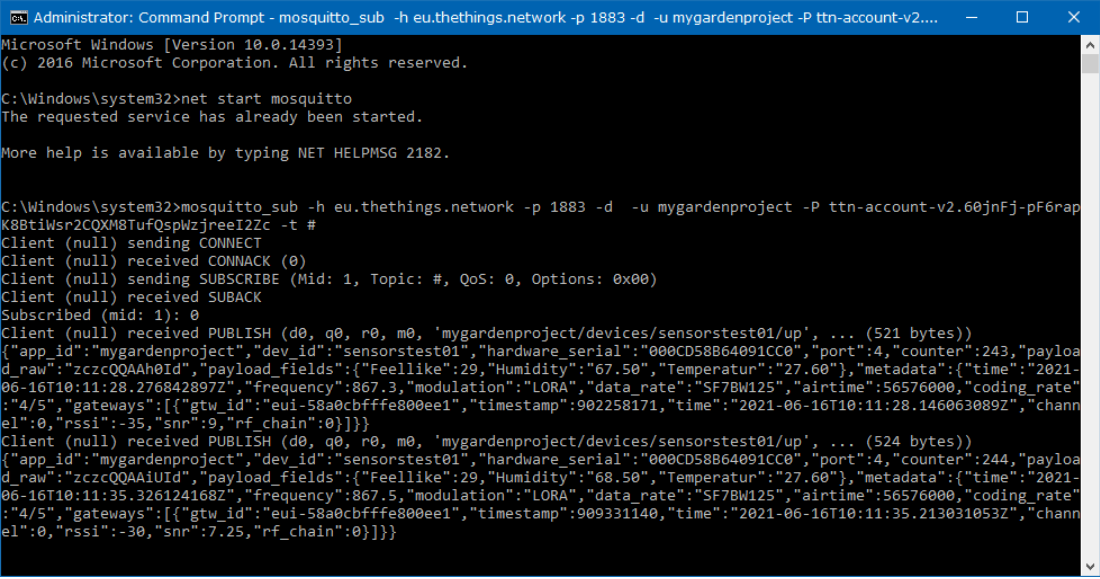
```
net start mosquitto
```

Subscribe our topic:

```
mosquitto_sub -h eu.thethings.network -p 1883 -d -u mygardenproject -P
```

```
ttn-account-v2.60jnFj-pF6rapK8BtiWsr2CQXM8TufQspWzjreeI2Zc -t #
```

In nachfolgendem Bild sieht man die Ausgabe in der CMD ür MQTT-Mosquitto.



```

Administrator: Command Prompt - mosquitto_sub -h eu.thethings.network -p 1883 -d -u mygardenproject -P ttn-account-v2....
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net start mosquitto
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.

C:\Windows\system32>mosquitto_sub -h eu.thethings.network -p 1883 -d -u mygardenproject -P ttn-account-v2.60jnFj-pF6rapK8BtiWsr2CQXM8TufQspWzjreeI2Zc -t #
Client (null) sending CONNECT
Client (null) received CONNACK (0)
Client (null) sending SUBSCRIBE (Mid: 1, Topic: #, QoS: 0, Options: 0x00)
Client (null) received SUBACK
Subscribed (mid: 1): 0
Client (null) received PUBLISH (d0, q0, r0, m0, 'mygardenproject/devices/sensorstest01/up', ... (521 bytes))
{"app_id":"mygardenproject","dev_id":"sensorstest01","hardware_serial":"000CD58864091CC0","port":4,"counter":243,"payload_raw":"zczcQQA0Id","payload_fields":{"Feellike":29,"Humidity":"67.50","Temperatur":"27.60"},"metadata":{"time":"2021-06-16T10:11:28.276842897Z","frequency":867.3,"modulation":"LORA","data_rate":"SF7BW125","airtime":56576000,"coding_rate":"4/5","gateways":[{"gtw_id":"eui-58a0cbfffe800ee1","timestamp":"902258171","time":"2021-06-16T10:11:28.146063089Z","channel":0,"rssi":-35,"snr":9,"rf_chain":0}]}}
Client (null) received PUBLISH (d0, q0, r0, m0, 'mygardenproject/devices/sensorstest01/up', ... (524 bytes))
{"app_id":"mygardenproject","dev_id":"sensorstest01","hardware_serial":"000CD58864091CC0","port":4,"counter":244,"payload_raw":"zczcQQA0IUIId","payload_fields":{"Feellike":29,"Humidity":"68.50","Temperatur":"27.60"},"metadata":{"time":"2021-06-16T10:11:35.326124168Z","frequency":867.5,"modulation":"LORA","data_rate":"SF7BW125","airtime":56576000,"coding_rate":"4/5","gateways":[{"gtw_id":"eui-58a0cbfffe800ee1","timestamp":"909331140","time":"2021-06-16T10:11:35.213031053Z","channel":0,"rssi":-30,"snr":7.25,"rf_chain":0}]}}
  
```

Abbildung 8: MQTT Mosquitto

Technische Spezifikation - Smarte Gartenbewässerung

2. MQTT-Explorer (Client)

Alternativ können die Daten auch mit Hilfe von MQTT-Explorer ausgelesen werden.

Hier kann sich der User mit den selben Anmeldedaten ins System einloggen.

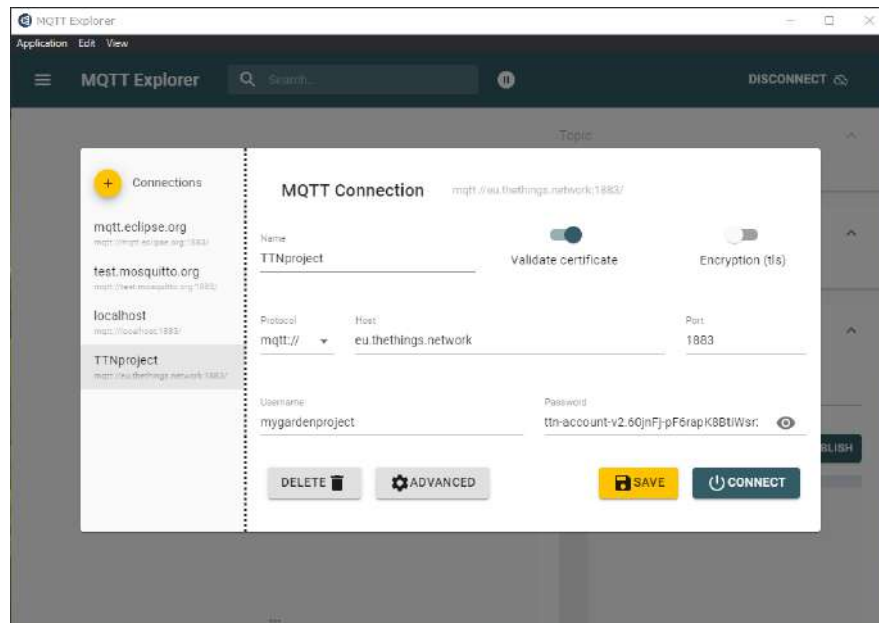


Abbildung 9: Einloggen bei MQTT-Explorer

Hier kann die Ausgabe im MQTT-Explorer gesehen werden:

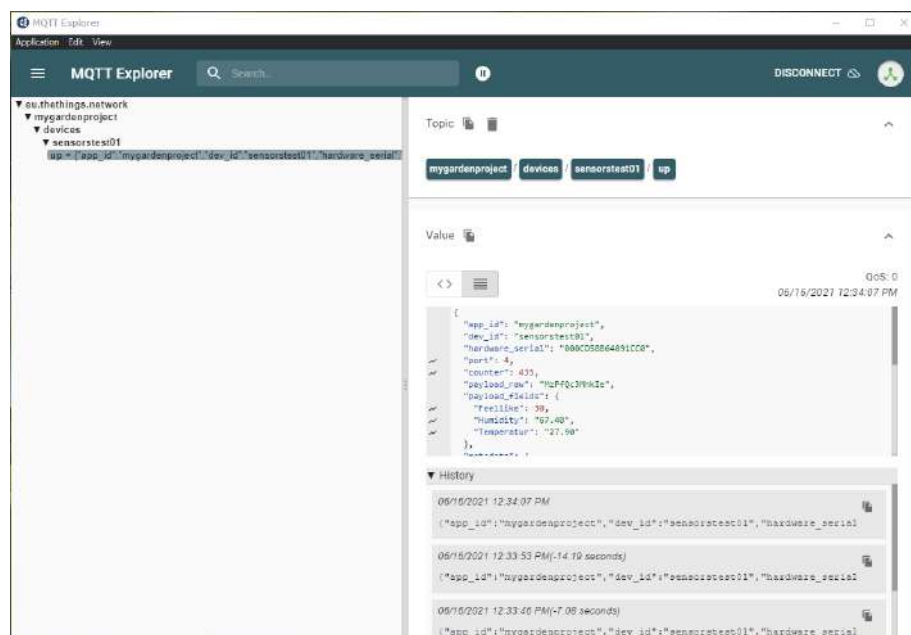


Abbildung 10: Ausgabe der Sensordaten MQTT-Explorer

3. TTN-Website (Server + Client)

Die schönste und einfachste Möglichkeit, um die Daten auszulesen, ist das direkte Nutzen der Website von TTN.

Um die Daten dekodieren zu können, muss zuerst das Payload-Format auf der Website von TTN geschrieben werden, um die empfangenen Daten von Byte zurück in ein lesbares Format umzuwandeln.

In folgendem Bild ist die Funktion zum Dekodieren zu sehen:

```

function Bytes2Float32(bytes) {
  var sign = (bytes & 0x80000000) ? -1 : 1;
  var exponent = ((bytes >> 23) & 0xFF) - 127;
  var significand = (bytes & ~(1 << 23));

  if (exponent == 128)
    return sign * ((significand) ? Number.NaN : Number.POSITIVE_INFINITY);

  if (exponent == -127) {
    if (significand === 0) return sign * 0.0;
    exponent = -126;
    significand /= (1 << 22);
  } else significand = (significand | (1 << 23)) / (1 << 23);

  return sign * significand * Math.pow(2, exponent);
}

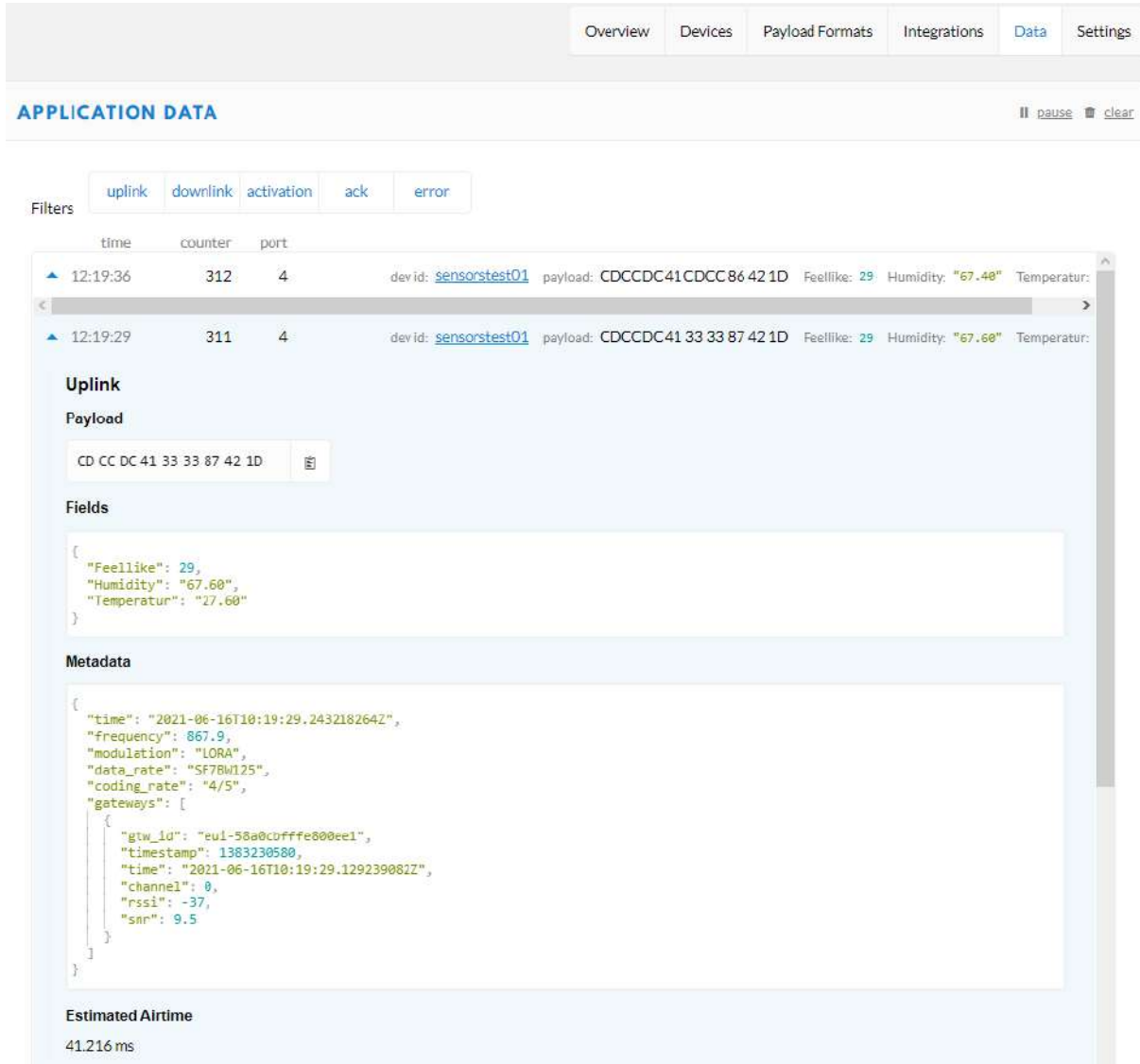
function Decoder(bytes, port) {
  switch (port) {
    case 1:
      var decoded_var = String.fromCharCode.apply(null, bytes.slice(0, 12));
      return { var: decoded_var };
    case 2:
      return { temp: String.fromCharCode.apply(null, bytes) };
    case 3:
      var t = bytes[3] << 24 | bytes[2] << 16 | bytes[1] << 8 | bytes[0];
      var h = bytes[7] << 24 | bytes[6] << 16 | bytes[5] << 8 | bytes[4];
      var f1 = bytes[9] << 8 | bytes[8];
      return {
        "6097c45e1c3320001c7c97ab": Bytes2Float32(t).toFixed(2),
        "6097c45e1c3320001c7c97ac": Bytes2Float32(h).toFixed(2),
        "6097c45e1c3320001c7c97ad": f1
      };
    case 4:
      var t = bytes[3] << 24 | bytes[2] << 16 | bytes[1] << 8 | bytes[0];
      var h = bytes[7] << 24 | bytes[6] << 16 | bytes[5] << 8 | bytes[4];
      var f1 = bytes[9] << 8 | bytes[8];
      return {
        "Temperatur": Bytes2Float32(t).toFixed(2),
        "Humidity": Bytes2Float32(h).toFixed(2),
        "Feellike": f1
      };
    default:
      return { var: String.fromCharCode.apply(null, bytes) };
  }
}

```

Abbildung 11: Payload-Dekodier-Code

Technische Spezifikation - Smarte Gartenbewässerung

Dann können die Daten bei TTN gelesen werden:



The screenshot displays the TTN web interface for an application. At the top, there are navigation tabs: Overview, Devices, Payload Formats, Integrations, Data (selected), and Settings. Below the tabs, the 'APPLICATION DATA' section is visible, with a 'pause' button and a 'clear' button. A filter bar shows 'uplink', 'downlink', 'activation', 'ack', and 'error' filters, with 'uplink' selected. The main data table lists two uplink messages from device 'sensorstest01'.

| time | counter | port | dev id | payload | Feellike | Humidity | Temperatur |
|----------|---------|------|---------------|--------------------|----------|----------|------------|
| 12:19:36 | 312 | 4 | sensorstest01 | CDCCDC41CDCC86421D | 29 | 67.48 | |
| 12:19:29 | 311 | 4 | sensorstest01 | CDCCDC41333387421D | 29 | 67.60 | |

Below the table, the 'Uplink' section is expanded, showing the 'Payload' as 'CD CC DC 41 33 33 87 42 1D'. The 'Fields' section displays a JSON object:

```
{
  "Feellike": 29,
  "Humidity": "67.60",
  "Temperatur": "27.60"
}
```

The 'Metadata' section shows a detailed JSON object:

```
{
  "time": "2021-06-16T10:19:29.243218264Z",
  "frequency": 867.9,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eu1-58a0c0fffe800ee1",
      "timestamp": 1383230580,
      "time": "2021-06-16T10:19:29.129239081Z",
      "channel": 0,
      "rssi": -37,
      "snr": 9.5
    }
  ]
}
```

At the bottom, the 'Estimated Airtime' is shown as 41.216 ms.

Abbildung 12: Datenausgabe TTN

4.6 Entwicklung der User Interface (UI)

Entsprechend der ersten Testphase, in der in der Testschaltung ein DHT22 Temperatur- und Luftfeuchtigkeitssensor und zwei LEDs verbaut waren, wurde ein User Interface erstellt, das folgende Funktionen hat:

- Zahlenwerte der aktuellen Sensordaten (Temperatur, Luftfeuchtigkeit, Feel-Like-Temperatur)
- Verlaufsdiagramme für diese Sensorwerte
- Schalter für den Wechsel zwischen Auto und Manual Mode
- Schalter für das Aktivieren/Deaktivieren des Aktors (LED) im Manual Mode
- Hinweis-Leuchte, die grün leuchtet, sobald ein Signal über TTN empfangen wird
- Diagramme für Signal Noise und Signal Strength (Werte sind gut, wenn: Signal Noise zwischen -20 dB bis 10 dB und Signal Strength zwischen -120 dBm bis -30 dBm)

Das User-Interface wurde mit Node-Red entwickelt und mit TTN über eine Schnittstelle verbunden.

Nachfolgend ist das User Interface zu sehen:

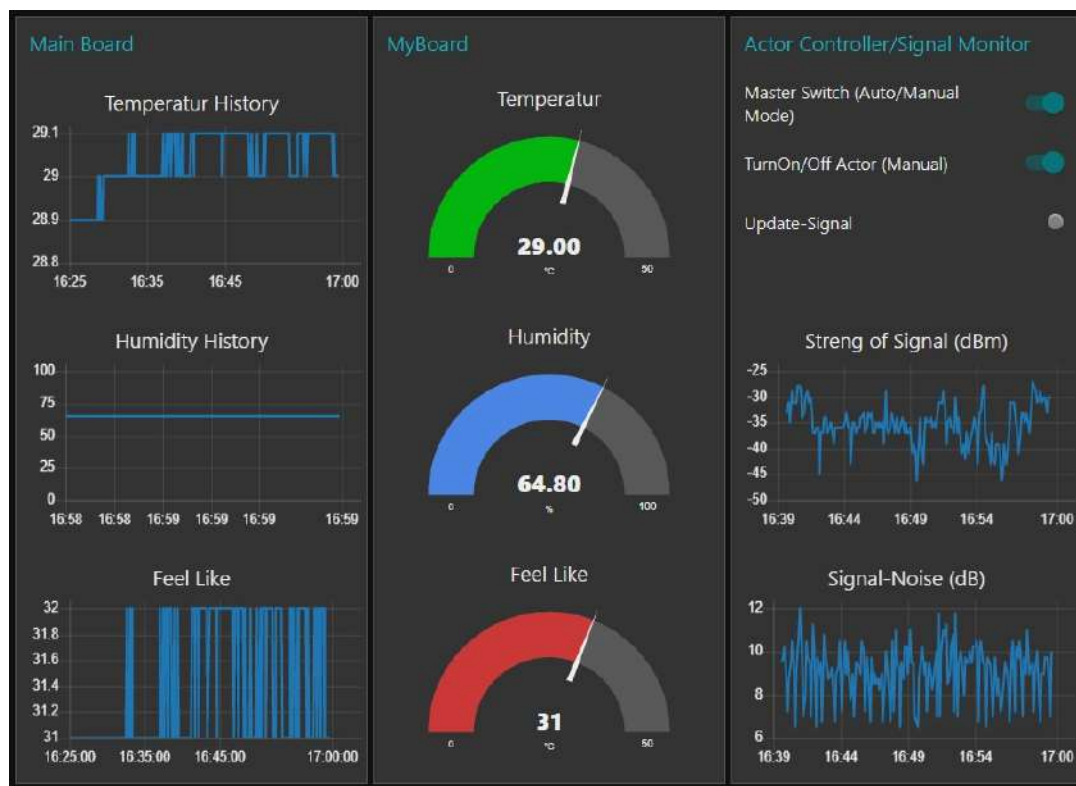


Abbildung 13: Erstes Test-UI

Technische Spezifikation - Smarte Gartenbewässerung

Nachdem das System mit den richtigen Sensoren und Aktoren erweitert wurde, musste auch das User Interface erweitert werden.

Nachfolgende Informationen bzw. Aktionen können im finalen User Interface gesehen und benutzt werden:

- Gauge: Aktuelle Werte der Sensoren, die einen Minimal- und Maximalwert haben. (Temperatur, Luftfeuchtigkeit, Bodentrockenheit, Luftdruck, Wasserdruck, Druck in Leitung und Wasserstand)
- Histogram: Verlauf der Sensordaten, um einen zeitlichen Überblick zu erhalten. (Temperatur, Luftfeuchtigkeit, Bodentrockenheit, Wasserdruck, Wasserstand, Wasserdurchfluss-Geschwindigkeit, Strength of Signal und Signal Noise)
- Text: Aktuelle Werte von Sensoren, deren Werte kein Maximum haben (Wasserdurchfluss Geschwindigkeit, Wasserdurchfluss Volumen)
- LED: visualisiert den aktuellen Status von Aktor und den Update-Status: (Update-Signal, Ventil und Pumpe Status)
- Switch/Schalter: Wechsel zwischen verschiedenen Modi /Zuständen (Manuell oder Auto Modus, Turn On/Off Watering System in Manual Mode und Debug Mode)
- Slider: Steuerungswerte, die zum autonomen Betrieb vom System gehören. (Bodenfeuchtigkeit zu an/aus, Wasserstand zu aus, Wasservolumen zu aus)

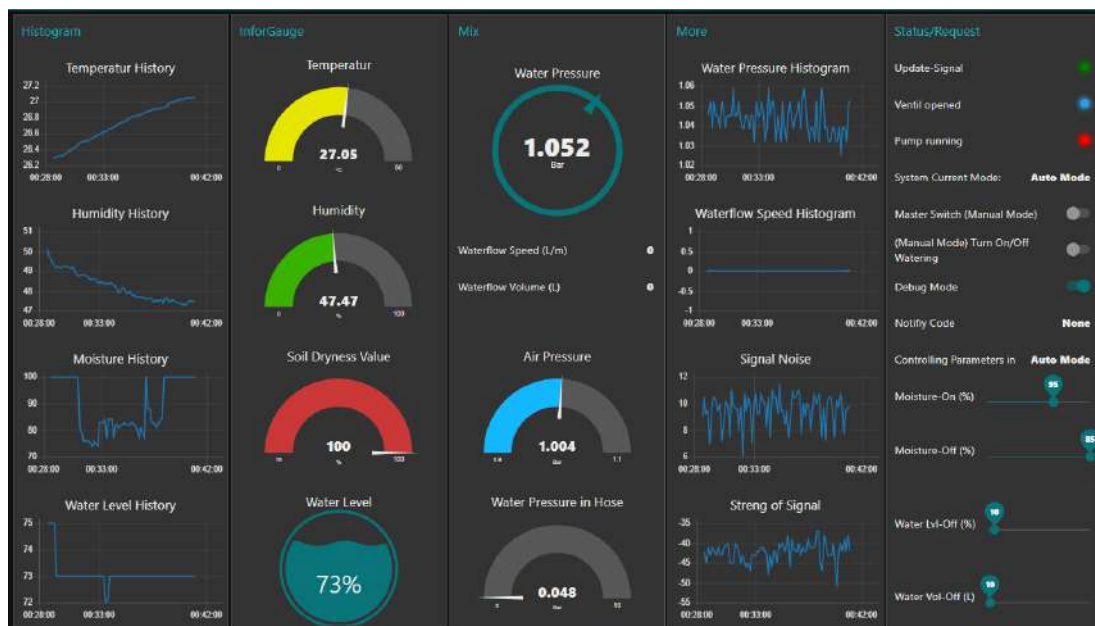


Abbildung 14: Finales UI

4.7 Integration von Telegram

Um eine weitere und sehr einfache Möglichkeit für den Nutzer bereitzustellen, um die Sensorwerte auszulesen. Deswegen wurde unser System mit Hilfe der Telegram Bot API mit dem Telegram-Messenger verbunden.

Der Nutzer kann somit schnell wichtige Informationen über aktuell Stand von Garten abfragen. Der Admin bekommt zudem jedes mal eine Nachricht, wenn die Bewässerung aktiviert oder deaktiviert wurde.

Nachfolgend sind Screenshots aus der Telegram-App auf einem Android-Smartphone.

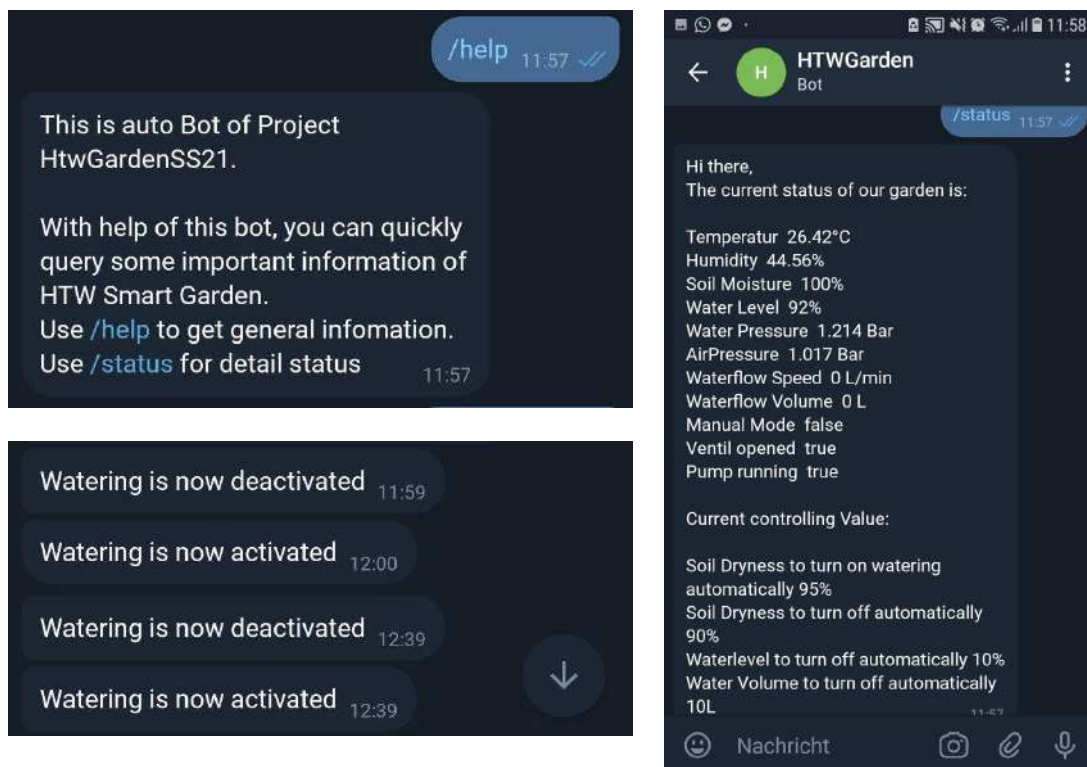


Abbildung 15: Screenshots von Telegram

Daten:

Telegram bot: HTWGarden (nicht HTW GardenSS21, da ist die alte Version)

Befehle:

- /help: allgemeine Informationen und Hinweise
- /status: aktuellen Stand vom Garten abfragen

4.8 Lora32-Komponente: Klassendiagramm

Um das System Modular zu gestalten und es einfach zu machen später neue Sensoren und Aktoren hinzuzufügen, wurde folgende Klassenhierarchie für die Komponenten, die an den Lora32 angeschlossen werden, definiert:

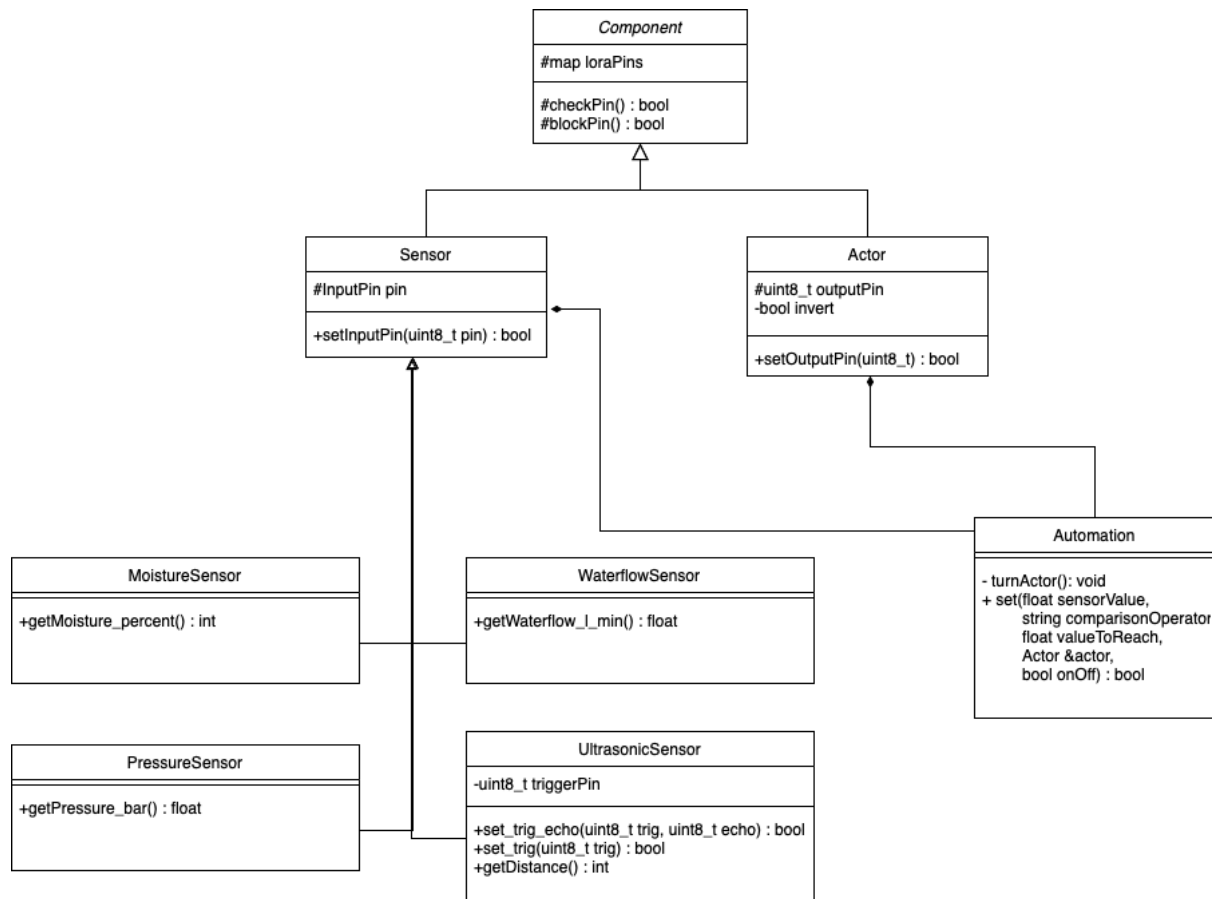


Abbildung 16: Klassendiagramm

Die Klassen Actor und Sensor erben von der Klasse Component. In der Klasse Component werden die Pins am Lora32 verwaltet:

- wird ein Sensor oder Aktor mit einem Pin registriert, kann dieser nicht nochmal verwendet werden
- es können nur Pins verwendet werden, die der Lora32 überhaupt hat

Von der Klasse Sensor erben die verwendeten Sensoren. Diese sind in einzelnen Klassen, da die Werte von jedem Sensor auf unterschiedliche Art und Weise ausgelesen werden müssen (analog, digital, interrupt).

Die Aktoren agieren alle gleich: man kann sie ein- oder ausschalten. Allerdings gibt es unterschiedliche Relais-Arten. Deswegen muss noch der bool-Wert "invert" hinzugefügt werden. Unser Relais schaltet in den Zustand "ein", wenn kein Signal anliegt. Das Signal muss also dementsprechend invertiert werden.

Zusätzlich gibt es noch eine Klasse Automation, die in der Set-Funktion einen Ist-Sensor-Wert, einen Operator als string, einen Soll-Sensor-Wert, den Actor und einen bool, ob der Aktor aktiviert oder deaktiviert werden soll, entgegen nimmt. Wird die Funktion aufgerufen, werden die Werte entsprechend dem Operator verglichen und der Aktor aktiviert oder deaktiviert.

4.9 Autonome Steuerung

Unser System kann autonom und ganz unabhängig von der Verbindung mit TTN und damit auch von Eingaben des Nutzers betrieben werden.

Dazu muss der Nutzer einen Schalter (Master-Switch) umlegen, um die autonome Steuerung zu aktivieren. Diese Information wird dann an den Mikrocontroller gesendet, woraufhin dieser unabhängig von LoraWan die Bewässerung steuern kann.

Ist der autonome Modus aktiviert, können Parameter für die autonome Steuerung mit vier Slidern eingestellt werden:

- Moisture-On: ab welchem Trockenheitswert soll die Bewässerung aktiviert werden
- Moisture-Off: unter welchem Trockenheitswert soll die Bewässerung deaktiviert werden
- Water Lvl-Off: unter welchem Wasserstandswert soll die Bewässerung deaktiviert werden
- Water Vol-Off: ab welchem Wert für das durchgeflossene Wasser soll die Bewässerung deaktiviert werden

Hinweis: Water Vol-Off funktioniert mit Hilfe von FreeRTOS (Free Real Time Operating System) möglich und bis jetzt nur ein experimentelles Feature. Nachdem der eingestellte

Technische Spezifikation - Smarte Gartenbewässerung

Water Vol-Off Wert erreicht wird, wird das System automatisch einen Neustart machen, um den Water-Volume-Wert wieder auf 0 zurückzusetzen

Nachfolgend können der Master-Switch und die Slider für die Parameter gesehen werden (links ist noch der Manual Mode aktiviert, rechts der Auto Mode) :

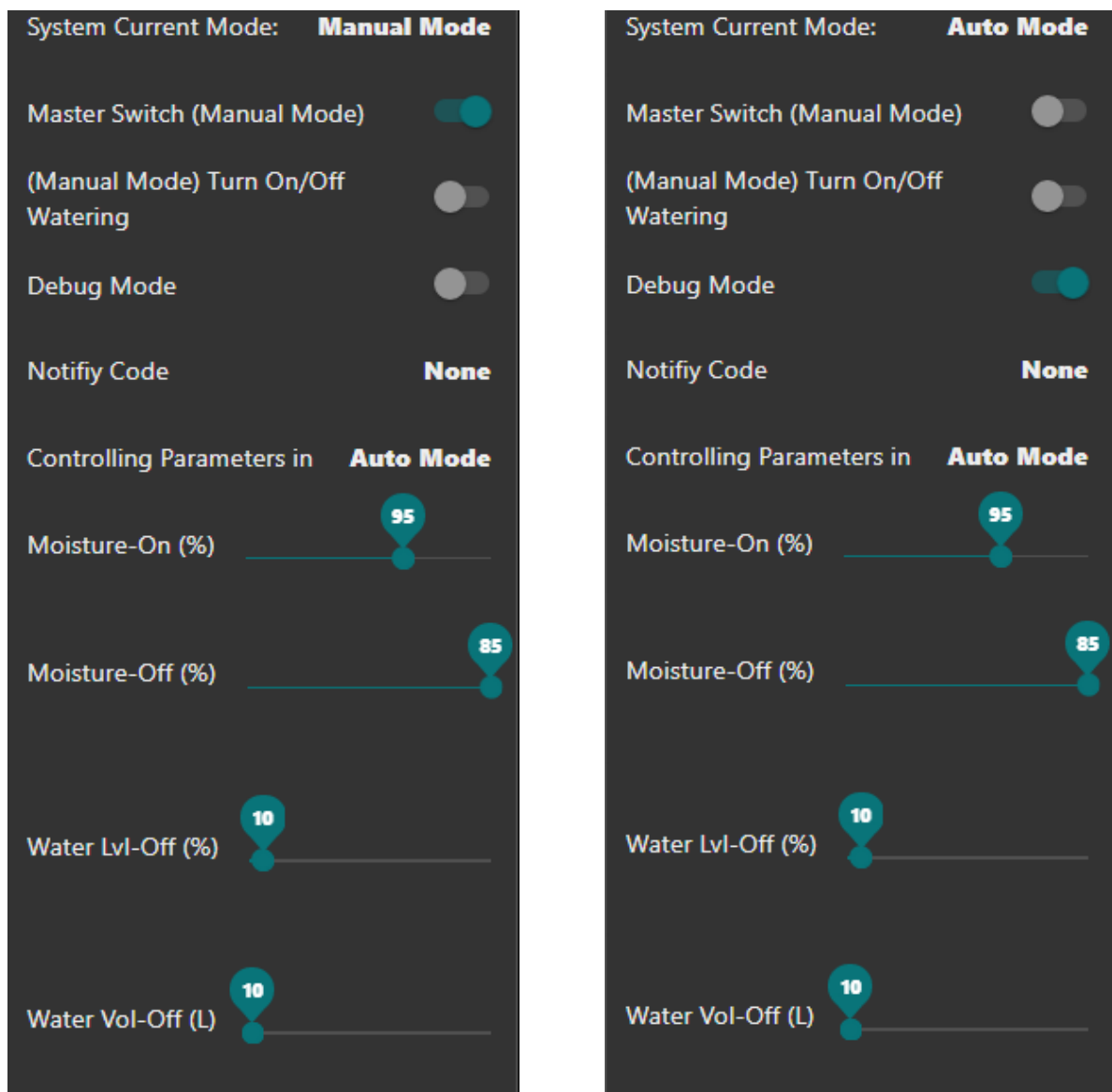


Abbildung 17: Einstellungen für die Autonome Steuerung

4.10 Fehlererkennung und -behandlung

Das System hat die Fähigkeit, selbst Fehler zu erkennen und die Bewässerung dann automatisch zu deaktivieren, um kein Wasser zu verschwenden und Schäden zu vermeiden.

Es sind zwei Fehlerszenarien definiert:

1. Druck in der Leitung sinkt unter 1 bar (wahrscheinlich Leck oder defekte Pumpe)
2. Druck in der Leitung steigt über 5,5 bar (wahrscheinlich Verstopfung)

Die voreingestellten Werte von 1 bar und 5,5 bar sind nur zur Simulation und müssen nach dem kompletten Aufbau (mit allen Schläuchen, Verbindungsstücken und Tropfern bzw. Regnern und weiteren Teilen) neu ermittelt werden, da sich dadurch die Vorbedingungen für die Standard-Werte im System ändern.

In nachfolgender Tabelle stehen Fehlercodes und der Text für die Fehleranzeige im UI:

Tabelle 4: Fehleranzeige und Fehlercodes

| Fehler | (Fehler-)code in TTN | Fehleranzeige UI |
|-------------------|----------------------|------------------|
| 1. (unter 1 bar) | 102 | Pump defect |
| 2. (über 5,5 bar) | 101 | Got stuck |
| Kein Fehler | 100 | None |

5. Spezifikation der Hardware




5.1 Einzelteile

Nachfolgend aufgelistete Komponenten wurden von uns gezielt herausgesucht. Wir haben auf die Kompatibilität zwischen den Teilen geachtet und wenn nötig auch passende Adapter besorgt.

Ein detaillierter Schaltplan und Bauplan für die Bewässerungsanlage folgt demnächst.

Auflistung aller elektronischen Komponenten:

Tabelle 5: Elektronische Komponenten

| Name | Beschreibung | Anwendungen | Schnittstellen & Spannungsbelegung | Bild |
|--|--|---|---|---|
| LoRa32 Mikrocontroller | Mikrocontroller board, wie ESP32 + LoRaWAN-Antenne | Entwicklerboard, Steuerung, Regelung, Berechnungen, Drahtlose Verbindung mit WLAN, Bluetooth, LoRaWAN | Eingang 5V(default) Eingang 3.3V(option) Ausgang 5V, 3.3V |  <p>https://www.makershop.de/plattformen/esp8266/ttgo-esp32-paxcounter-lora/</p> |
| Coolty High Pressure Diaphragm Water Pump | Pumpe | Wasser pumpen | Betriebsspannung DC 12V 100PSI 4L/min |  <p>https://amzn.to/3zsKp1O</p> |
| AZDelivery 4-relay module 5 V | Relais | Steuerung der Pumpe | Eingang 5V/3V Ausgang max 250V/10A AC, 30V/10A DC |  <p>https://amzn.to/3iHNncN</p> |

Technische Spezifikation - Smarte Gartenbewässerung

| | | | | |
|--|------------------------|---|---|--|
| EXLECO 12 V NC G1/2 Inch Electro Solenoid Valve | Magnetventil | Schließen des Wasserdurchflusses nach der Pumpe | G 1/2" Betriebsspannung DC 12 V |  https://amzn.to/3goN8BY |
| G1/4 inch pressure sensor | Wasserdruck- sensor | Messung des Wasserdrucks | Betriebsspannung DC 5 V G 1/4" (muss durch Spannungsteiler mit Lora32 verbinden) |  https://amzn.to/3iDe3vh |
| ARCELI YF-S201 1-30L/min Water Flow Meter | Durchfluss- sensor | Messung des Wasserdurchflusses | The lowest rated operating voltage: DC4.5 5V-24V. Maximum working current: 15mA (DC 5V) Working Voltage Range: DC 5~18V |  https://amzn.to/3xmhiM0 |
| Aukru Ultrasonic Module HC-SR04 | Ultraschall- sensor | Messung des Wasserstandes | Static Current(Max) :2mA ; Electrical Level Eingang:5V. ranging distance: 2cm~500 cm resolution: 0.3 cm |  https://amzn.to/3gmwvGU |

Technische Spezifikation - Smarte Gartenbewässerung

| | | | | |
|---|------------------------------|---|----------------------------|---|
| DHT22 Digital Temperature and Humidity Sensor AM2302 | Temperatur-sensor (optional) | Messung der Temperatur und Luftfeuchtigkeit | Betriebsspannung 3.3V |  https://amzn.to/3xkpc8r |
| AZDelivery Soil Moisture Sensor Hygrometer Module V1.2 | Bodenfeuchtig-keitssensor | Messung der Bodenfeuchte | Eingang 5V Ausgang 3.3V |  https://amzn.to/3pSUMkS |

Zusätzlich werden noch ein Steckbrett und Kabel zur Verbindung der Komponenten eingesetzt.





Technische Spezifikation - Smarte Gartenbewässerung

Auflistung Bewässerungskomponenten:

Tabelle 6: Bewässerungskomponenten

| Name | Beschreibung | Anwendungen | Schnittstellen & Spannungsbelegung | Bild |
|---|---|---|--|--|
| Flexible Tube TX | Verbindung Schlauch | Verbindung der Teile für den Wasserfluss | 13 mm x 3 mm (1/2 Inch), 5 m Long PVC Hose with Fabric |  https://amzn.to/3xr0hjP |
| Homgif Garden Irrigation System, Micro Drip Irrigation Kit | Bewässerung Schlauch (Drip System) | Beförderung des Wassers nach Magnetventil zu den Beeten, Bewässerung der Pflanzen | 30 m Schlauch bis zu 35-40 m ² bewässern |  https://amzn.to/3xlwLvH |
| T-piece with external/internal/external thread | Schlauchverbindung Schlauch Wasserdrucksensor | Anschluss des Wasserdrucksensors an der Wasserleitung | Size: G 1/2" max. Druck: 16 bar |  https://amzn.to/3iElBy3 |
| Threading Adapter | Wasserdrucksensor Adapter | Adapter um Wasserdrucksensor mit T-Stück zu verbinden | NPT 1/2 Male to G 1/4 Female |  https://koolance.com/threading-adapter-npt-1-2-male-to-g-1-4-female-adt-n12m-g14f |

Technische Spezifikation - Smarte Gartenbewässerung

| | | | | |
|--|--|--|---|--|
| Gardena Tap Connector | Schlauchverbindung Schlauch Ventil-Drip System | Verbindung nach dem Magnetventil mit dem Drip-System | 21mm bzw. 1/2 Zoll zum Gartenschlauch (Gardena-Anschluss) |  https://amzn.to/3gx6jIK |
| Sourcing Map 3 x Brass Hose Connector | Schlauchverbindung Sensor zum Schlauch | Verbindung der Komponenten mit dem Verbindungsschlauch | Size: 14mm spike x G1/2 socket; Total length: 35mm; Hex width: 23mm; Spike length: 23mm |  https://amzn.to/3wkJpuK |
| BGS 8095-12x20 | Hose Clamps | Befestigung der Schläuche zu den Komponenten (festziehen) | 12 x 20 mm 10 Pieces |  https://amzn.to/3gwqlmL |
| Hose Clamps 2 | Hose Clamps | Befestigung der Schläuche zu den Komponenten (festziehen) (andere Größe) | 8x12 mm, Stainless Steel, 10 pcs. |  https://amzn.to/3gBAw9B |

5.2 Schaltplan

Nachfolgend ist der Schaltplan für unser umgesetztes System:

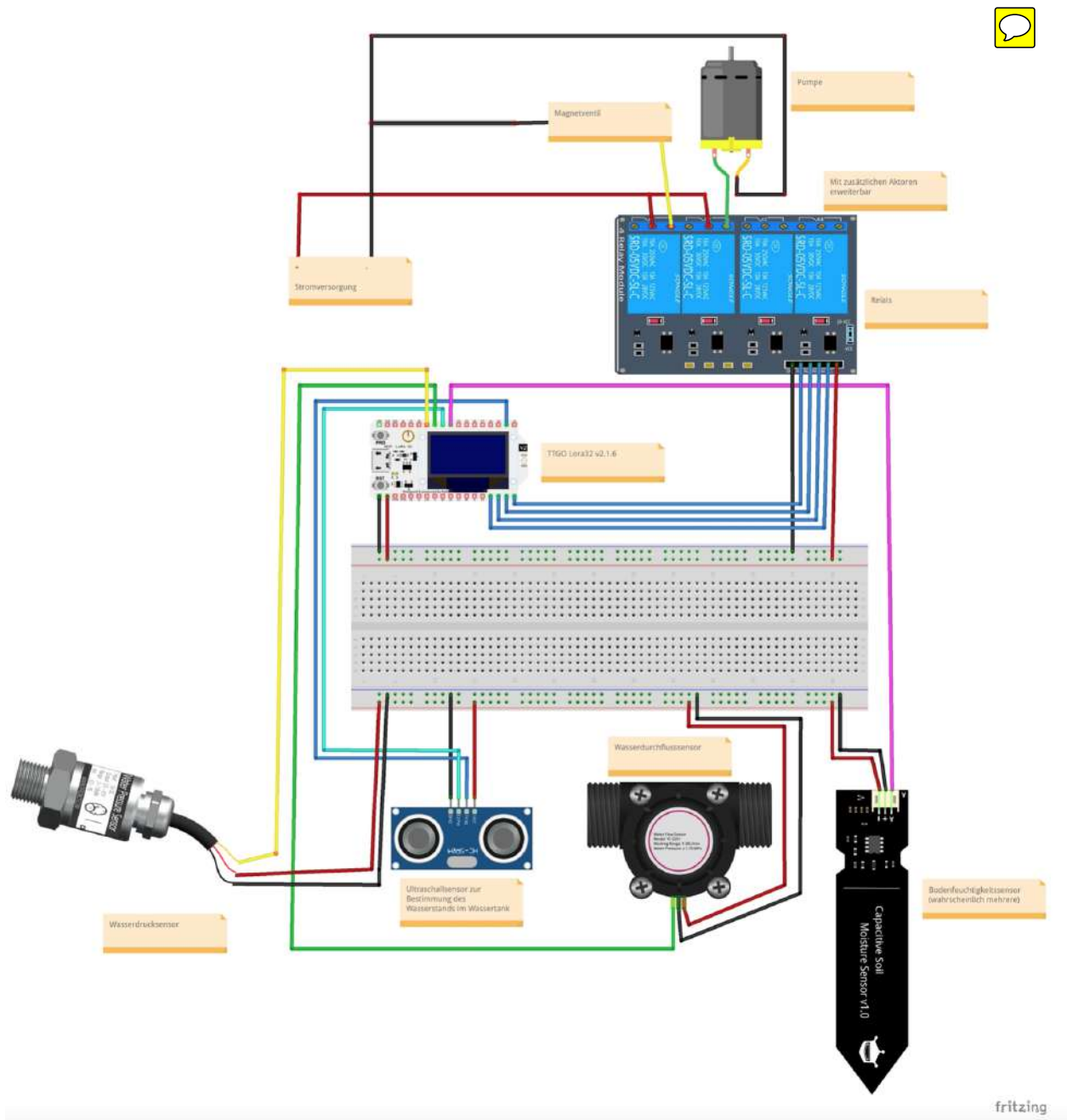


Abbildung 18: Schaltplan

Technische Spezifikation - Smarte Gartenbewässerung

Da das System modular erweiterbar ist, bleibt auch der Schaltplan dynamisch und veränderbar. Je nachdem welche und wie viele Aktoren und Sensoren angeschlossen sind, verändert sich der Schaltplan. Der Grundaufbau bleibt jedoch der gleich:

- Sensoren werden an die Stromversorgung und Masse angeschlossen (+ und -) und mit einem (oder in einigen Fällen auch mehreren) Pins mit dem Lora32-Mikrocontroller verbunden. Je nachdem welches Signal die Sensoren ausgeben (digital, analog, interrupt), müssen sie an passende Pins angeschlossen werden.
- Aktoren werden über ein Relais gesteuert. D. h. der Mikrocontroller ist nicht direkt mit dem Aktor, sondern nur mit dem Relais verbunden. Das liegt daran, da die Aktoren wesentlich mehr Strom benötigen, als der Lora32 liefern kann. Weitere Aktoren würden also ebenfalls an ein Relais und eine andere externe Stromquelle (z. B. Solaranlage im Urban Garden) verbunden werden.

5.3 Halterung für den Ultraschallsensor

Zur Wasserstandsmessung wird ein Ultraschallsensor verwendet, der im Deckel vom Wassertank angebracht wird. Damit er angebracht werden kann, wurde eine Halterung entwickelt und 3D-gedruckt. Somit kann der Sensor in die Halterung gesteckt werden und die Halterung an den Deckel geklebt werden und der Sensor bleibt unbeschädigt.

In nachfolgendem Bild kann der Deckel des Wassertanks und als Referenz der Ultraschallsensor gesehen werden:

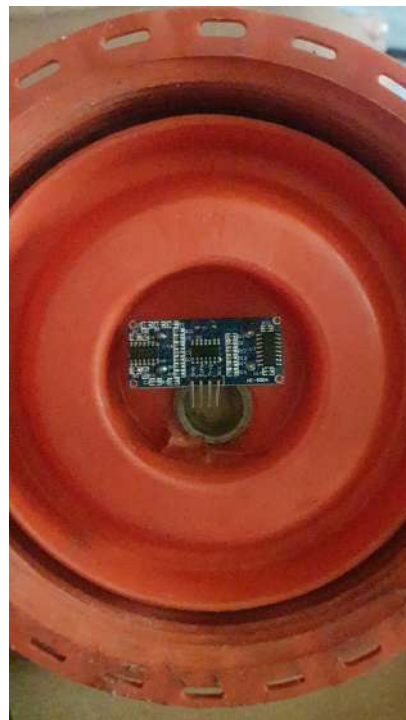


Abbildung 19: Wassertankdeckel

Zum Erstellen der Halterung wurde der Sensor vermessen und die Halterung in CAD konstruiert:

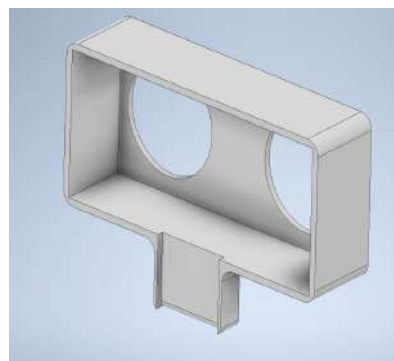


Abbildung 20: CAD-Konstruktion vom Halter

Technische Spezifikation - Smarte Gartenbewässerung

Anschließend wurde die Halterung an der HTW in einem 3D-Drucker gedruckt.

In folgenden Bildern kann gesehen werden, wie der Sensor in der Halterung angebracht ist:

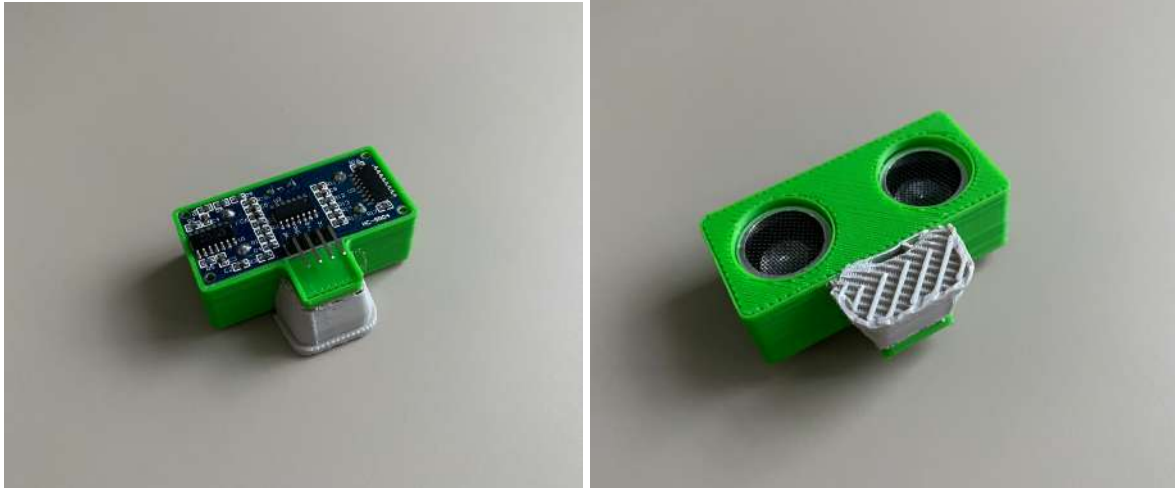


Abbildung 21: Ultraschallsensor in der Halterung

5.4 Gesamtaufbau

Anhand des Schaltplans wurde das System komplett aufgebaut.

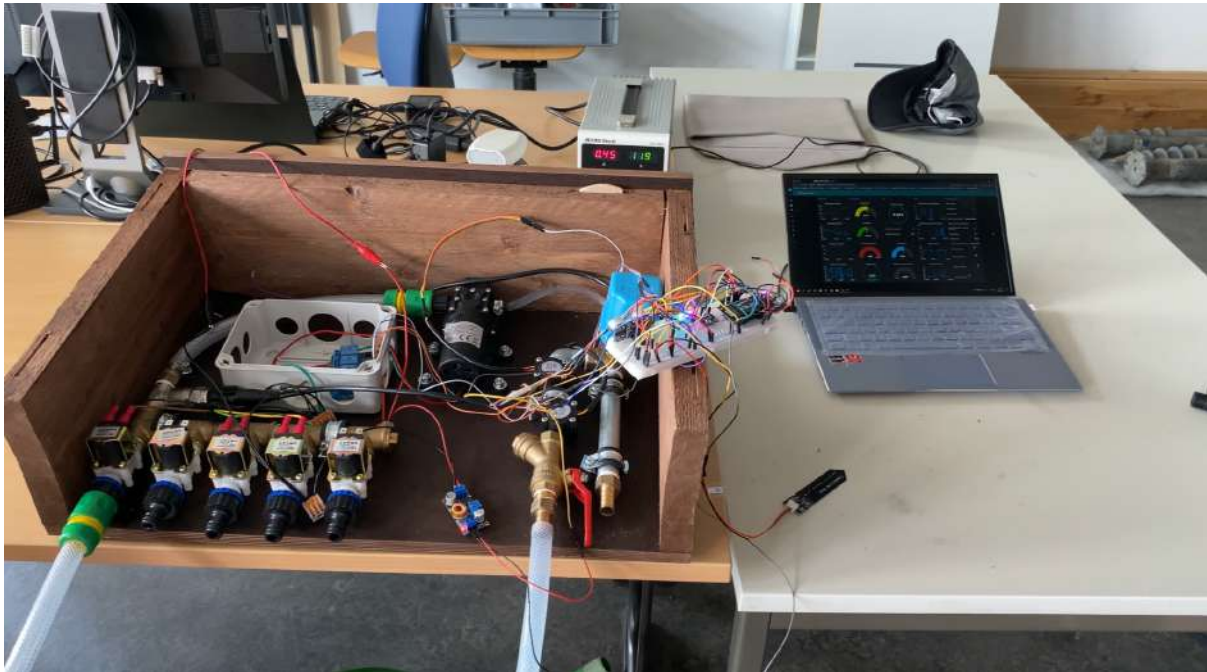


Abbildung 22: Bild des aufgebauten Gesamtsystems

Als externe Stromversorgung wurde hierbei ein Labornetzteil verwendet, um die Aktoren mit Strom zu versorgen. Der Mikrocontroller ist mit Hilfe eines Spannungswandler mit der Stromversorgung verbunden, da er weniger Spannung als die Aktoren benötigt.

Das komplette System und weitere Erklärungen sind in nachfolgendem YouTube-Video zu sehen, das von uns erstellt wurde:

<https://www.youtube.com/watch?v=hPiacfdj6qs>

Hier kann gesehen werden, dass bereits fleißig an Erweiterungen und zusätzlichen Funktionen von Herr Holger Martin gearbeitet wird. An dieser Stelle auch nochmal vielen Dank an Holger, der uns beim Aufbau unterstützt hat und das Projekt noch spannender gemacht hat.