

Technische Spezifikation

Smarte Gartenbewässerung über LoRaWAN

Sehr gute tech. Spezifikation, obwohl die Komponentenbeschreibung nicht ganz vollständig ist (für dieses Sprint). Weitere Anmerkungen siehe unten!

Wertung:

TechSpec : 4,5 Punkte

Anwendung : 10 Punkte

Mitarbeiter und Autoren:

- Rami Hammouda
- Khac Hoa Le
- Jaro Machnow

Letzte Änderung: 16.06.2021

Version: 1.3

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 Komponenten	12
4.2 Schnittstellen zwischen den Komponenten	14
4.3 Technologiestack	15
4.4 Anmeldedaten für TTN	16
4.5 Testphase 1 - Senden und Empfangen von Daten über TTN	17
5. Spezifikation der Hardware	23
5.1 Einzelteile	23

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 der Dokument

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:

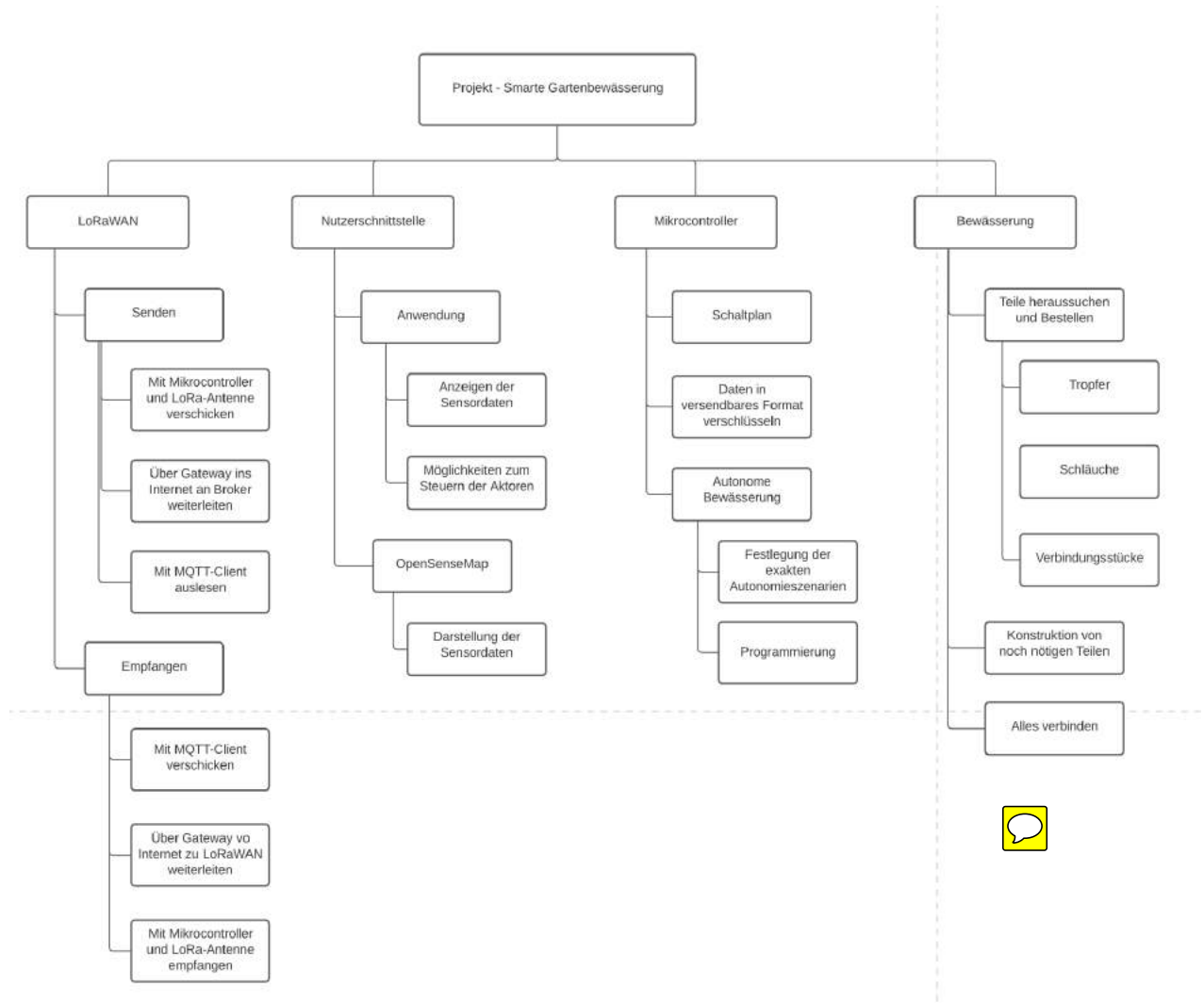


Abbildung 1: Übersicht über die Aufgabenstruktur

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:

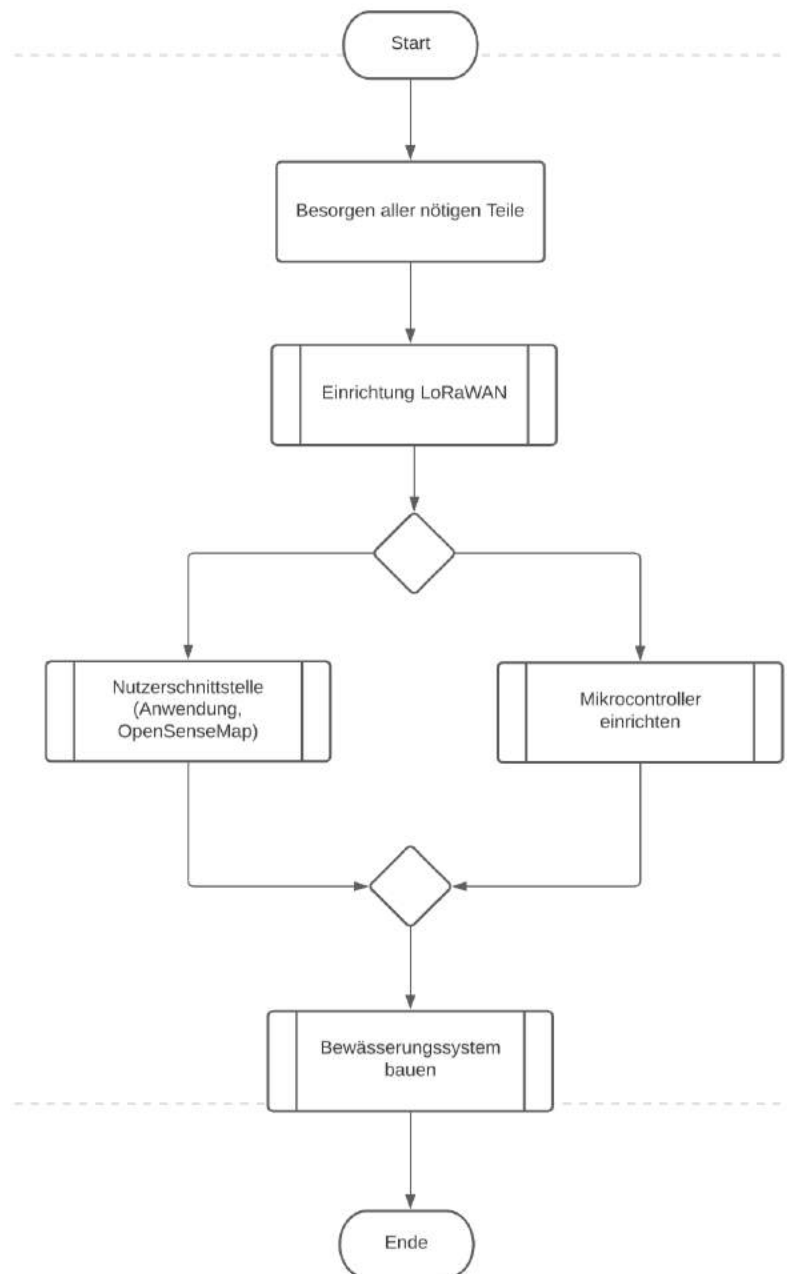


Abbildung 2: Diagramm des Realisierungsprozesses

2.2 Fachlicher Workflow

Nachfolgend 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.

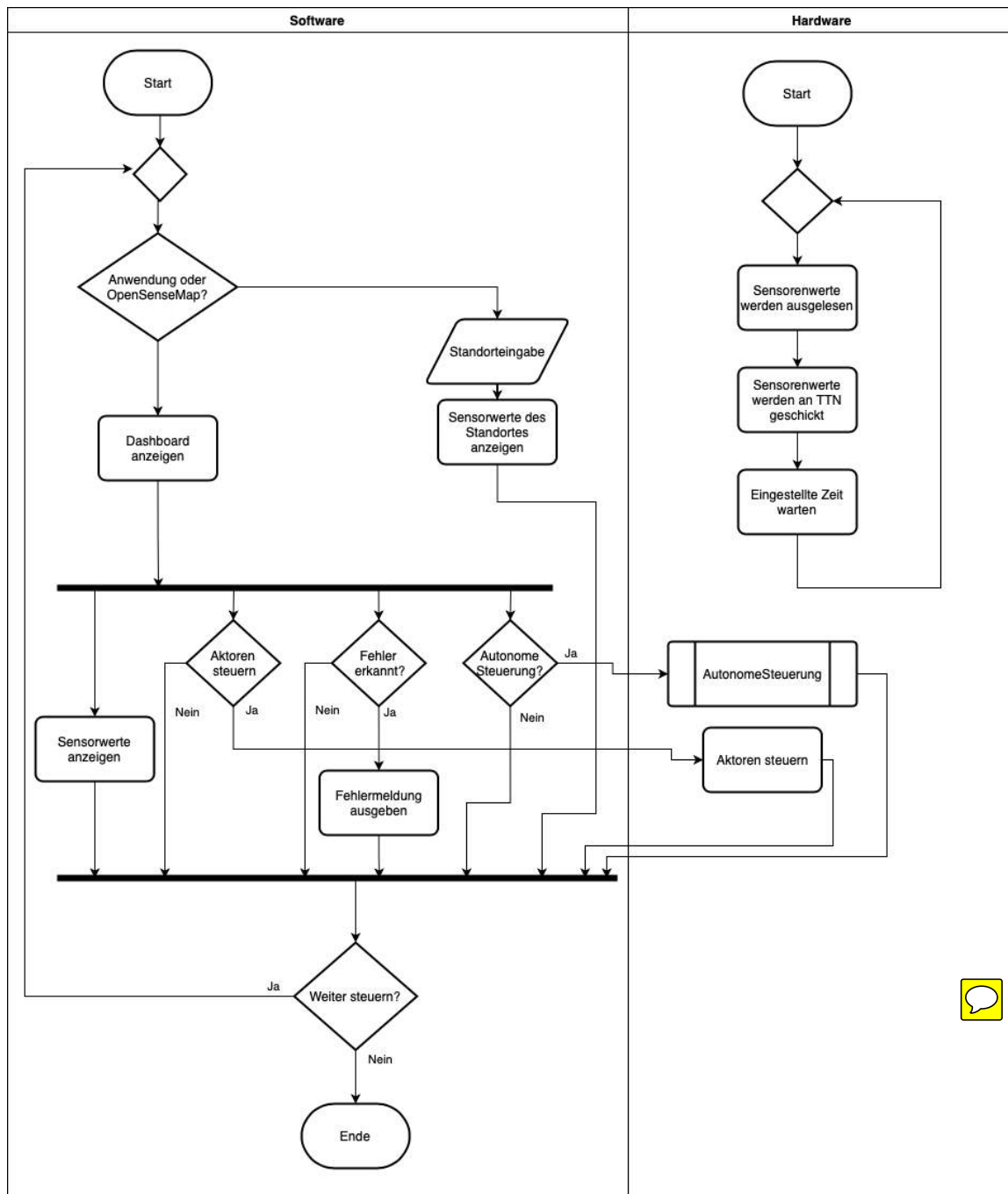


Abbildung 3: Fachlicher Workflow

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

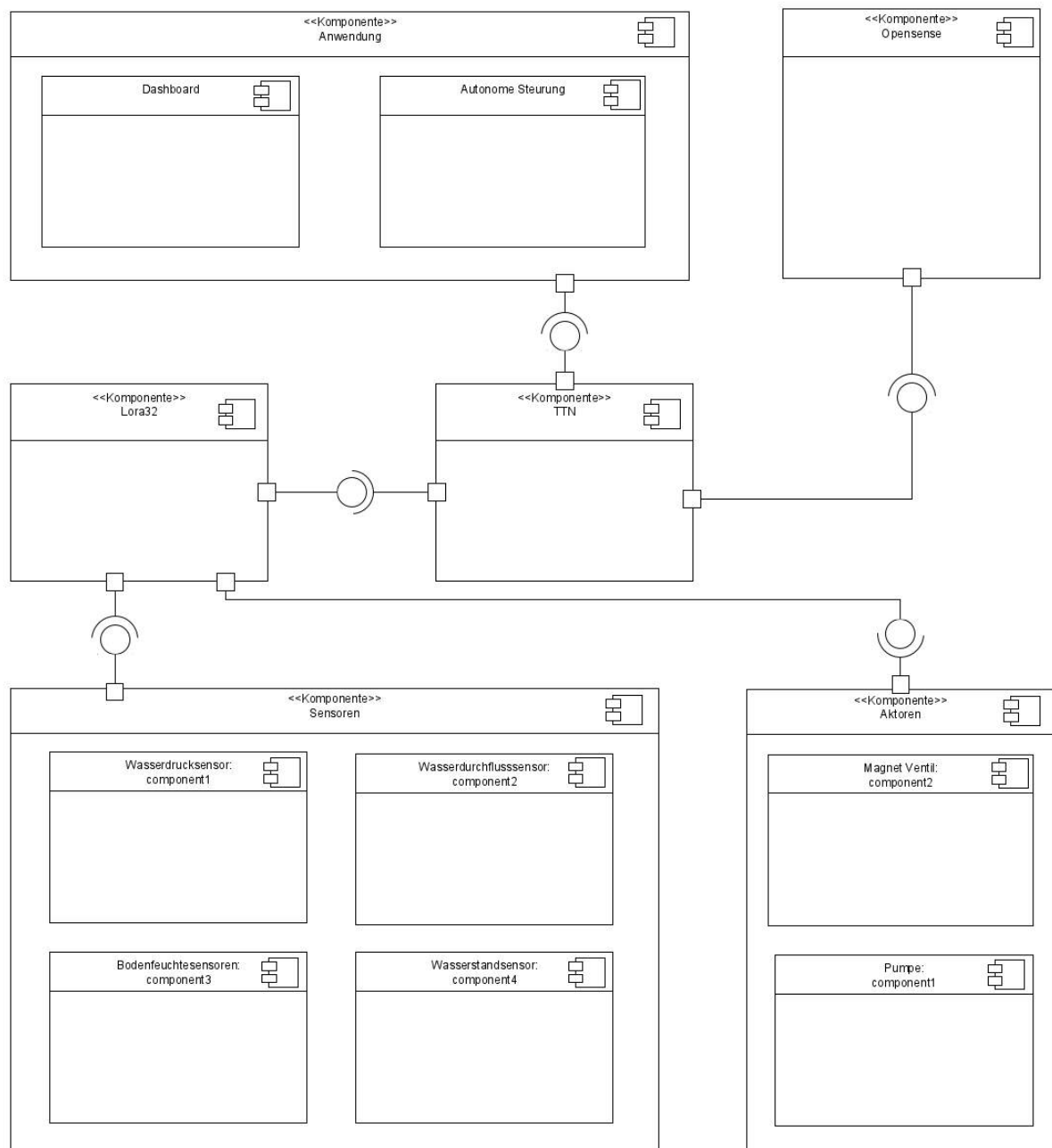


Abbildung 4: Komponentendiagramm

3.2 Beschreibung der Komponenten

Das Komponentendiagramm zeigt die verschiedenen Schichten **unseres** 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 Komponenten

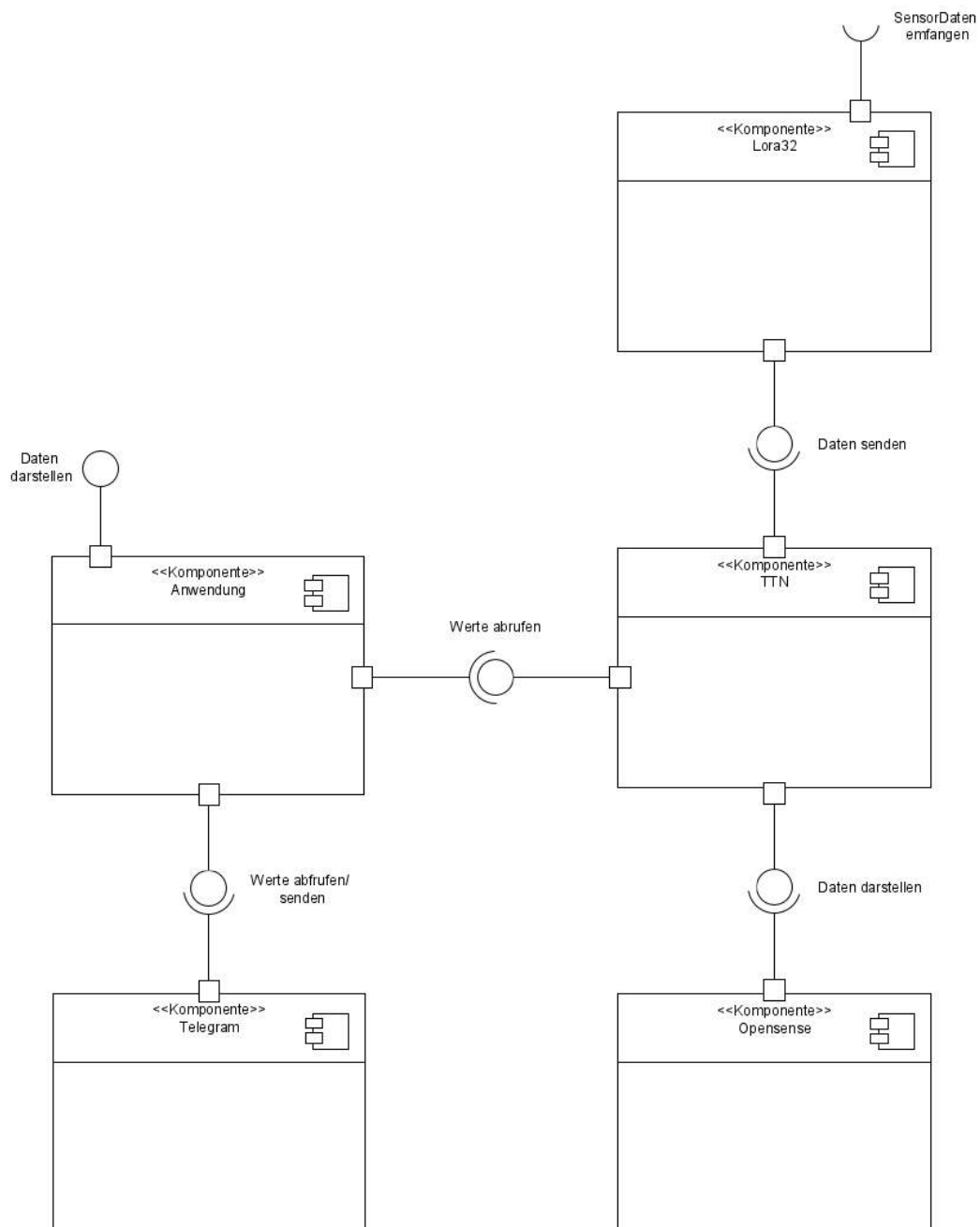



Abbildung 5: Komponentendiagramm Software

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 automatische 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 Testphase 1 - Senden und Empfangen von Daten über TTN

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

Wenn die Daten von LoRa32 sicher erzeugt werden, verbinden **wir** LoRa32 mit dem TTN-Server. Bei der TTN-Cloud kann **man** Uplinks (Signal von Sensoren) lesen, und die Downlinks (Signal zur Sensoren) schicken. **Wir** testen das Empfangen von Sensordaten und das Schicken von Daten zur Steuerung von Aktoren.

In **unserer** Testschaltung werden die Daten des DHT22 Temperatur- und Luftfeuchtigkeitssensors vom Lora32 erfasst. Dann werden die Daten weiter an TTN 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 **unserem** ersten Test stand uns ein Lora32 v1 zur Verfügung. Ein Bild dieser Schaltung ist in nachfolgender Abbildung zu sehen:

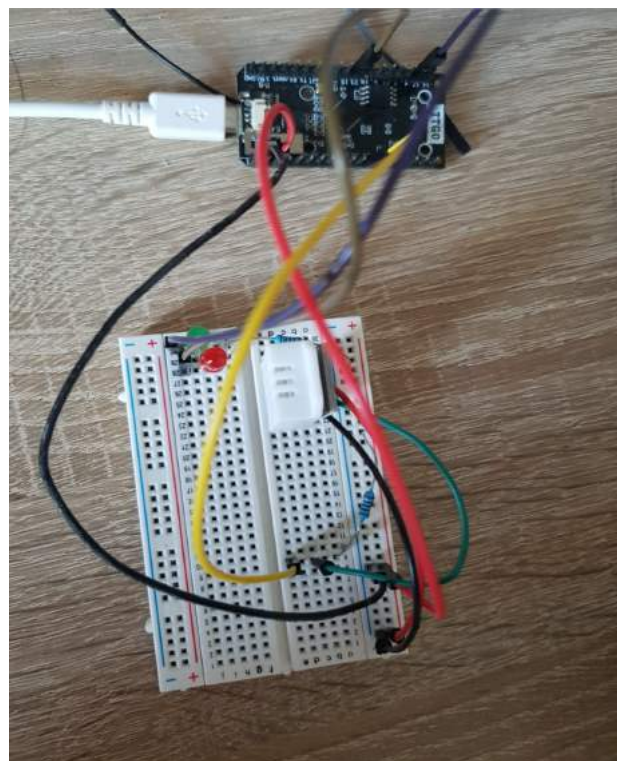


Abbildung 6: Aufbau der Schaltung für den Test der LoRaWan-Verbindung (Lora32 V1 ohn Oled)

Technische Spezifikation - Smarte Gartenbewässerung

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

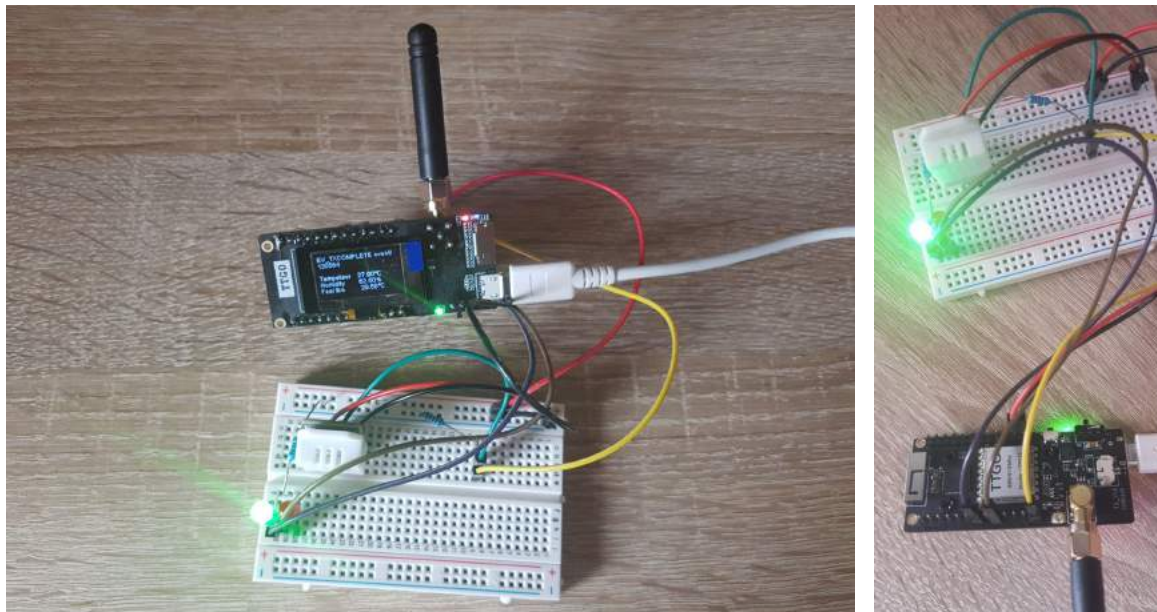


Abbildung7: Aufbau der Schaltung für den Test der LoRaWan-Verbindung (Lora32 OLED V2.1.6)

In Gitlab → Software → Node-Lora32 → src → main.cpp ist der C++ Code zu finden, der auf dem Lora32 installiert ist. Grundsätzlich 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 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:

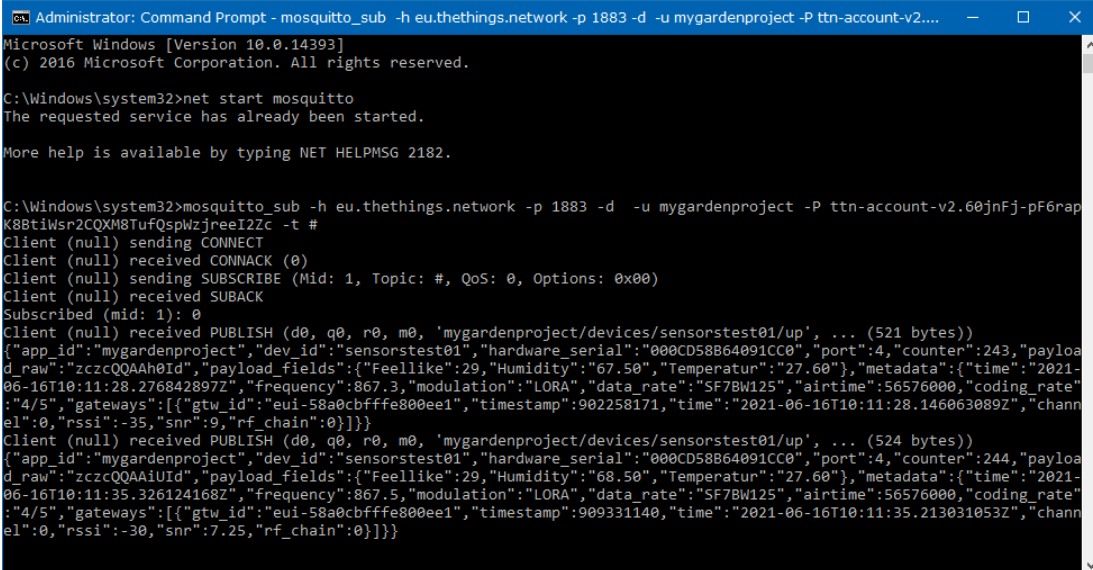
net start mosquitto

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":"000CD58B64091CC0","port":4,"counter":243,"payload_raw":"zczcQAAh0Id","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":"000CD58B64091CC0","port":4,"counter":244,"payload_raw":"zczcQAAh0Id","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 Mosquito

Technische Spezifikation - Smarte Gartenbewässerung

2. MQTT-Explorer (Client)

Alternative können **wir** auch mit Hilfe von MQTT-Explorer die Daten auslesen.

Hier kann **man** sich mit den selben Anmeldedaten ins System einloggen.

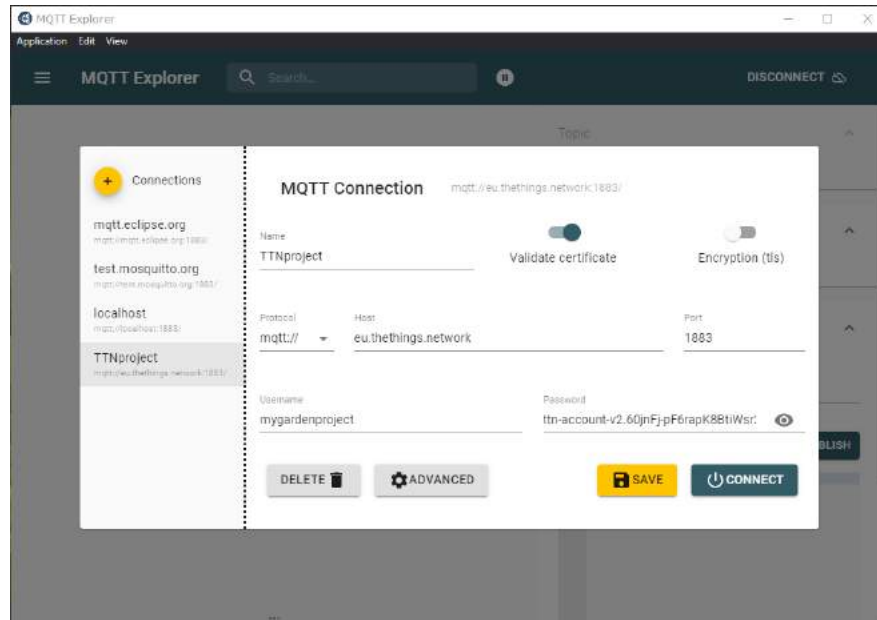


Abbildung 9: Einloggen bei MQTT-Explorer

Hier kann die Ausgabe im MQTT-Explorer gesehen werden:

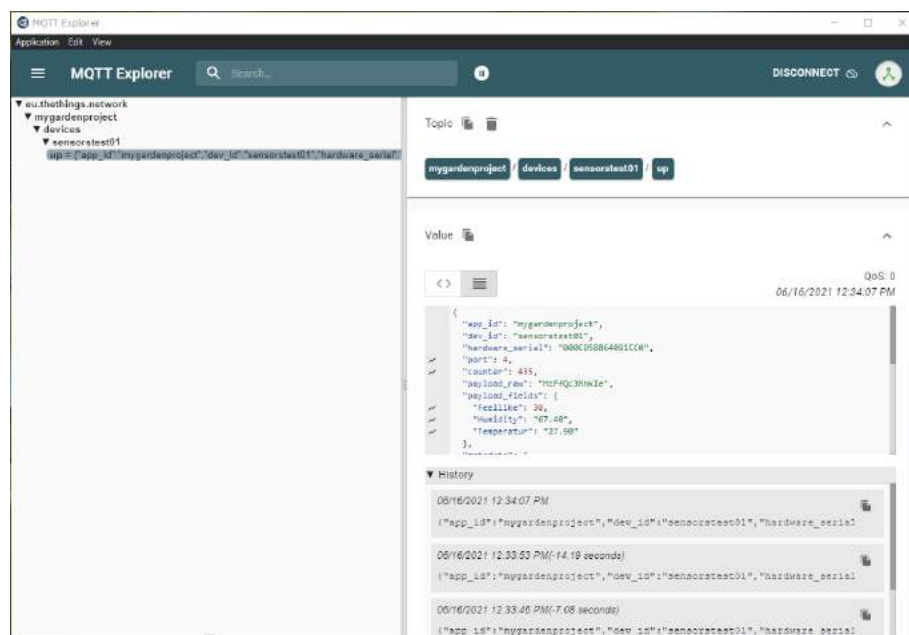


Abbildung 10: Ausgabe der Sensordaten MQTT-Explorer

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 fl = bytes[9] << 8 | bytes[8];
      return {
        "6097c45e1c3320001c7c97ab": Bytes2Float32(t).toFixed(2),
        "6097c45e1c3320001c7c97ac": Bytes2Float32(h).toFixed(2),
        "6097c45e1c3320001c7c97ad": fl
      };
    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 fl = bytes[9] << 8 | bytes[8];
      return {
        "Temperatur": Bytes2Float32(t).toFixed(2),
        "Humidity": Bytes2Float32(h).toFixed(2),
        "Feellike": fl
      };
    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:

[Overview](#)
[Devices](#)
[Payload Formats](#)
[Integrations](#)
[Data](#)
[Settings](#)

APPLICATION DATA
|| pause
🗑 clear

[uplink](#)
[downlink](#)
[activation](#)
[ack](#)
[error](#)

Filters

	time	counter	port		dev id:	payload:	Feellike:	Humidity:	Temperatur:
▲	12:19:36	312	4		sensorstest01	CDCCDC41CDCC86421D	29	"67.40"	
<									
▲	12:19:29	311	4		sensorstest01	CDCCDC41333387421D	29	"67.60"	
<div> <div>Uplink</div> <div>Payload</div> <div>CD CC DC 41 33 33 87 42 1D</div> <div></div> </div> <div> <div>Fields</div> <div> <pre>{ "Feellike": 29, "Humidity": "67.60", "Temperatur": "27.60" }</pre> </div> </div> <div> <div>Metadata</div> <div> <pre>{ "time": "2021-06-16T10:19:29.243218264Z", "frequency": 867.9, "modulation": "LORA", "data_rate": "SF7BW125", "coding_rate": "4/5", "gateways": [{ "gtw_id": "eui-58a0cbfffe800ee1", "timestamp": 1383230580, "time": "2021-06-16T10:19:29.129239082Z", "channel": 0, "rssi": -37, "snr": 9.5 }] }</pre> </div> </div> <div> <div>Estimated Airtime</div> <div>41.216 ms</div> </div>									

Abbildung 12: Datenausgabe TNN

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 4: 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	 https://www.makershop.de/plattformen/esp8266/ttgo-esp32-paxcounter-lora/
Coolty High Pressure Diaphragm Water Pump	Pumpe	Wasser pumpen	Betriebsspannung DC 12V 100PSI 4L/min	 https://amzn.to/3zsKp1O
AZDelivery 4-relay module 5 V	Relais	Steuerung der Pumpe	Eingang 5V/3V Ausgang max 250V/10A AC, 30V/10A DC	 https://amzn.to/3iHNncN

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	Bodenfeuchtigkeitssensor	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 5: 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