

Hochtaunusschule Oberursel
Berufliches Gymnasium
Schwerpunkt Technische Informatik



Hochtaunusschule
Karriere durch Technik

Besondere Lernleistung
als fünftes Prüfungsfach,
im Landesabitur Hessen 2024

Wateringvision

Intelligentes Steuerungssystem für Grünflächenbewässerung

Erstgutachter: Simon Betz
Zweitgutachter: Christian Stückrath

Verfasser:

Denck, Alexander Lionel
01.10.2004
Hannah-Arendt-Str. 40
60438 Frankfurt am Main
info@wateringvision.de
Berufliches Gymnasium, Schwerpunkt Technische Informatik

Abgabetermin: 21.03.2024
Anzahl der Zeichen (inkl. Leerzeichen, ohne Anhang): 95.245
Anzahl der Wörter: 12.056

„Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do.“

Steve Jobs

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken,
die mich während dieser besonderen Lernleistung unterstützt
haben, besonders hervorheben möchte ich:

Mein Vater, Michael, der mich jederzeit gefördert hat und das
komplette Projekt finanziert hat
Meiner Mutter, Petra, welche mich mit ihren Fähigkeiten unter-
stützt hat

Meinen Korrektoren Simon Betz und Christian Stückrath, welche
die BLL erst möglich gemacht haben, mich beraten und unter-
stützt haben

Meinen Großeltern, Alf-Rico Denck und Annerose Staudenmayer,
weil sie so großartige Großeltern sind <3

Alexander Lionel Denck
Frankfurt am Main, 19.03.2024

Inhaltsverzeichnis

Abbildungsverzeichnis	7
Linkverzeichnis	8
Quellenverzeichnis.....	9
1 Einleitung	11
1.1 Aufgabenstellung und Ziel	11
1.2 Motivation für Wateringvision	12
2 Planung	13
2.1 Planung vor Projektumsetzungsbeginn und Weiterentwicklung während der Entwicklung	13
2.2 Zeitplanung inkl. Vergleich zur geplanten Zeitverteilung	14
2.3 Arbeitsprozess.....	16
2.4 Rekapitulation der Planung	17
3 Methodik	18
3.1 Fallstudie/Experteninterview: Besuch im Palmengarten Frankfurt.....	18
3.2 Experteninterview: Systemintegration im Eigenheim	20
3.3 Fallstudie/Experteninterview: Unser Garten	21
3.4 Debugging	22
3.5 Agile Sprintentwicklung	23
3.6 Künstliche Intelligenz in der IT-Entwicklung	25
3.7 Parallele Projektentwicklung.....	27
3.8 Wetterdaten.....	28
4 Umsetzung	29
4.1 Module	29

4.1.1 Konzept der Module.....	29
4.1.2 ESP32	31
4.1.3 Technische Umsetzung des Sensorik-Moduls	32
4.1.4 Softwaretechnische Umsetzung des Sensorik-Moduls	34
4.1.5 Technische Umsetzung des Aktorik-Moduls.....	36
4.1.6 Softwaretechnische Umsetzung des Aktorik-Moduls..	37
4.2 Sensorik und Aktorik.....	38
4.2.1 Magnetventile.....	38
4.2.2 Bodenfeuchtesensor	39
4.2.3 Füllstandssensor für eine Zisterne	40
4.3 Informationstechnische Umsetzung.....	42
4.3.1 Betriebssystem und Server Hardware	42
4.3.2 Kommunikationsprotokolle und Entwicklungswerkzeuge	43
4.3.3 JavaScript als Fullstack-Lösung	44
4.3.4 Aufbau des Backend	45
4.3.5 Arduino C/C++ Dialekt	46
4.4 Netzwerktechnische Umsetzung	47
4.4.1 Domainverwaltung und Netzwerkinfrastruktur	47
4.4.2 MQTT-Protokoll	48
4.4.3 Netzwerksicherheit.....	51
4.5 Benutzeroberfläche	52
4.5.1 Konzept	52
4.5.2 React.....	53
4.5.3 WebSocket.....	54
4.6 Datenabfrage, -verwaltung und -verarbeitung	56
4.6.1 Bright Sky.....	56

4.6.2 MariaDB.....	57
4.6.3 Event-basierte Bewässerung.....	58
4.7 Wassermanagement	59
4.7.1 Wasserquellen	59
4.7.2 Speicherung und Transport des Wassers	59
4.8 Modellhafte Darstellung / Showroom.....	60
5 Ausblick	61
5.1 Mögliche Optimierungen	61
5.1.1 Weiterentwicklung der Website	61
5.1.2 Hardwaretechnische Weiterentwicklung der Module ..	62
5.1.3 Softwaretechnische Weiterentwicklung der Module ...	63
5.1.4 Bessere Netzwerksicherheit (geplant)	64
5.1.5 MongoDB	65
5.1.6 Komplette Installation im Garten	66
5.1.7 Scalable Cloudhosting.....	66
5.1.8 Künstliche Intelligenz zur Datenauswertung	66
5.1.9 Überprüfung des Zisternenfüllstandes.....	67
5.1.10 Präventive Maßnahmen gegen einen Wasserschaden.....	67
5.1.11 Integration von Strommanagment.....	68
6 Fazit	68
Abbildungsquellenverzeichnis.....	72
Eigenständigkeitserklärung	74
Anhang	75

Abbildungsverzeichnis

Abbildung 1 - Karte des Palmengartens aus dem Jahre 1909 im Verhältnis 1:500	18
Abbildung 2 - Unterirdische Zisternenanlage im Palmengarten	19
Abbildung 3 - Bewässerungsmethoden vereinfacht dargestellt	19
Abbildung 4 - erste fehlerhafte Ausführung mit 'localhost' config, zweite erfolgreiche Ausführung mit IPv4 Maske	23
Abbildung 5 - SWE Benchmark, Vergleich der Erfolgsquote beim Lösen von echten GitHub Problemen verschiedener KI Modelle	26
Abbildung 6 - DWD Wetterstation auf einer Deutschlandkarte visualisiert	28
Abbildung 7 - Pin-Layout eines ESP32 WEMOS LOLIN32 Lite	31
Abbildung 8 - Technologieschema eines Sensorik-Moduls	32
Abbildung 9 - Sensordatenabfrage vereinfacht dargestellt	34
Abbildung 10 - Eichmessung des kapazitiven Feuchtigkeitssensor in Wasser	35
Abbildung 11 - Technologieschema eines Aktorik-Moduls	36
Abbildung 12 - Ventilsteuerungslogik vereinfacht dargestellt	37
Abbildung 13 - YL-69 Leitfähigkeitssensor (links) & kapazitiver Sensor v1.2 (rechts)	39
Abbildung 14 - Benutzerzugriff in einem Linux System von der Software bis zur Hardware	42
Abbildung 15 - Vereinfachte Flowchart ähnliche Darstellung des Backend	45
Abbildung 16 - Netzwerkzugehörigkeit der Komponenten, WWW (links), LAN (rechts)	47
Abbildung 17 - Publish-Subscribe-Model dargestellt, Sensor veröffentlicht Temperaturwert Endgeräte empfangen diesen	49
Abbildung 18 - Funktionsweise des Virtual DOM visualisiert	53
Abbildung 19 - Prototyp Aufbau des Showroom Moduls	60

Linkverzeichnis

YouTube-Links:

<https://bit.ly/wv-goals> Web - insert DB - CronJob - select DB - MQTT - ESP
<https://bit.ly/wv-sensorik> ersten Aufbau des Sensorik-Moduls
<https://bit.ly/wv-aktorik> ersten Aufbau der Relais-Fernschaltung
<https://bit.ly/wv-ultra> erste erfolgreiche Inbetriebnahme des Ultraschall-Sensor
<https://bit.ly/wv-ethernet> PC-Ethernet-Arduino HTTP API (ältestes Video)
<https://bit.ly/wv-mqtt> erste erfolgreiche bidirektionale Verbindung zweier ESP via MQTT
<https://bit.ly/wv-webdb> erste erfolgreiche Implementierung einer Web-DB-Verbindung

GitHub-Links:

<https://bit.ly/wv-sensor> Arduino Sensorik Firmware
<https://bit.ly/wv-zisterne> Arduino Zisterne Firmware
<https://bit.ly/wv-zisterne-config> Arduino Zisterne Config
<https://bit.ly/wv-showroom> Arduino Showroom Firmware
<https://bit.ly/wv-action> JavaScript event-basierte Bewässerungssteuerung
<https://bit.ly/wv-broker> JavaScript aedes MQTT-Broker
<https://bit.ly/wv-websocket> JavaScript WebSocket Code für die Showroom-App
<https://bit.ly/wv-weather> JavaScript BrightSky Abfrage und Verarbeitung
<https://bit.ly/wv-weather-json> JavaScript Wetterprognose BrightSky-API into JSON-File
<https://bit.ly/wv-json-db> JavaScript Read JSON-File insert into MariaDB
<https://bit.ly/wv-sensor-db> JavaScript Catch Sensor Data insert into DB

Weitere:

<https://bit.ly/wv-so> StackOverflow-Post zu MariaDB TimeOut Bug
<https://bit.ly/wv-backend> eigen erstellter Backend GPT
<https://bit.ly/wv-electric> „Electric and Electronic Circuits Tutor“ für e-technische Fragen

Quellenverzeichnis

1. 'Expertengespräch mit Herr Bergdoll am 23.11.2023'
2. 'Informationstechnologie Q2 bei Herr Endreß'
3. 'Antwort auf den Prompt „Wie wurdest du gepresstet? Beschreibe deine Funktion selber in einem Satz.“, <https://chat.openai.com/share/c7939efb-b4bc-4a7f-9f26-ee8a1f53e24f>, (Stand 15.03.2024)'
4. 'SWE-Benchmark, „Can Language Models Resolve Real-World GitHub Issues?“, <https://www.swebench.com/#>, (Stand 15.03.2024)'
5. 'AI just officially took our jobs... I hate you Devin, Firestone am 13.03.2024, <https://youtu.be/AgyJv2Qelwk>, (Stand 15.03.2024)'
6. 'Globalmodell ICON, Deutscher Wetterdienst, https://www.dwd.de/DE/forschung/wettervorhersage/num_modellierung/01_num_vorhersagemodelle/icon_beschreibung.html (Stand 03.03.2024)'
7. 'Stationskarte DWD, https://www.dwd.de/DE/fachnutzer/landwirtschaft/appl/stationskarte/_node.html, (Stand 03.03.2024)'
8. 'Preisliste der Openweather API, <https://openweathermap.org/price> (Stand 03.03.2024)'
9. 'Website des Herstellers (Espressif), <https://www.espressif.com/en/products/socs/esp32> (Stand 03.03.2024)'
10. 'Marek Babiuch, Petr Foltýnek, Pavel, Smutnyek, Using the ESP32 Microcontroller for Data Processing, S. 1-4'
11. 'Website des Herstellers (Espressif), <https://www.espressif.com/en/products/socs/esp8266> (Stand 03.03.2024)'
12. 'Technische Informatik Q3 bei Herr Betz'
13. 'eBay Listing des Magnetventils; <https://www.ebay.de/itm/384318597260>; (Stand 05.03.2024); (Es konnte kein reines Datenblatt gefunden werden)'
14. 'Informationstechnologie Q3 bei Herr Geißler'
15. 'Benne de Bakker; Waterproof JSN-SR04T Ultrasonic Distance Sensor with Arduino Tutorial; 28.03.2021 ; https://saliterman.umn.edu/sites/saliterman.umn.edu/files/files/general/arduino_waterproof_jsn-sr04t_ultrasonic_sensor_2_examples.pdf; (Stand 05.03.2024)'
16. 'Wolfgang Ewald; HC-SR04 und JSN-SR04T-2.0 Abstandssensoren; 31.08.2019; <https://wolles-elektronikkiste.de/hc-sr04-und-jsn-sr04t-2-0-abstandssensoren>; (Stand 05.03.2024)'
17. 'Jolle W. Jolles, Broad-scale applications of the Raspberry Pi: A review and guide for biologists, British Ecological Society, S. 2-9'

18. 'Emilio Raggi, Keir Thomas, Sandar van Vugt, Beginning of Linux Sixth Edition, ap-
pres, New York, USA, 2011, S.13-22'
19. 'Preisliste IONOS VPS, <https://www.ionos.de/server/vps> (Stand 03.03.2024)'
20. 'Kostenkalkulator AWS, [https://calculator.aws/#/createCalculator/ec2-enhance-](https://calculator.aws/#/createCalculator/ec2-enhance-ment)
ment bei Region FFM, 4 CPU-Kerne und 8GB RAM (Stand 03.03.2024),'
21. 'Daniel J. Barrett, Ph.D., Richard Silverman, SSH, The Secure Shell: The Definitive
Guide, O'Reilly, Sebastopol, CA, USA, S. 1-14'
22. 'Cameron Newham, Learning the bash Shell, 3rd Edition, O'Reilly, Sebastopol, CA,
USA, S. 23-42'
23. 'Putty Homepage, <https://www.putty.org/>, Stand (03.03.2024)'
24. 'VSCode SSH Extension Dokumentation, [https://code.visualstudio.com/docs/re-](https://code.visualstudio.com/docs/re-mote/ssh)
mote/ssh, (Stand 03.03.2024)'
25. 'Filezilla Homepage, <https://filezilla-project.org/>, (Stand 03.03.2024)'
26. 'Ali Mesbah, Understanding Asynchronous Interactions in Full-Stack JavaScript,
University of British Columbia, Vancouver, BC, Canada, o.J., S.1'
27. 'Azat Mardam, Full Sack JavaScript, apress, 2015, S.137-154'
28. 'Julzer Ali, Node Package Manager, Packet Publishing, Birmingham, 2013, S.18-21'
29. 'npm About Website, <https://www.npmjs.com/about> (Stand 05.03.2024)'
30. 'Website von pm2, <https://pm2.keymetrics.io/>, (Stand 05.03.2024)'
31. 'Basher Modallal, Cron Jobs Explained With Nodejs, [https://medium.com/@bas-](https://medium.com/@bas-har.modallal/cron-jobs-explained-with-nodejs-79099001a6f1)
har.modallal/cron-jobs-explained-with-nodejs-79099001a6f1, (Stand 09.03.2024)'
32. 'Sanmit Charatterjee, What are the Key Pros and Cons of the Arduino Programming
Language?, 2023, <https://emeritus.org/blog/coding-arduino-programming-language/>
(Stand 03.03.2024)'
33. 'Offizielle Arduino Libraries GitHub Repository, <https://github.com/arduino-libraries>
(Stand 03.03.2024)'
34. 'Lukas Schulz, "Datenübertragung mittels MQTT", Bachelorarbeit, Universität Ulm,
2017, S. 5-7'
35. 'Ebd., S. 7-9'
36. 'Ebd., S. 10'
37. 'GitHub Repository von aedes, <https://github.com/moscajs/aedes>, (Stand
08.03.2024)'
38. 'Lukas Schulz, "Datenübertragung mittels MQTT", Bachelorarbeit, Universität Ulm,
2017, S. 5'
39. 'Technische Systeme E2 bei Herr Stückrath'
40. 'Virtual DOM and Internals, <https://legacy.reactjs.org/docs/faq-internals.html>,
(Stand 09.03.2024)'

41. 'Website von React, <https://legacy.reactjs.org/community/team.html>, (Stand 09.03.2024)'
42. 'Most used web frameworks among developers worldwide, as of 2023, <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>, (Stand 09.03.2024)'
43. 'Top 10 Big Companies Using React, <https://career Karma.com/blog/companies-that-use-react/>, (Stand 09.03.2024)'
44. 'The complete guide to WebSockets with React, <https://ably.com/blog/websockets-react-tutorial>, (Stand 09.03.2024)'
45. 'Technische Informatik Q3 bei Herr Betz'
46. 'Informationstechnologie Q2 bei Herr Endreß'
47. 'Open Data Bereich des Climate Data Center, <https://www.dwd.de/DE/leistungen/cdc/climate-data-center.html> (Stand 03.03.2024)'
48. 'Website von BrightSky, <https://brightsky.dev/> (Stand 03.03.2024)'
49. 'Technische Informatik Q4 bei Herr Betz'
50. 'Website von MariaDB; <https://mariadb.org/de/>; (Stand 05.03.2024)'
51. 'MySQL Produkte; <https://www.mysql.com/products/>; (Stand 05.03.2024)'
52. 'Website von phpMyAdmin; <https://www.phpmyadmin.net/>; (Stand 05.03.2024)'
53. 'Deepak Vohra, "Using MongoDB Database", 2016, S. 167-200'
54. 'Ciprian-Octavian Truică, A. Boicea, Ionut Trifan, "CRUD Operations in MongoDB", 2013, S. 347-350'
55. 'Deepak Vohra, "Creating an Apache Hive Table with MongoDB", 2015, S. 405-426'

1 Einleitung

1.1 Aufgabenstellung und Ziel

Das Projekt Wateringvision ist ein IT-System, das auf folgender Leitfrage basiert: „Wie ist es technisch möglich, eine Grünfläche effizient, automatisiert und intelligent zu bewässern?“ Ursprünglich für unseren privaten Garten konzipiert, wurde mir im Verlauf der Entwicklung klar, dass das System auch für andere potenzielle Anwender wie Gärtnereien oder Baumschulen von Interesse sein könnte. Diese Betriebe verfügen zwar über das Fachwissen, um Bewässerungsanlagen effektiv zu verlegen, benötigen jedoch eine

intelligente Steuerungseinheit, die die Ventile effizient und automatisiert betätigt.

Es werden verschiedene Parameter rund um die Pflanzen gesammelt und von einem Algorithmus ausgewertet. Die wesentlichen Parameter sind Wetterprognosen und die Bodenfeuchtigkeit, die durch Sensoren ermittelt, über Mikrocontroller ausgelesen und über das WLAN-Netzwerk an den zentralen Server übermittelt werden. Darüber hinaus werden Benutzerpräferenzen als Parameter berücksichtigt. Benutzer können über eine Webseite alle Prozesse einsehen und steuern.

Die einzigen physischen Komponenten, die bei den Benutzern installiert werden müssen, sind die Sensoren und die Magnetventile; zudem ist ein WLAN-Netzwerk erforderlich. Somit bietet Wateringvision eine maßgeschneiderte, effiziente und benutzerfreundliche Lösung für die automatisierte Bewässerung verschiedenster Grünflächen.

1.2 Motivation für Wateringvision

Meine Begeisterung für IT-Projekte kristallisierte sich während meiner Schulzeit heraus, als ich feststellte, dass diese Arbeit mir nicht nur Freude bereitet, sondern auch meine Fähigkeiten und mein Verständnis für technische Systeme erheblich erweitert. Die Möglichkeit, im Rahmen einer besonderen Lernleistung ein eigenes Projekt zu realisieren, war für mich daher eine willkommene Gelegenheit, meine Leidenschaft weiter zu vertiefen.

Durch meinen Schwerpunkt Technische Informatik habe ich ein fundiertes Verständnis für Informationstechnologie entwickelt. Diese Kenntnisse bilden die Grundlage für mein Projekt Wateringvision, das ich als Chance sehe, Themen zu erforschen, die mich persönlich faszinieren, aber nicht im Kerncurriculum des Beruflichen Gymnasiums enthalten sind. Wateringvision ist mehr als nur ein Schulprojekt; es sollte einen langfristigen Nutzen haben

und einen veralteten Prozess in unserem Haushalt digitalisieren. Die Motivation hinter Wateringvision ist meine tiefe Leidenschaft für die IT, gepaart mit dem Bestreben, etwas Innovatives und Nützliches zu schaffen.

Die Herausforderung, ein umfangreiches Projekt wie Wateringvision zu leiten, war auch ein Mittel für mich, meine Neigung zur Prokrastination zu bekämpfen. Indem ich mich einem so breiten Themenspektrum widmete, wollte ich nicht nur mein Wissen vertiefen, sondern auch herausfinden, welche Bereiche der IT mich am meisten ansprechen. Diese Erkenntnisse könnten entscheidend für meine anstehende Studiengangswahl sein. Die Arbeit an Wateringvision war daher nicht nur eine akademische Übung, sondern auch eine persönliche Entdeckungsreise, auf der ich meine Fähigkeiten, Interessen und beruflichen Ambitionen weiter erforscht habe.

2 Planung

2.1 Planung vor Projektumsetzungsbeginn und Weiterentwicklung während der Entwicklung

Die Vision für Wateringvision hat sich im Projektverlauf stetig weiterentwickelt. Die intensive Auseinandersetzung mit dem Projekt führte zu neuen Blickwinkeln und zu einer Vertiefung meines technischen Verständnisses. Mit jedem neu erschlossenen Thema entstanden frische Ideen für die Umsetzung des Systems.

Ein entscheidender Moment für die Ausrichtung von Wateringvision war ein Besuch im Palmengarten. Ursprünglich war die Planung eines maßgeschneiderten Bewässerungssystems für unser Eigenheim gedacht, ohne dabei eine Skalierbarkeit des Systems in Betracht zu ziehen. Der Gedanke, dass Wateringvision auch außerhalb unseres Grundstücks weitere Interessenten finden könnte, kam erst mit dem Besuch im Palmengarten.

Seither besteht der Ansatz, ein universell einsetzbares System zu entwickeln. Dadurch gewannen Themen wie Netzwerksicherheit, die aufgrund der geplanten Abkapselung im lokalen Netzwerk zunächst wenig Beachtung erhielten, nun an Bedeutung. Zudem sollte das System von Personen bedient werden, die nicht an der Entwicklung beteiligt waren, was Benutzerfreundlichkeit und Sicherheitsaspekte in den Vordergrund rückte.

2.2 Zeitplanung inkl. Vergleich zur geplanten Zeitverteilung

Während der Entwicklung von Wateringvision hat sich die Idee des Projekts signifikant geändert, was unweigerlich zu Abweichungen von der anfangs veranschlagten Zeitplanung führte. Es ist wichtig zu betonen, dass alle Stundenangaben nur ungefähre Zeitangaben darstellen. Bei der Entwicklung eines solchen Projekts sind es vor allem die Ideen, die das System letztendlich ausmachen. Diese Ideen kamen zumeist nicht während der direkten Arbeitszeit an Wateringvision, sondern zufällig im Alltag.

Besonders das Erlernen benötigter neuer Fähigkeiten habe ich in andere private Projekte ausgelagert, was in Kapitel "##3.9 Parallele Projektentwicklung" näher erläutert wird. Zudem fließt Debugging nicht in die Stundenberechnung ein, da unvorhergesehene Fehler auftreten können, deren Behebungsdauer nicht kalkulierbar ist. Über die Monate haben sich jedoch sowohl die Veranschlagungen als auch die Namensgebungen geändert.

Arbeitsdauer: Vergleich veranschlagte und reale Dauer

<u>Themenbereich</u>	<u>Veranschlagte Dauer</u>	<u>Reale Dauer</u>
Planung	15 Stunden	15 Stunden
Algorithmus	15 Stunden	25 Stunden
Website	15 Stunden	15 Stunden

Wetterdatenbank	6 Stunden	6 Stunden
Ventilmodul	10 Stunden	10 Stunden
Füllstandssensor	5 Stunden	5 Stunden
Wasserdurchlaufssensor	6 Stunden	ausgeplant
zentrale Steuerungseinheit	6 Stunden	ausgeplant
Vorbereitung Kolloquium	10 Stunden	5 Stunden
Wassernetzwerk	12 Stunden	4 Stunden
Thesis	nicht eingeplant	15 Stunden
Insgesamt	100 Stunden	100 Stunden

Die Entwicklung des Backend hat mit ca. 25 Stunden länger gedauert als geplant, da es der größte Teil von Wateringvision ist. Die Schätzung für die Website waren mit zutreffend, da diese als letzte Komponente umgesetzt wurde und daher nicht häufig angepasst werden musste. Die Wetterdatenbank und der Füllstandssensor wurden jeweils innerhalb der geplanten Zeit fertiggestellt.

Die Vorbereitung auf das Kolloquium wurde mit überschätzt; tatsächlich benötigte ich aufgrund meiner intensiven Einarbeitung nur noch 5 Stunden. Die angesetzten 10 Stunden für das Ventilmodul wurden eingehalten, obwohl sich der Anwendungsfall geändert hat. Es ging dabei um die Verpackung der Hardware wie Mikrocontroller und Relais.

Durch die Transformation von einem maßgeschneiderten IoT-Projekt zu einem IT-System und den geringeren tatsächlichen Einbau im eigenen Garten, ebenfalls bedingt durch die Wetterlage im Winter und Frühling, hat sich

die veranschlagte Dauer von 12h auf 5h reduziert. Das Thema Wasserdurchlaufssensor wurde komplett gestrichen, da dieser als nicht mehr notwendig erachtet wurde. Zudem wurde die Arbeit an der Thesis, die in etwa 15 Stunden beanspruchte, in der anfänglichen Planung nicht berücksichtigt.

2.3 Arbeitsprozess

Im Zuge der Entwicklung wurde ein methodischer Ansatz verfolgt, der sich durch die Unterteilung in spezifische Projektziele und wesentliche Meilensteine auszeichnete. Essenzielle Etappen umfassten die Etablierung einer Kommunikation zwischen Mikrocontrollern und einem Server, die Konfiguration eines Linux-Servers, die Integration einer Wetterdatenquelle, das Speichern und Abrufen von Daten in einer Datenbank, die Entwicklung und das Hosting einer Webseite sowie die Implementierung von Schnittstellen zwischen Frontend, Backend, Datenbanken und Mikrocontrollern. Zudem wurden Prototypen für Sensorik- und Aktorik-Module entwickelt, deren Dauerbetrieb im eigenen Garten getestet wurde, gefolgt von der Erstellung einer ersten Webbenutzeroberfläche und der Entwicklung eines Beta-Algorithmus für die Datenauswertung.

Die Projektumsetzung erfolgte weitestgehend sequenziell, wobei gleichzeitig an verschiedenen Komponenten gearbeitet wurde. Zur Sicherstellung des Fortschritts wurden regelmäßig Checkpoints definiert. Ein exemplarisches Vorgehen wurde anhand der Schnittstellenentwicklung demonstriert: Eine einfache Webseite mit einem Toggle-Schalter wurde erstellt, dessen Status in einer Datenbank gespeichert und zu festgelegten Zeiten abgerufen wurde, an einen Mikrocontroller übertragen und umgesetzt wurde, wobei als Test-Aktorik die BUILTIN_LED diente.

Die Architektur und die Komponenten des Systems wurden iterativ an die steigenden Anforderungen angepasst. Initial wurde mit einem Raspberry Pi gearbeitet, welcher durch ein 1GB RAM VPS ersetzt wurde. Dieser wurde

wegen der Mindestanforderung von React, nämlich mindestens 4GB RAM, durch einen 8 GB RAM VPS ersetzt. Die Auswahl der Mikrocontroller entwickelte sich von Arduino-Boards mit Ethernet Shield über ESP8266 bis hin zu ESP32-Modulen, um Bluetooth-Optionen zu berücksichtigen. Während der Entwurfs- und Testphase wurden unvorhergesehene Bauteile wie Logik Level Konverter als notwendig identifiziert und der Entwurf entsprechend angepasst.

Die Implementierungsphase erwies sich als kritisch für die Überprüfung der Nachhaltigkeit von Entwürfen, Entwicklungen und Tests. Insbesondere in der Software wurden mehr Fehler identifiziert als in der Hardwarekomponente. Aktuell ist eine Bewässerungslinie im Einsatz, deren nächste Generation unter Berücksichtigung neuer Erkenntnisse weiter optimiert wird. Die Entwicklung des Datenauswertungs-Algorithmus befindet sich noch in einer frühen Phase und wird in den kommenden Monaten intensiv weiterentwickelt. ##KI Auswertung

Abschließend lässt sich feststellen, dass die Realisierung des Wateringvision-Projekts kein linearer Prozess war. Vielmehr war es ein dynamisches Vorhaben, das durch Schwankungen in Motivation, das Auftreten von Herausforderungen, das Erreichen oder Ausbleiben von Erfolgserlebnissen sowie das Einfließen neuer Einflüsse und Ideen geprägt wurde.

2.4 Rekapitulation der Planung

Im Zuge der Rekapitulation der Planung von Wateringvision hat sich gezeigt, dass es nicht "die eine Planung" gibt. Vielmehr hat sich der Ansatz zur Bewältigung verschiedener Themenblöcke kontinuierlich verändert und verbessert – ein Prozess, der beständig fortschreitet.

Eine weitere wichtige Erkenntnis war die Notwendigkeit, den aktuellen Stand des Projekts stetig zu überdenken, wenn sich ein besserer Ansatz

offenbarte. Ein exemplarisches Beispiel hierfür war der Wechsel von SQLite zu einem server-client-basierten DBMS. Obwohl alle Softwarekomponenten ursprünglich auf SQLite abgestimmt waren, ermöglichte erst dieser Wechsel eine nachhaltige Weiterentwicklung des Systems.

Zum Vergleich: Wateringvision startete mit der simplen Idee, Magnetventile an einen Mikrocontroller anzuschließen, der durch eine Steuerbox mit Öffnungs- und Schließmechanismen gesteuert werden sollte. So hatte ich Herrn Endreß eine Mail am 11. Juli 2023 geschrieben, wo ich fragte, ob ein Mikrocontroller gesteuertes Projekt als BLL möglich wäre. Anfänglich waren weder ein Server, Sensordaten, Wetterdaten noch eine Webbenutzeroberfläche geplant. Diese Elemente wurden erst im Laufe des Projekts, durch stetiges Weiterdenken und Erweitern des ursprünglichen Konzepts, integriert.

3 Methodik

3.1 Fallstudie/Experteninterview: Besuch im Palmengarten Frankfurt

Im Rahmen einer Fallstudie für das Wateringvision-Projekt stand ein Besuch beim technischen Leiter des Frankfurter Palmengartens an, Herrn Bergdoll, um Einblicke in etablierte Bewässerungstechniken zu gewinnen. Der Aufbau des Palmengartens ist in Abbildung ##1 zu erkennen.



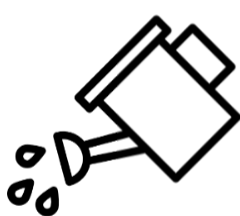
Abbildung 1 - Karte des Palmengartens aus dem Jahre 1909 im Verhältnis 1:500

Die Tour bot einen tiefen Einblick in die verschiedenen Bewässerungssysteme. Besonders beeindruckend war die Vielfalt der Bewässerungstechniken, die für die unterschiedlichen Pflanzensammlungen angewendet wurden. Es wurde deutlich, dass jede Pflanzenart spezifische Anforderungen an die Bewässerung hat, was die Komplexität der Wasserversorgung in einem so vielfältigen Garten wie dem Palmengarten unterstreicht.

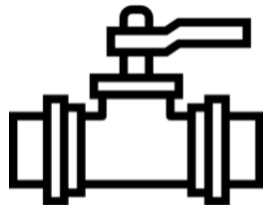


Abbildung 2 - Unterirdische Zisternenanlage im Palmengarten

Der Technische Leiter des Palmengartens in Frankfurt am Main, Herr Bergdoll, hat Bewässerungsarten in vier Kategorien eingeteilt: händische, manuelle, zeitgesteuerte und intelligente Bewässerung (siehe Abbildung ##3).



1. Händisch



2. Manuell



3. Zeitgesteuert



4. Intelligent

Abbildung 3 - Bewässerungsmethoden vereinfacht dargestellt

An erster Stelle steht die händische Bewässerung, die traditionellste Form, bei der Gärtner die Pflanzen direkt mit Gießkannen oder Gartenschläuchen versorgen. Diese Methode ist stark von menschlicher Arbeit abhängig und erfordert ein hohes Maß an Aufmerksamkeit für jede Pflanze. Ab der manuellen Bewässerung muss eine Bewässerungsanlage aufgebaut sein. Die Bewässerungskategorien unterscheiden sich nur im Bedienen der Ventile. Bei der manuellen Bewässerung werden die Ventile von Menschen gesteuert.

Die dritte Kategorie, die zeitgesteuerte Bewässerung, repräsentiert einen ersten Schritt in Richtung Automatisierung. Hierbei werden die Ventile der Bewässerungsanlage nach einem festgelegten Zeitplan automatisch geöffnet und geschlossen, unabhängig von den tatsächlichen Bedürfnissen der Pflanzen.

Die intelligente Bewässerung ist die fortschrittlichste Kategorie, die durch Systeme wie Wateringvision umgesetzt wird. Hier werden IT-Technologien genutzt, um basierend auf Echtzeit-Daten wie Wetterverhältnisse und Bodenfeuchtigkeit eine optimierte Bewässerung zu berechnen. Die Ventile können ohne menschliche Aufmerksamkeit gesteuert werden.

Im Palmengarten werden aktuell überwiegend händische und manuelle Bewässerungsmethoden eingesetzt, wie Herr Bergdoll feststellt.¹

Im Gespräch mit Herrn Bergdoll präsentierte ich das Konzept von Wateringvision. Wir diskutierten über die Möglichkeiten, wie ein intelligentes Bewässerungssystem zur Effizienzsteigerung und Ressourcenoptimierung beitragen könnte. Er zeigte sich interessiert, besonders an der Fähigkeit des Systems, Bewässerungspläne auf der Grundlage von Echtzeit-Daten anzupassen. Die Erkenntnisse aus diesem Besuch sind von großem Wert für die Weiterentwicklung von Wateringvision. Sie haben nicht nur den Bedarf einer anpassungsfähigen und datengesteuerten Bewässerung gezeigt, sondern auch potenzielle Herausforderungen und Anpassungsbedarf in einem professionellen Anwendungsumfeld.

3.2 Experteninterview: Systemintegration im Eigenheim

Bei der Inspektion der vorhandenen Anlagen im Keller unseres Hauses wurden mehrere Schlüsselkomponenten identifiziert, die für die Implementierung des Systems essenziell sind. Zentral ist hierbei die Zisterne,

¹ Expertengespräch mit Herr Bergdoll am 23.11.2023

die als primäre Wasserspeicherquelle dient. Ein weiterer Aspekt der Fallstudie war das Auskünden der Gartenhütte, die sich als Standort für die Module eignet. Ein Experteninterview mit meinem Vater, der Strom- und Wasserleitungen für unser Haus und den Garten selbst gelegt hat, war äußerst wertvoll für die Projektplanung. Sein detailliertes Wissen ermöglichte die effektive Umsetzung des Wateringvision-Systems im Eigenheim.

3.3 Experteninterview: Unser Garten

Grafikidee: Aufbauplan vom Garten

Unser Garten gliedert sich in drei Höhenebenen, die jeweils unterschiedliche Anforderungen an das Bewässerungssystem stellen: Die erste Ebene umfasst die Gartenhütte und den angrenzenden Gemüsegarten. Die zweite Ebene ist durch einen etwa 25m² großen Rasen charakterisiert, flankiert von Pflanzenstreifen an den östlichen und westlichen Seiten. Diese Pflanzenstreifen beherbergen eine Vielfalt an Pflanzen, hier bietet sich die Verwendung von Tropfschläuchen an, um die Wasseraustrittsrate individuell anzupassen. Auf der dritten und höchsten Ebene befinden sich eine Terrasse sowie eine kleinere Rasenfläche von etwa 5 Quadratmetern. In diesem Bereich sind zwei separierte Bewässerungsgruppen umgesetzt.

Im Rahmen dieser Fallstudie wurde ein Experteninterview mit meiner Mutter durchgeführt, die als Hauptverantwortliche für die Gartenbewässerung eine fundierte Kenntnis über die spezifischen Bedürfnisse der unterschiedlichen Gartenbereiche besitzt.

Aktuell ist im Rahmen des Wateringvision-Projekts lediglich eine Bewässerungslinie in Betrieb. Diese Beschränkung ist primär auf zeitliche Restriktionen sowie die Wetterbedingungen im Winter und Frühjahr zurückzuführen, welche eine umfassende Implementierung vor der Deadline der BLL limitieren. Trotz dieser vorübergehenden Einschränkung ist es langfristig

definitiv vorgesehen, den gesamten Garten mit der entwickelten Hardware auszurüsten.

3.4 Debugging

Das Identifizieren eines Fehlers kann viel Zeit in Anspruch nehmen, da die Fehlerquelle nicht immer offensichtlich ist. Je größer und komplexer das betroffene System, desto schwieriger kann es werden, den Ursprung des Fehlers zu finden. Um effektiv zu debuggen, ist es essenziell, die potenziellen Fehlerquellen einzugrenzen. Das betroffene System wird dazu in immer kleinere Teile unterteilt, bis die Fehlerquelle identifiziert werden kann. Besonders herausfordernd wird es, wenn die Kommunikation zwischen Soft- und Hardwarekomponenten fehlerhaft verläuft.

Eine der einfachsten, aber effektiven Methoden ist der Einbau von Konsolenausgaben an kritischen Systempunkten. Dies ermöglicht es, nachzuvollziehen, an welchem Punkt das System aufhört, richtig zu arbeiten. Da man jedoch nicht überall Konsolenausgaben einbauen kann, greift man auf Log-Dateien zurück, welche die Ereignisse aufzeichnen und in Dateien abspeichert.

Die Nutzung von künstlicher Intelligenz zur Fehleridentifikation brachte gemischte Erfahrungen. Während Syntaxfehler gut von einer KI gefunden werden können, kann es vorkommen, dass die KI selbst Fehler einbaut, ohne diese als solche zu erkennen, oder Fehler einfach nicht kennt.

Vermeehrt sind Bugs aufgetreten, wenn Techniken angewendet wurden, die noch nicht vollständig verstanden sind. Es ist daher ratsam, sich mit den Grundlagen der Technik auseinanderzusetzen, um den Fehler selbst finden zu können. Manchmal kann simples Googeln einer Fehlermeldung helfen, allerdings ist dies nicht immer der Fall, vor allem wenn man nicht in der Lage ist, den Fehler zu beschreiben.

Ein konkretes Beispiel für ein Debugging-Problem war ein JavaScript, das eine JSON-Datei in eine Datenbank überführen sollte und konstant abstürzte, weil es keine Verbindung mit der Datenbank aufbauen konnte (siehe Abb. 4). Nachdem keine der üblichen Methoden

funktioniert hatte, stellte ich eine Frage auf StackOverflow. Tatsächlich konnte ein Nutzer den Fehler aufzeigen: Das JavaScript hatte 'localhost' als Host-Adresse für MariaDB und versuchte damit, eine IPv6-Verbindung aufzubauen, während MariaDB, in diesem Fall, nur IPv4-Verbindungen annimmt. Die Lösung war es 'localhost' durch die IPv4 Maske '127.0.0.1' zu ersetzen und somit die IPv4 Verbindung zu erzwingen. Mein StackOverflow-Post kann unter folgender URL eingesehen werden: <https://bit.ly/wv-so>

Zusätzlich muss das System auch auf fehlerhafte Nutzereingaben vorbereitet sein. Dafür gibt es verschiedene Ansätze. Eine von mir präferierte Methode ist, präventiv falsche Nutzereingaben zu verhindern, was besonders gut im UI/UX-Design umgesetzt werden kann, beispielsweise durch die Verwendung eines Sliders anstatt eines Texteingabefeldes.

3.5 Agile Sprintentwicklung

Agiles Management ist gekennzeichnet durch seine Flexibilität und das fortlaufende Feedback. Diese Prinzipien erwiesen sich als ideal, da sie es

```
alexpi@alexpi:~/weather $ node JSON_into_DB.js
Error: SqlError: (conn=-1, no: 45028, SQLState: HY000) retrieve
ve connection from pool timeout after 10003ms
  (pool connections: active=0 idle=0 limit=5)
    at module.exports.createError (/home/alexpi/node_modules/m
ariadb/lib/misc/errors.js:64:10)
    at Pool._requestTimeoutHandler (/home/alexpi/node_modules/
mariadb/lib/pool.js:349:26)
    at listOnTimeout (node:internal/timers:569:17)
    at process.processTimers (node:internal/timers:512:7) {
  sqlMessage: 'retrieve connection from pool timeout after 100
03ms\n' +
    '  (pool connections: active=0 idle=0 limit=5)',
  sql: null,
  fatal: false,
  errno: 45028,
  sqlState: 'HY000',
  code: 'ER_GET_CONNECTION_TIMEOUT'
}
alexpi@alexpi:~/weather $ nano JSON_into_DB.js
alexpi@alexpi:~/weather $ node JSON_into_DB.js
Data inserted successfully.
```

Abbildung 4 - erste fehlerhafte Ausführung mit 'localhost' config, zweite erfolgreiche Ausführung mit IPv4 Maske

erlaubten, schnell auf neue Erkenntnisse und sich ändernde Anforderungen zu reagieren. In diesem Projekt war die Funktionalität eines Teilsystems das oberste Ziel, wobei jeder Teil in Grundfunktionen unterteilt und nach ihrer Wichtigkeit in Sprints abgearbeitet wurde.

Ein entscheidender Vorteil der agilen Entwicklung liegt in ihrer Nicht-Linearität. Während der Sprints können sich die Ziele dynamisch anpassen, basierend auf den neuesten Ideen und Erkenntnissen.² Jeder Sprint brachte einen Zuwachs an Fähigkeiten und Perspektiven, wodurch die folgenden Entwicklungsteile aus neuen Blickwinkeln bewertet werden konnten.

Obwohl die agile Sprintentwicklung viele Vorteile bietet, gibt es auch Herausforderungen. Eine davon ist die Gefahr, den Überblick zu verlieren. Es ist entscheidend, die Sprintziele stets im Auge zu behalten, um sicherzustellen, dass das Projekt auf Kurs bleibt. Zudem erfordert die Gesamtarchitekturplanung eine ganzheitliche Betrachtung des Systems – ein "Big-Picture"-Ansatz, der über die kleinteilige Sprintarbeit hinausgeht.

Ein Sprintziel war, dass man über eine Website Dateien eingeben kann, welche in einer Datenbank abgespeichert werden. Diese Daten sollen zu einer festgelegten Zeit aus der Datenbank wieder abgerufen werden und dann über MQTT an ein ESP verschickt werden. Dabei werden alle Komponenten auf das minimale reduziert, beispielsweise wird nur ein einfacher boolean-Wert abgespeichert oder eine CSS-freie Website erstellt, den bei solch einem Sprintziel geht es darum, die Kommunikation dieser einzelnen Komponenten zu testen. Die erste erfolgreiche Ausführung des Sprintziel wurde in einem Video, was unter <https://bit.ly/wv-goals> aufrufbar ist, festgehalten.

² Informationstechnologie Q2 bei Herr Endreß

Bei der Hardware wurde ebenfalls ein Prototyp-Sprint-Ansatz angewandt. Bei der Entwicklung eines Moduls wurden anfänglich die Grundbausteine, also ein ESP32 und ein Sensor mit Jumper Cables und einem Breadboard zusammengesteckt. Als die Grundfunktion gegeben war, wurden weitere Komponenten, wie ein LCD-Display mit eingebaut. Erst als alle gewünschten Komponenten funktionabel waren, wurden die Bauteile auf einer Lochrasterplatine fest verlötet.

3.6 Künstliche Intelligenz in der IT-Entwicklung

Künstliche Intelligenz spielte eine Schlüsselrolle bei der Software-Entwicklung, über 2.000 Prompts haben es möglich gemacht, Wateringvision in dieser Dimension zu realisieren. Künstliche Intelligenz ist aus unserem täglichen Leben nicht mehr wegzudenken und repräsentiert den „State of the Art“ in der IT-Entwicklung. Ihr Einsatz lässt sich in zwei Kernkompetenzen aufteilen: Beratung und Programmierung.

Dafür habe ich einen GPT entwickelt, dieser beschreibt sich selber wie folgt: „Ich bin ein spezialisiertes ChatGPT, das als Programmierer, Softwareberater und Lehrer dient, um bei der Entwicklung eines Projekts namens "Wateringvision" zu helfen, indem ich Code in JavaScript, JSX/React schreibe und erkläre, basierend auf deinem Hintergrund in OOP und Java.“³

Besonders hilfreich ist der Vergleich neuer Konzepte zur objektorientierten Programmierung in Java, weil ich mit diesen Konzepten vertrauter bin. Durch die Nutzung dieses GPT können deutlich kürzere Prompts benutzt werden und die künstliche Intelligenz weiß, wie sie die Anforderungen umsetzen und erklären soll. Der GPT ist <https://bit.ly/wv-backend> aufrufbar.

³ Antwort auf den Prompt „Wie wurdest du gepresset? Beschreibe deine Funktion selber in einem Satz.“, <https://chat.openai.com/share/c7939efb-b4bc-4a7f-9f26-ee8a1f53e24f>, (Stand 15.03.2024)

Beim Erlernen neuer Programmierkonzepte kam die KI beratend und erklärend zum Einsatz. Zunächst wurden die Grundkonzepte erklärt und solange Rückfragen gestellt, bis ein grundsätzliches Verständnis der Grundkonzepte angenommen wurde. Daraufhin wurde eine Roadmap für die Implementierung der jeweiligen Features erstellt.

In der Programmierphase wurde Künstliche Intelligenz eingesetzt, um Code-Snippets zu generieren. Diese Snippets durchliefen Tests und Anpassungen, bis sie in das Hauptsystem implementiert werden können. Es ist hervorzuheben, dass eine Abhängigkeit von ausschließlich durch ChatGPT generiertem Code nicht ratsam ist. Ein fundamentales Verständnis der generierten Inhalte ist essenziell, um den Code verstehen zu können. Ferner bietet ChatGPT nicht durchgängig Unterstützung beim Debugging, ein Thema, das im Kapitel "##3.4 Debugging" näher behandelt wird.

Meiner Erfahrung nach ist das GPT-4 Modell genial zum Code generieren, aber beim Debuggen hat es enorme Defizite. KI-Modelle basieren zu einem großen Teil auf Datensätzen aus dem Internet, wobei viele auch viele fehlerhafte Codes dabei sind.

Derzeit kann das GPT-4 Modell, laut dem SWE-Benchmark, nur 1,74% der

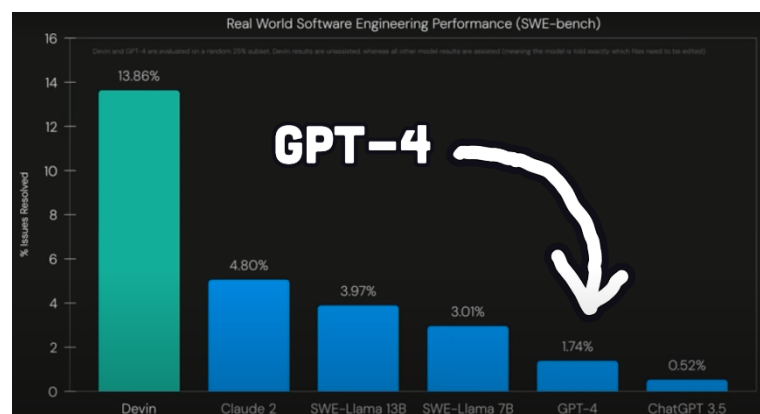


Abbildung 5 - SWE Benchmark, Vergleich der Erfolgsquote beim Lösen von echten GitHub Problemen verschiedener KI Modelle

echten GitHub Probleme erfolgreich lösen. Im Vergleich, dass GPT-3.5 Modell schafft im selben Benchmark nur 0,52%. Hingegen das neuentwickelte Devin-Modell, welches auf dem GPT-4 Modell aufbaut, eine vollständige Entwicklungsumgebung, zur Verfügung hat, schafft bereits 13,86%.⁴⁵

Für Fragen im Bereich Elektrotechnik wurde der GPT „Electric and Electronic Circuits Tutor“ befragt, welcher unter <https://bit.ly/wv-electric> aufrufbar ist.

3.7 Parallele Projektentwicklung

Während meiner Ausbildung im Bereich der technischen Informatik habe ich ein grundlegendes Verständnis für informationstechnologische Konzepte entwickelt. Allerdings erforderte die praktische Umsetzung von Wateringvision, die auch moderne Konzepte umfassen sollte, den Einsatz zahlreicher mir bislang unbekannter Konzepte. Es besteht ein Unterschied zwischen dem Verständnis eines Konzeptes und der Fähigkeit, dieses erfolgreich zu implementieren. Angesichts der Begrenzung auf 100 Stunden für eine besondere Lernleistung war es mir nicht möglich, alle erforderlichen Techniken im Rahmen des Projekts zu erlernen.

Um diese Herausforderung zu bewältigen, verlagerte ich das Erlernen, Testen und Erforschen neuer IT-Konzepte in private Projekte. Ein Beispiel dafür ist die Auseinandersetzung mit React, die ich außerhalb von Wateringvision erlernte. Dies geschah unter anderem durch die Entwicklung eines File-Sharing-Systems für meinen Vater, oder einem Trinkgeld-Managementsystem, für meinen ehemaligen Arbeitgeber. Zusätzlich setzte ich mir

⁴ SWE-Benchmark, „Can Language Models Resolve Real-World GitHub Issues?“, <https://www.swebench.com/#>, (Stand 15.03.2024)

⁵ AI just officially took our jobs... I hate you Devin, Firestone am 13.03.2024, <https://youtu.be/AgvJv2Qelwk>, (Stand 15.03.2024)

kleinere Mini-Projekte als Lernziele und arbeitete daran, bis ich genug Verständnis für ein Thema hatte, um es in Wateringvision umzusetzen.

3.8 Wetterdaten

Wetterdaten spielen eine zentrale Rolle in der intelligenten Bewässerungsplanung. Faktoren wie Niederschlag, Temperatur und Luftfeuchtigkeit haben direkten Einfluss auf den Wasserbedarf der Pflanzen. Eine effektive Bewässerung berücksichtigt diese Parameter, um sowohl Über- als auch Unterbewässerung zu vermeiden und den Wasserbedarf präzise zu decken.

Der Deutsche Wetterdienst wurde als Quelle für Wetterdaten ausgewählt, da er als Bundesamt für seine Zuverlässigkeit und Genauigkeit bekannt ist. Der DWD nutzt das ICON-Wettermodell für seine Wettervorhersagen, welches sich durch hohe Präzision und umfassende Datenabdeckung auszeichnet. Die Vorteile des ICON-Modells liegen in seiner Fähigkeit, komplexe Wetterdynamiken zu modellieren und genaue Vorhersagen zu ermöglichen.⁶ Der DWD betreibt ein deutschlandweites Netz von Wetterstationen (siehe Abbildung ##4), welche als Datenquelle ausgewählt werden können. Für die private Umsetzung von Wateringvision wurde die

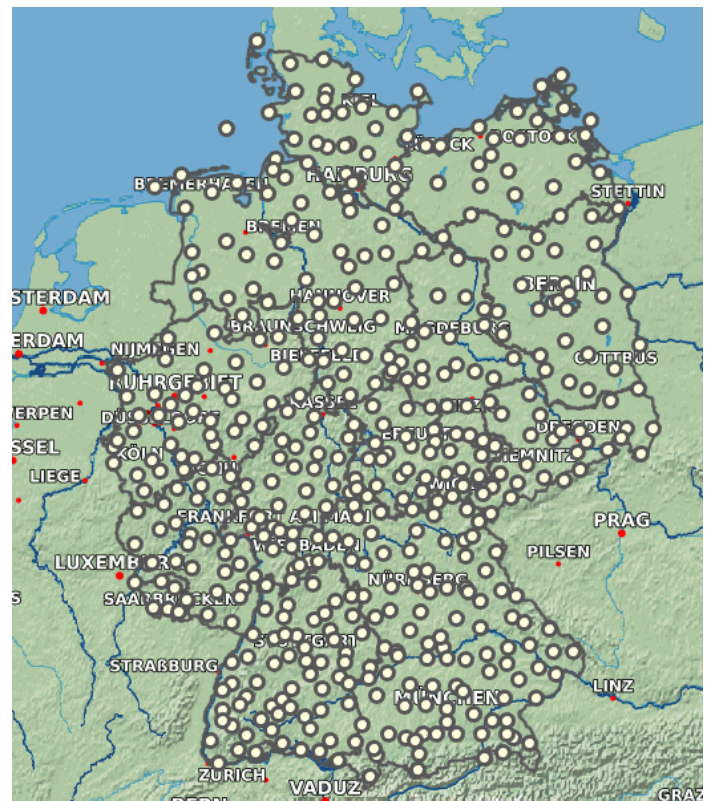


Abbildung 6 - DWD Wetterstation auf einer Deutschlandkarte visualisiert

⁶ Globalmodell ICON, Deutscher Wetterdienst, https://www.dwd.de/DE/forschung/wettervorhersage/num_modellierung/01_num_vorhersagemodelle/icon_beschreibung.html (Stand 03.03.2024)

DWD-Wetterstation am Frankfurter Palmengarten als nächstgelegene Station gewählt.⁷

Im Gegensatz zu Daten von privaten Wetterdiensten, die oft in ihrem kostenlosen Zugriff eingeschränkt sind (z.B. durch begrenzte Wettervorhersagen oder eine limitierte Anzahl an Anfragen)⁸, bietet der DWD umfassende und frei zugängliche Daten. Diese Aspekte unterstreichen die Entscheidung für den DWD als primäre Datenquelle für Wateringvision. Die technische Umsetzung der Datenabfrage wird im Kapitel ## „Bright Sky“ beschrieben.

4 Umsetzung

4.1 Module

4.1.1 Konzept der Module

Die benötigte Technik, welche die Ansteuerung der Sensorik und Aktorik durch das IT-System ermöglicht, wird in sogenannten Modulen verbaut. Die Sensorik umfasst Feuchtigkeits- und Füllstandsensoren für Zisternen, die Aktorik beinhaltet Magnetventile.

Zur Einbindung dieser Technik wird ein Mikrocontroller, speziell ein ESP32, verwendet. Es wurde ein modulares Konzept entwickelt, das sich in zwei Kategorien teilt: Module für Sensorik und Module für Aktorik. Ursprünglich wurde versucht, ein universelles Modul zu entwickeln, das sowohl für Sensorik als auch für Aktorik einsetzbar ist. Jedoch erwies sich dies aufgrund der unterschiedlichen Anforderungen der beiden Systeme als nicht effizient umsetzbar.

⁷Stationskarte DWD, https://www.dwd.de/DE/fachnutzer/landwirtschaft/appl/stationskarte/_node.html, (Stand 03.03.2024)

⁸ Preisliste der Openweather API, <https://openweathermap.org/price> (Stand 03.03.2024)

Damit die Module auseinandergehalten werden können, wurden sie nach Städten benannt. Dieser Städtenamen wird ebenfalls für einen MQTT-Username verwendet. Anfangs wurden die Module durchnummeriert, aber Städtenamen haben sich als besser merkbar herausgestellt.

Alle Bauteile werden auf einer Lochrasterplatine verbaut und verkabelt. Derzeit existiert noch kein witterungsfestes Gehäuse, was die Einsatzorte auf trockene Umgebungen eingegrenzt. Langfristig ist geplant, eigene PCB-Platinen zu entwerfen und die Gehäuse via 3D-Drucker zu produzieren. (## Verweis Module 2. Gen) Ein wesentliches Ziel des Konzepts ist es, den Aufbau der Module nach dem Plug-and-Play-Prinzip zu gestalten. Dies bedeutet, dass alle Verbindungen, wie die Spannungsversorgung oder Sensoranschlüsse, über Steckverbindungen realisiert werden. Dadurch ist ein schneller Auf- und Abbau möglich und es können je nach Bedarf unterschiedlich lange Kabel verwendet werden.

In der Anfangsphase wurde erwogen, Systeminformationen mittels mehrerer Status-LEDs oder einer RGB-LED anzuzeigen. Nachdem jedoch die Erfahrung mit einer Smart-Home-Steckdose und der damit verbundenen Notwendigkeit, ein Handbuch zur Interpretation der Statusfarben zu benutzen, gemacht wurde, entfiel dieser Ansatz. Stattdessen ist die Wahl auf ein 16 Zeichen auf zwei Zeilen LCD-Display gefallen, das Statusmeldungen benutzerfreundlich mithilfe von menschlicher Sprache darstellen kann.

4.1.2 ESP32

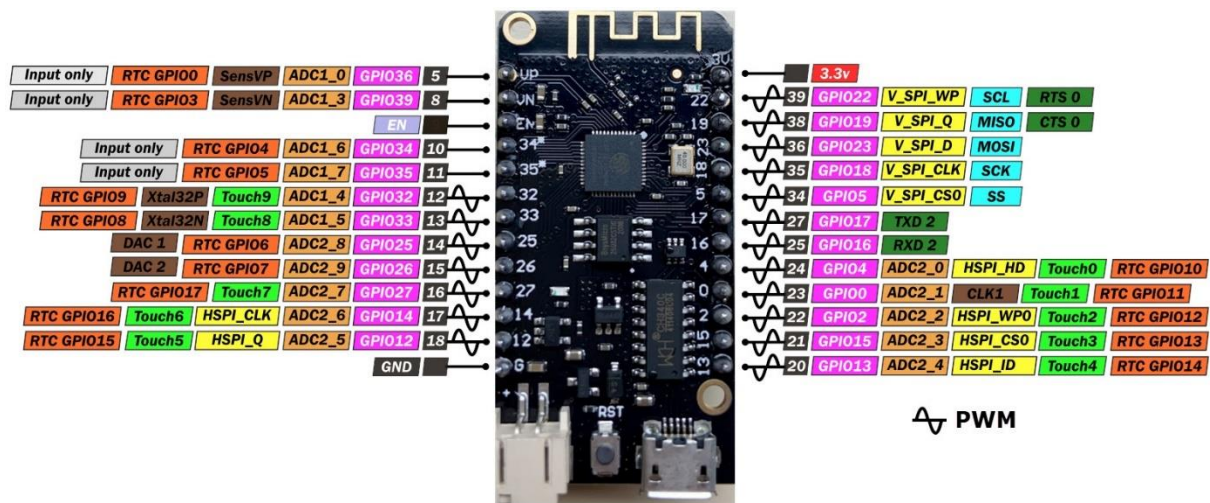


Abbildung 7 – Pin-Layout eines ESP32 WEMOS LOLIN32 Lite

Der ESP32 ist ein kostengünstiger Mikrocontroller, der sich durch seine integrierte 2.4 GHz WLAN- und Bluetooth-Funktionalität auszeichnet.⁹ Sein 32-Bit Dual-Core-Prozessor arbeitet mit einer Taktfrequenz von 240MHz, was ihn zu einer leistungsstarken Wahl für vielfältige Anwendungen macht. Die Betriebsspannung des ESP liegt bei 3,3 Volt.¹⁰ Es wird die WEMOS LOLIN32 Lite Version genutzt, welches Pin-Layout in Abbildung ##7 dargestellt wird.

Vor dem ESP32 wurde mit dem Vorgängermodell, dem ESP8266, gearbeitet. Dieser Mikrocontroller verfügt jedoch nur über einen Single-Core-Prozessor und bietet keine Bluetooth-Unterstützung.¹¹ Da zu diesem Zeitpunkt über die Benutzung von Bluetooth nachgedacht wurde, wurde auf den ESP32 umgestellt.

⁹ Website des Herstellers (Espressif), <https://www.espressif.com/en/products/socs/esp32> (Stand 03.03.2024)

¹⁰ Marek Babiuch, Petr Foltyněk, Pavel, Smutnyek, Using the ESP32 Microcontroller for Data Processing, S. 1-4

¹¹ Website des Herstellers (Espressif), <https://www.espressif.com/en/products/socs/esp8266> (Stand 03.03.2024)

Ganz zu Beginn des Projekts wurde mit einem Arduino Uno in Kombination mit einem Ethernet Shield experimentiert. Jedoch stellte sich heraus, dass die Vorteile einer Ethernet-Verbindung für dieses spezifische Projekt nicht erforderlich waren und zudem der Arduino Uno im Vergleich zum ESP32 teurer und leistungsschwächer ist.

4.1.3 Technische Umsetzung des Sensorik-Moduls

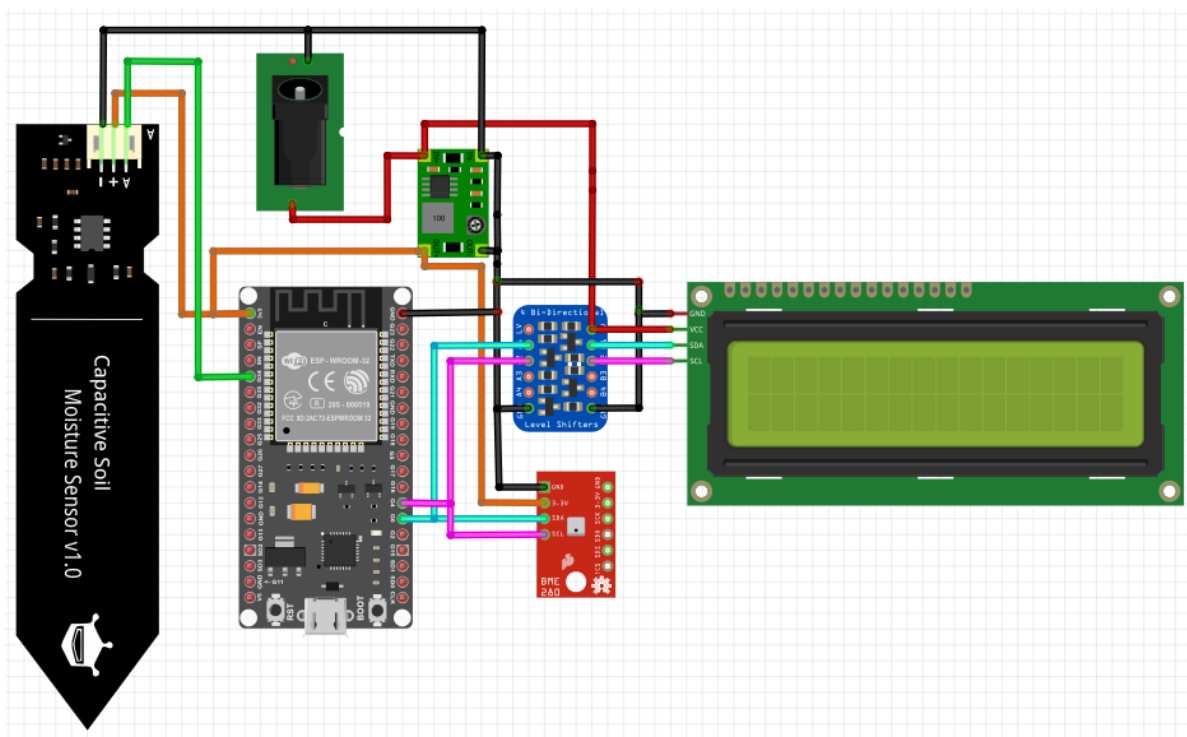


Abbildung 8 - Technologieschema eines Sensorik-Moduls

Das Sensorik Modul besteht aus sechs Bauteilen: einem ESP32, einem 1602 LCD-Display inkl. I2C Controller, einem bidirektionalen 3,3V zu 5V Logik-Konverter, einem DC-DC Spannungsteiler, der Auswertungs elektronik des Feuchtigkeitssensors, ein Bosch BME280 Sensor und eine USB C Buchse. Die Verdrahtung ist in Abbildung ##8 zu sehen.

Die Versorgungsspannung wird über die USB C Buchse zugeführt und liegt für das Sensorik-Moduls bei 5V, diese wird von dem Spannungsteiler auf 3,3V heruntergeregelt. Dadurch können alle Bauteile mit der vorgesehenen Betriebsspannung versorgt werden. Das Display und der I2C-Logik-Konverter benötigen 5V, der ESP und der Sensor benötigen 3,3V. Alle Bauteile haben einen gemeinsamen Masse-Bezugspunkt, um ein gemeinsames Potenzial zu haben.

Der unverarbeitete Dateneingang des Sensors wird von der mitgelieferten Auswertungselektronik auf ein auslesbares analoges Datensignal umgewandelt. Dazu wird ein Bosch BME280 Sensor verbaut, dieser misst die Temperatur, Luftfeuchtigkeit und Luftdruck. Ausgelesen wird er über eine 3,3V I2C Verbindung.

Der ESP kann das Display nicht nativ über I2C ansteuern, da ein ESP ein Logik Level von 3,3V hat und das Display ein Logik Level von 5V benötigt. Deswegen wird ein I2C-Logik-Konverter verbaut, dieser kann, wenn mit 5V Betriebsspannung versorgt, die vom ESP ausgegeben 3,3V I2C-Signale auf 5V übersetzen. Diese I2C Signale werden vom HD44780 Controller empfangen, welcher das Initialisierung des Displays, die Einstellung des Cursors sowie die Anzeige von Text und Zeichen übernimmt. Dadurch ist es möglich, mit nur zwei Datenanschlüssen 16 Zeichen pro Zeile auf zwei Zeilen darzustellen.

4.1.4 Softwaretechnische Umsetzung des Sensorik-Moduls

Im Rahmen der Entwicklung des Sensorik-Moduls habe ich mich für eine Request-and-response-Methodik entschieden, sodass man den Sensorwert bedarfsgerecht abfragen kann. Diese Vorgehensweise beruht darauf, dass der Server bei Bedarf eine Request-Nachricht an den ESP32 sendet, um

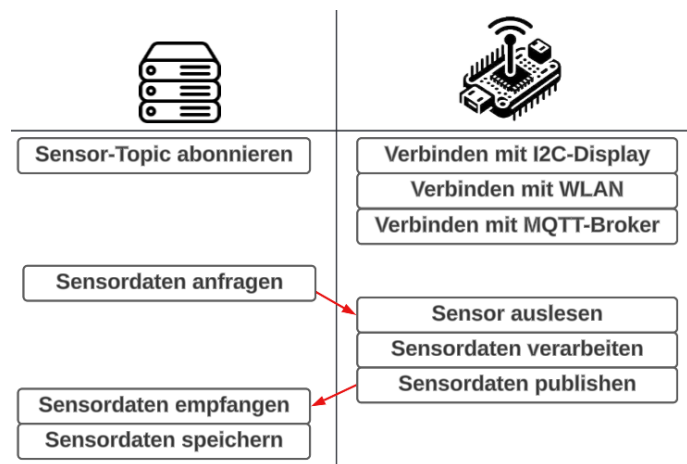


Abbildung 9 - Sensordatenabfrage vereinfacht dargestellt

den aktuellen Sensorwert zu erfragen. Nach Empfang dieser Aufforderung liest der ESP32 den Feuchtigkeitswert analog aus und verarbeitet ihn. Der BME280 kann direkt ausgelesen werden. Alle Sensorwerte werden in einem String zusammengefasst und über das Response-Topic an den Server übermittelt (siehe Abbildung ##9). Diese Methodik wurde eingeführt, da die kontinuierliche Veröffentlichung von Sensorwerten in festen Intervallen als ineffizient identifiziert wurde, besonders wenn keine Notwendigkeit für die Datenaktualisierung besteht.

Für die Implementierung des Codes des Sensorik-Moduls im Arduino C/C++ Dialekt sind vier essenzielle Bibliotheken erforderlich: ``WiFi.h`` für die WLAN-Verbindung, ``PubSubClient.h`` für die MQTT-Kommunikation, ``Wire.h`` für die I2C-Kommunikation und ``LiquidCrystal_I2C.h`` für die Ansteuerung des LCD-Displays. WLAN- und MQTT-Konfigurationsdaten werden in Konstanten gespeichert.

Die Initialisierung beginnt mit der Einrichtung der I2C-Kommunikation, gefolgt von einer Informationsmeldung auf dem Display, die den aktuellen Status des Codes anzeigt. Da der ESP32 mit einer Taktfrequenz von 240 MHz arbeitet, wurden Ein-Sekunden-Verzögerungen nach jedem bedeutenden Schritt integriert, um die Lesbarkeit der Displaymeldungen zu ermöglichen. Anschließend wird versucht, eine WLAN-Verbindung aufzubauen, bis diese erfolgreich etabliert ist. Darauffolgend wird die Verbindung zum MQTT-Broker hergestellt und das Request-Topic abonniert.

Der Feuchtigkeitswert wird analog ausgelesen und dann über das Mapping verarbeitet, was dafür sorgt, dass der Sensorwert in einem Prozentwert zwischen 0% (komplett ausgetrocknet) und 100% (steht in Wasser) liegt. Dafür werden, durch Testungen herausgefunden, Grenzwerte angegeben. Der Grenzwert für die Trockenheit liegt bei 4095

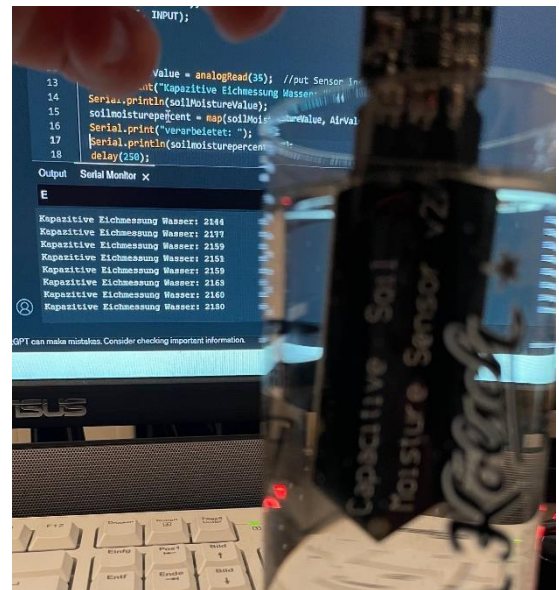


Abbildung 10 - Eichmessung des kapazitiven Feuchtigkeits-Sensor in Wasser

und entsteht dabei, wenn der Sensor in die Luft gehalten wird. Der Grenzwert für die Nässe liegt bei ungefähr 1000, was der Mittelwert von Messungen ist, wobei der Sensor in reines Wasser gehalten wird. Die Map-Funktion ist nativ in der Arduino Sprache, somit lässt sich die Verarbeitung durch folgende Zeile Code umsetzen:

```
Feuchtigkeitsprozentwert = map(gemesseneFeuchtigkeit, 4095, 1000, 0, 100);
```

Im kontinuierlichen Betrieb überprüft der ESP32, ob eine aktive WLAN-Verbindung besteht; bei Unterbrechungen wird der Verbindungsaufbau automatisch wiederholt. Ebenso wird kontinuierlich auf eingehende MQTT-Nachrichten geachtet. Der Sensorwert wird auf dem Display dargestellt und zur Verbesserung der Lesbarkeit nur alle 1,5 Sekunden aktualisiert, um ein Flackern zu vermeiden.

Ein Video, was den ersten Aufbau des Sensorik-Moduls zeigt, kann unter <https://bit.ly/wv-sensorik> angesehen werden. Die komplette Firmware kann unter <https://bit.ly/wv-sensor> eingesehen werden.

4.1.5 Technische Umsetzung des Aktorik-Moduls

Das Aktorik Modul besteht aus acht Bauteilen: einem ESP32, einem Relais, ein Magnetventil, einem 1602 LCD-Display inkl. I2C Controller, einem I2C-Logik-Konverter, zwei DC-DC Spannungsteiler und eine USB C-Buchse. Die

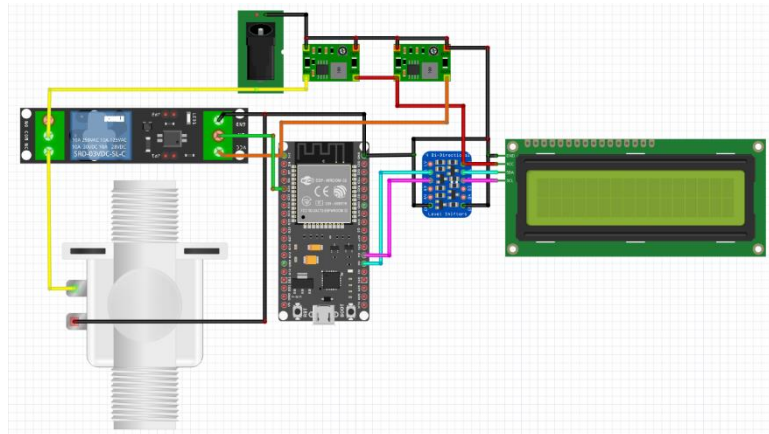


Abbildung 11 - Technologieschema eines Aktorik-Moduls

Verkabelung ist in Abbildung #10 zu erkennen.

Die Versorgungsspannung wird über die USB C-Buchse zugeführt und liegt bei 12V DC. Weil die Bauteile verschiedene Betriebsspannungen benötigen, wird die Versorgungsspannung von zwei Spannungsteiler erst auf 5V heruntergeregelt und die 5V nochmal auf 3,3V heruntergeregelt. Das Magnetventil wird mit 12V betrieben. Das Display und der I2C-Logik-Konverter benötigen 5V, der ESP benötigt 3,3V. Alle Bauteile haben einen gemeinsamen Masse-Bezugspunkt, um ein gemeinsames Potenzial zu haben.

Dem ESP32 ist es technisch nicht möglich, das Magnetventil direkt anzusteuern, deswegen wird mit einem H1-51 v1.0 Relais gearbeitet. Der ESP32 gibt ein 3,3V-Signal aus, wenn er dazu aufgefordert wird. Mit dem 3,3V Signal wird das Relais angesteuert, welches die 12V schaltet, wodurch sich das Magnetventil öffnet. Sobald die Öffnungsdauer abgelaufen ist, gibt der

der ESP32 wieder 0V aus, wodurch das Relais wieder schließt und ebenfalls das Magnetventil.

Der Anschluss des LCD-Displays ist gleich wie beim Sensorik-Modul und wurde im Kapitel ## erklärt.

4.1.6 Softwaretechnische Umsetzung des Aktorik-Moduls

Die Firmware des Aktorik-Moduls teilt wesentliche Grundstrukturen mit dem des Sensorik-Moduls, weshalb sich die nachfolgende Erörterung primär auf die spezifischen Anpassungen konzentriert, die für das Aktorik-Modul vorgenommen wurden.

Anfangs werden die MQTT-Topics abonniert, welche zur Steuerung des Aktorik-Moduls notwendig

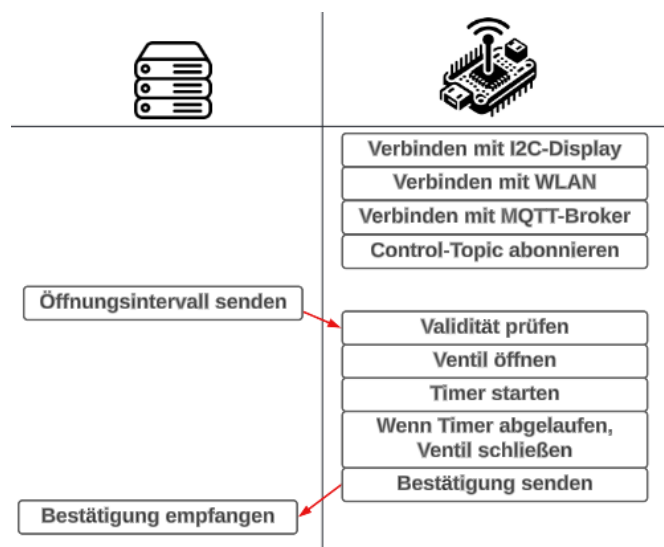


Abbildung 12 - Ventilsteuerungslogik vereinfacht dargestellt

sind. Der Server übermittelt die Dauer des Bewässerungsintervalls in Sekunden. Beim Eingang einer solchen Nachricht wird das Magnetventil geöffnet, woraufhin eine Verzögerung, in Form einer Timer-Funktion, eingeleitet wird, die der übertragenen Intervalldauer entspricht. Nach deren Ablauf schließt das Ventil und es wird eine Rückmeldung an den Server gesendet, um den erfolgreichen Abschluss des Bewässerungsvorgangs zu bestätigen.

Um die Betriebssicherheit zu garantieren, wurden drei essenzielle Kontrollmechanismen in die Firmware integriert. Erstens erfolgt eine Überprüfung der vom Server gesendeten Intervalldauer auf den richtigen Datentyp, um Fehlfunktionen durch inkorrekte Daten zu vermeiden. Dazu wird überprüft, ob der Bewässerungszeitraum zwischen 0 Sekunden und einer Stunde liegt.

Dazu besteht die Möglichkeit, dass der Server jederzeit eine Notfall-Schließung des Ventils anordnen kann, die umgehend umgesetzt wird, um potenzielle Schäden abzuwenden. Diese Logik ist in Abbildung ##11 zu erkennen. Ein Video, was den ersten Aufbau der Relais-Fernschaltung zeigt, kann unter <https://bit.ly/wv-aktorik> angesehen werden.

4.2 Sensorik und Aktorik

4.2.1 Magnetventile

Magnetventile sind die ausführenden Komponenten des Systems, wobei jedes eine Bewässerungslinie, also eine Gruppe von Schläuchen, Sprinklern und anderen Bewässerungskomponenten, die an einem Ventil angeschlossen sind, definiert. Da diese Komponenten nur gemeinsam ein- oder ausgeschaltet werden können, ist das Ventil entscheidend für die Kontrolle des Wasserflusses in einer bestimmten Zone. Die Entscheidung darüber, wie viele Pflanzen von einer Bewässerungslinie versorgt werden, liegt beim Benutzer.

Die Installation der Magnetventile ist so konzipiert, dass sie zwischen bereits existierenden Leitungen eingebaut werden können. Für die private Implementierung von Wateringvision wurde ein Magnetventil mit einer Betriebsspannung von DC 12V ausgewählt. Diese Ventile sind in der Lage, einen Wasserdruck von 0,2 Bar bis zu 8 Bar zu schalten, was sie für eine Vielzahl von Bewässerungsszenarien geeignet macht.

Ein entscheidender Sicherheitsaspekt bei der Auswahl war die Entscheidung für ein Normally Closed (NC) Ventil. Dies bedeutet, dass die Ventile im stromlosen Zustand geschlossen sind. Sollte also die Spannungsversorgung ausfallen, würde nicht unkontrolliert Wasser fließen. ¹²

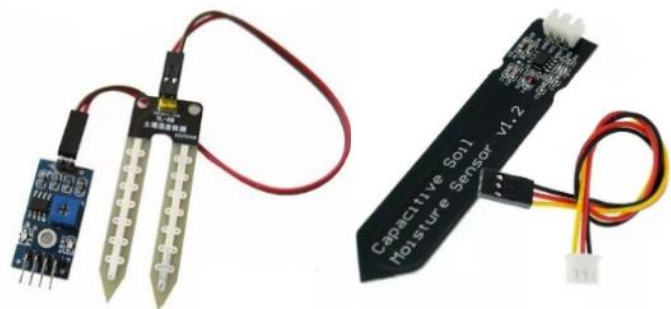
¹² Technische Informatik Q3 bei Herr Betz

Bezüglich der Anschlüsse besitzen die Ventile standardmäßig einen G3/4" Ein- und Ausgang. In meiner Installation wird mit klassischen Gardena-Systemanschlüssen gearbeitet. Dabei werden Adapter für die gewünschten Schlauchanschlüsse verwendet. Diese Adapter können jedoch nach Bedarf und individuellen Präferenzen angepasst werden, was das System flexibel und anpassungsfähig macht.

Das verwendete Ventil besteht aus Messing und ist als 2/2-Wege-Ventil konzipiert. Es ist zu beachten, dass auch andere Magnetventile verwendet werden können, beispielsweise wenn ein höherer Wasserdruck geschaltet werden soll. Die Wahl des Ventils hängt somit von den spezifischen Anforderungen der Bewässerungslinie und den Vorlieben des Benutzers ab.¹³

4.2.2 Bodenfeuchtesensor

Derzeit sind zwei Typen Feuchtesensoren im Einsatz, ein Leitfähigkeits- und Kapazitätssensor (siehe Abbildung ##12).



Leitfähigkeitssensor

Kapazitätssensor

Leitfähigkeitssensoren messen den elektrischen Wider-

Abbildung 13 - YL-69 Leitfähigkeitssensor (links) & kapazitiver Sensor v1.2 (rechts)

stand des Bodens, der sich mit der Feuchtigkeit ändert: je feuchter der Boden, desto geringer der Widerstand und umso höher die Leitfähigkeit. Derzeit wird ein YL-69 Sensor benutzt, bei dem die Mess- und Auswertungsteile durch ein Kabel verbunden werden. Ausgelesen wird der Sensor über ein analoges Datensignal.

¹³ eBay Listing des Magnetventils; <https://www.ebay.de/itm/384318597260>; (Stand 05.03.2024); (Es konnte kein reines Datenblatt gefunden werden)

Kapazitive Sensoren messen die Änderung der Kapazität zwischen zwei Elektroden, wenn sich die Feuchtigkeit des Bodens ändert: je feuchter der Boden, desto höher die Kapazität. Im Einsatz ist ein Capacitive Soil Moisture Sensor V1.2, welcher die Änderung der Kapazität in ein analoges Datensignal umgewandelt, dessen Spannung proportional zur Bodenfeuchtigkeit ist: je höher die Spannung, desto feuchter der Boden.

Die Anzahl und Verteilung der Sensoren beeinflussen die Genauigkeit des Bewässerungsprogramms. Mehr Sensoren ermöglichen eine präzisere Erfassung der Bodenfeuchtigkeit in verschiedenen Bereichen des Gartens. Allerdings ist es nicht notwendig, in einem Beet bei jeder Pflanze einen Sensor zu installieren, da benachbarte Sensoren oft ähnliche Werte erfassen. Die Entscheidung über Anzahl und Platzierung wird dem Benutzer überlassen.

4.2.3 Füllstandssensor für eine Zisterne

Die Kenntnis des Füllstands der Zisterne ist entscheidend, weil es wichtig ist, dass die Zisterne nicht leerläuft, um die kontinuierliche Funktionsfähigkeit der angeschlossenen Toilettenspülung zu gewährleisten.

Ich habe mich für einen Ultraschall-Sensor entschieden, da im Informationstechnologie-Unterricht verschiedene Sensortypen zur Abstandsmessung besprochen worden sind. Ein Ultraschallsensor überzeugt vor allem durch seine Reichweite und die Fähigkeit, Flüssigkeiten zu detektieren.¹⁴

Es kommt ein JSN-SR04T Sensor zum Einsatz, da die Messsonde und die Auswertungselektronik separat verbaut werden können. Dies ist vorteilhaft bei schlecht zugänglichen Anwendungsorten, wie bei einer Zisterne. Eben-

¹⁴ Informationstechnologie Q3 bei Herr Geißler

falls ist nur die Messsonde als wasserdicht angegeben. Die Maximalreichweite von 4,5 Metern ist ausreichend für unsere Zisterne.^{15 16} Ein Video, was die erste erfolgreiche Inbetriebnahme des Ultraschall-Sensor zeigt, ist unter <https://bit.ly/wv-ultra> anzusehen.

Die Firmware des Sensorik-Moduls für den Füllstandssensor für eine Zisterne wurde in der Datenverarbeitung verändert. Der Sensor gibt den gemessenen Abstand zwischen sich und der nächsten Oberfläche in cm zurück. Dieser Wert wird verwendet, um den prozentualen Füllstand und die Literanzahl zu berechnen. Für den prozentualen Füllstand wird wieder die Mapping-Methode verwendet, welchen in ## erläutert wurde. Dabei wird die Tiefe zwischen Sensor und Zisternenboden und der gemessene Abstand übergeben, wodurch der prozentuale Füllstand berechnet werden kann.

Die Formel für die Berechnung des Füllstandes in Litern basiert auf dem Prinzip der linearen Proportionalität bei konstanter Querschnittsfläche, was bedeutet, dass das Volumen der Flüssigkeit direkt proportional zur Höhe der Wassersäule in einem Behälter mit gleichbleibendem Querschnitt ist. Die Berechnung lässt sich auf folgende Zeile Code herunterbrechen:

```
currentLiters = ((maxDepth - distanceToWater)) * maxVolume) / maxDepth;
```

Dabei ist currentLiters die berechnete Literanzahl in der Zisterne, maxDepth die Tiefe der Zisterne als Konstante, distanceToWater der gemessene Abstand zwischen Sensor und Wasseroberfläche und maxVolume die maximale Literanzahl als Konstante.

¹⁵Benne de Bakker; Waterproof JSN-SR04T Ultrasonic Distance Sensor with Arduino Tutorial; 28.03.2021 ; https://saliterman.umn.edu/sites/saliterman.umn.edu/files/files/general/arduino_waterproof_jsn-sr04t_ultrasonic_sensor_2_examples.pdf; (Stand 05.03.2024)

¹⁶Wolfgang Ewald; HC-SR04 und JSN-SR04T-2.0 Abstandssensoren; 31.08.2019; <https://wolles-elektronikkiste.de/hc-sr04-und-jsn-sr04t-2-0-abstandssensoren>; (Stand 05.03.2024)

Die Firmware kann unter <https://bit.ly/wv-zisterne> und die Konfiguration unter <https://bit.ly/wv-zisterne-config> aufgerufen werden.

4.3 Informationstechnische Umsetzung

4.3.1 Betriebssystem und Server Hardware

In der Entwicklungsphase wurde zunächst mit einem Raspberry Pi gearbeitet, welcher ein kleiner, kostengünstiger Einplatinencomputer ist, der für eine Vielzahl von Anwendungen eingesetzt werden kann.¹⁷ Auf diesem lief das Betriebssystem Raspberry Pi OS, welches auf Debian basiert und somit ein Unix-ähnliches Betriebssystem darstellt, das auf dem Linux-Kernel basiert.¹⁸ Dieser Aufbau wird in Abbildung ##5 dargestellt. Während der Entwicklungsphase lag der Fokus

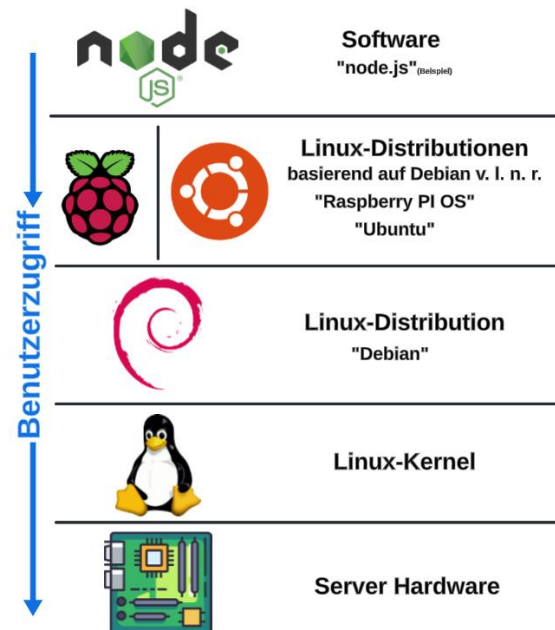


Abbildung 14 - Benutzerzugriff in einem Linux System von der Software bis zur Hardware

nicht auf dem dauerhaften Betrieb (Deployment) der Software, wodurch der Raspberry Pi insbesondere wegen seiner Platzierung im gleichen Netzwerk wie die Mikrocontroller und der vereinfachten Entwicklung ohne sofortige Notwendigkeit der Verschlüsselung oder umfangreichen Netzwerksicherheitsmaßnahmen bevorzugt wurde.

¹⁷ Jolle W. Jolles, Broad-scale applications of the Raspberry Pi: A review and guide for biologists, British Ecological Society, S. 2-9

¹⁸ Emilio Raggi, Keir Thomas, Sandar van Vugt, Beginning of Linux Sixth Edition, appres, New York, USA, 2011, S.13-22

Mit der Einführung von React und dem steigenden Bedarf an Leistungsfähigkeit wurde auf einen Virtual Private Server (VPS) gewechselt. Der gewählte VPS verfügt über 4 CPU-Kerne, 8 GB RAM und 240 GB NVMe SSD-Speicherplatz, ist in Deutschland gehostet und kostet 14 € im Monat.¹⁹ Als Betriebssystem für den VPS wurde Ubuntu 22 Server gewählt.

Die Entscheidung für IONOS als Hosting-Anbieter fiel aus Kostengründen. Obwohl auch größere Anbieter wie AWS, Azure oder Google Cloud in Betracht gezogen wurden, erwiesen sich deren Hosting-Gebühren als deutlich teurer. Beispielsweise würde eine EC2-Instance bei AWS mit vergleichbaren Spezifikationen und Hosting in Deutschland Kosten von ungefähr 80 bis 120 Euro pro Monat verursachen.²⁰ Die Preise anderer großer internationaler Hosting-Anbieter sind ähnlich hoch. Der gewählte VPS ist eine Cloud-Dienstleistung, die jedoch im Vergleich zu Diensten größerer Anbieter nicht so leicht skaliert werden kann, was für das Projekt jedoch eine akzeptable Einschränkung darstellte.

4.3.2 Kommunikationsprotokolle und Entwicklungswerkzeuge

Die gesamte Kommunikation zwischen dem Administrator und dem Server, insbesondere bei der Verwendung von Bash, wird über SSH (Secure Shell) verschlüsselt. SSH ist ein Protokoll, das eine sichere Methode zur Fernanmeldung auf einem Server bietet. Es verschlüsselt die gesamte Sitzung, wodurch die Sicherheit von Benutzerdaten gewährleistet ist.^{21 22}

¹⁹ Preisliste IONOS VPS, <https://www.ionos.de/server/vps> (Stand 03.03.2024)

²⁰ Kostenkalkulator AWS, <https://calculator.aws/#/createCalculator/ec2-enhancement> bei Region FFM, 4 CPU-Kerne und 8GB RAM (Stand 03.03.2024),

²¹ Daniel J. Barrett, Ph.D., Richard Silverman, SSH, The Secure Shell: The Definitive Guide, O'Reilly, Sebastopol, CA, USA, S. 1-14

²² Cameron Newham, Learning the bash Shell, 3rd Edition, O'Reilly, Sebastopol, CA, USA, S. 23-42

Anfangs wurde Putty, ein SSH-Client, für die Verbindung zum Server verwendet. Jedoch bietet es, mit dem Nano-Editor, nur eingeschränkte Funktionen zur Codebearbeitung.²³ Deswegen erfolgte ein Wechsel zu Visual Studio Code (VSCode) mit der SSH-Erweiterung. Diese Entwicklungsumgebung ermöglicht es Benutzern, Code direkt auf dem Server zu bearbeiten und zu testen, ohne dass eine ständige Übertragung von Dateien zwischen lokalem System und Server notwendig ist. Die SSH-Erweiterung für VSCode vereinfacht den Entwicklungsprozess erheblich, indem sie einen nahtlosen Zugriff auf entfernte Dateien über eine sichere Verbindung bietet und die Möglichkeit gibt, diese Dateien so zu bearbeiten, als wären sie lokal verfügbar.²⁴

Für die Dateiübertragung wird FileZilla verwendet, zur sicheren Kommunikation kommt SFTP (Secure File Transfer Protocol) zum Einsatz, welches die Übertragung über verschlüsselte Verbindungen gewährleistet.²⁵

4.3.3 JavaScript als Fullstack-Lösung

Bei Entwicklungsbeginn wurde eine moderne und flexible Programmiersprache gesucht, die die gesamte Bandbreite von Entwicklungsaufgaben abdecken kann. Die Entscheidung fiel auf JavaScript, eine Sprache, die sich durch ihre Vielseitigkeit und breite Unterstützung auszeichnet.²⁶

Ein wesentlicher Vorteil von JavaScript ist die Möglichkeit, es als Fullstack-Sprache zu verwenden. JavaScript ermöglicht diese durchgehende Entwicklung durch Node.js, eine Open-Source-JavaScript-Laufzeitumgebung, welche es ermöglicht JavaScript auch außerhalb eines Web-Browsers auszuführen²⁷.

²³ Putty Homepage, <https://www.putty.org/>, Stand (03.03.2024)

²⁴ VSCode SSH Extension Dokumentation, <https://code.visualstudio.com/docs/remote/ssh>, (Stand 03.03.2024)

²⁵ Filezilla Homepage, <https://filezilla-project.org/>, (Stand 03.03.2024)

²⁶ Ali Mesbah, Understanding Asynchronous Interactions in Full-Stack JavaScript, University of British Columbia, Vancouver, BC, Canada, o.J., S.1

²⁷ Azat Mardam, Full Sack JavaScript, apress, 2015, S.137-154

Für JavaScript gibt es dank seiner Beliebtheit zahlreicher hilfreicher Module und Bibliotheken. Diese sind über npm (Node Package Manager) leicht zugänglich und installierbar.^{28 29}

4.3.4 Aufbau des Backend

Das Backend besteht aus einer Vielzahl verschiedener Skripte, deren Zusammenspiel das Gesamtsystem bildet. Die Kommunikation zwischen diesen Teilsystemen erfolgt entweder über gemeinsamen Zugriff auf Datenbanken oder das Abspeichern von Daten in JSON-Objekten. Diese können entweder über eine API übermittelt oder in statischen Dateien abgespeichert und ausgelesen werden. Die Kommunikation zwischen Frontend und Backend wird in Kapitel ## erklärt.

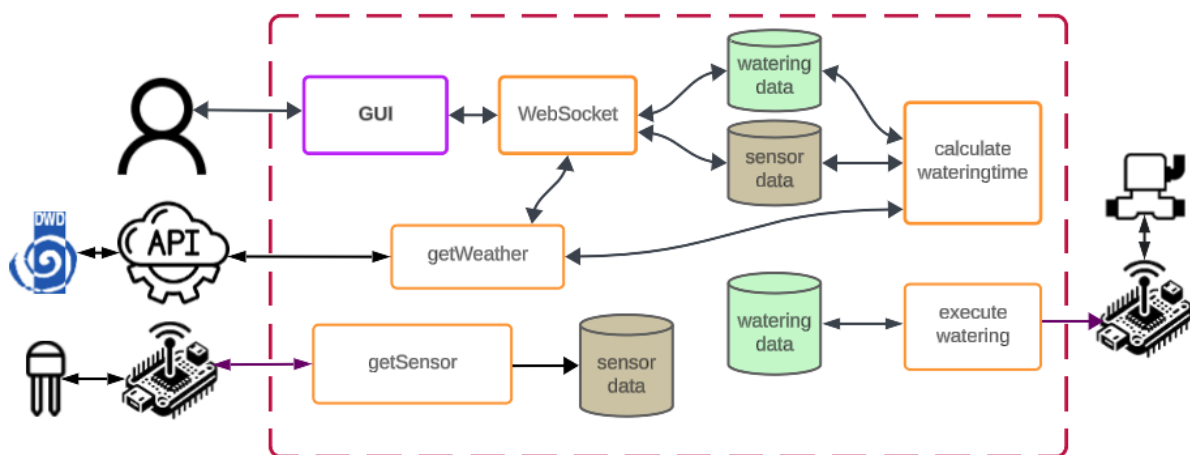


Abbildung 15 - Vereinfachte Flowchart ähnliche Darstellung des Backend

Die Unterteilung in verschiedene Teilsysteme bietet mehrere Vorteile. Erstens ermöglicht es eine einfachere Fokussierung auf spezifische Funktionen beim Arbeiten an einzelnen Teilen des Systems. Eine vereinfachte Darstellung des Systems ist in Abbildung ##6 zu erkennen. Zweitens können Bugs und Probleme schneller und effizienter identifiziert und behoben werden, da

²⁸ Julzer Ali, Node Package Manager, Packet Publishing, Birmingham, 2013, S.18-21

²⁹ npm About Website, <https://www.npmjs.com/about> (Stand 05.03.2024)

die Fehlerquelle auf ein bestimmtes Teilsystem begrenzt ist. Drittens ermöglicht diese Modularität eine leichtere Skalierbarkeit und Anpassung des Systems. Viertens führt eine klare Trennung der Funktionen zu einer verbesserten Wartbarkeit und Übersichtlichkeit des Gesamtsystems.

Die dauerhafte Ausführung von Skripten wird von pm2 gewährleistet.³⁰ Die zeitliche Steuerung der Ausführung der einzelnen Skripte wird durch den cron-Dämon realisiert. Dieser Dienst ermöglicht es, Skripte zu festgelegten Zeiten automatisch ausführen zu lassen.³¹ Der Prozess der Datenverarbeitung im System ist reihenweise organisiert. Angefangen bei den Eingangsdaten, wie Sensorwerten und Wetterinformationen, werden die Daten schrittweise verarbeitet. Jeder Schritt baut auf den vorherigen auf und führt zu einer kontinuierlichen Verarbeitung der Daten.

4.3.5 Arduino C/C++ Dialekt

Für die Programmierung der ESP32-Mikrocontroller wurde der C/C++ Arduino-Dialekt gewählt. Diese Entscheidung basierte auf meiner bereits vorhandenen schulischen sowie privaten Erfahrung mit dieser Programmiersprache. Der C/C++ Arduino-Dialekt zeichnet sich durch seine Einfachheit aus, da viele komplexe und traditionelle IT-Konzepte ausgegliedert oder vereinfacht wurden.³² Vorteilhaft ist ebenfalls die große Auswahl an Libraries, welche die Entwicklung von Projekten erheblich beschleunigen und vereinfachen. Diese Ressourcen ermöglichen es, komplexe Funktionen und Verbindungen mit wenigen Zeilen Code zu implementieren.³³

³⁰ Website von pm2, <https://pm2.keymetrics.io/>, (Stand 05.03.2024)

³¹ Basher Modallal, Cron Jobs Explained With Nodejs, <https://medium.com/@basher.modallal/cron-jobs-explained-with-nodejs-79099001a6f1>, (Stand 09.03.2024)

³² Sanmit Chartterjee, What are the Key Pros and Cons of the Arduino Programming Language?, 2023, <https://emeritus.org/blog/coding-arduino-programming-language/> (Stand 03.03.2024)

³³ Offizielle Arduino Libraries GitHub Repository, <https://github.com/arduino-libraries> (Stand 03.03.2024)

Anfangs wurde für die Entwicklung die Arduino IDE verwendet, da sie sowohl in privaten Projekten als auch in der Schule zum Einsatz kam. Die Arduino IDE bietet eine simple benutzerfreundliche Oberfläche und ist speziell auf die Programmierung von Arduino Mikrocontrollern zugeschnitten.

4.4 Netzwerktechnische Umsetzung

4.4.1 Domainverwaltung und Netzwerkinfrastruktur

Zunächst wurde die Domain „wateringvision.de“ erworben. Diese Domain ist mit einem A-DNS-Eintrag versehen, der auf die IP-Adresse des zugehörigen Servers verweist. Die Flexibilität dieses Eintrags ermöglicht es, die Verweisadresse bei Bedarf zu aktualisieren, sollte sich die IP des Servers ändern. Dieser Me-

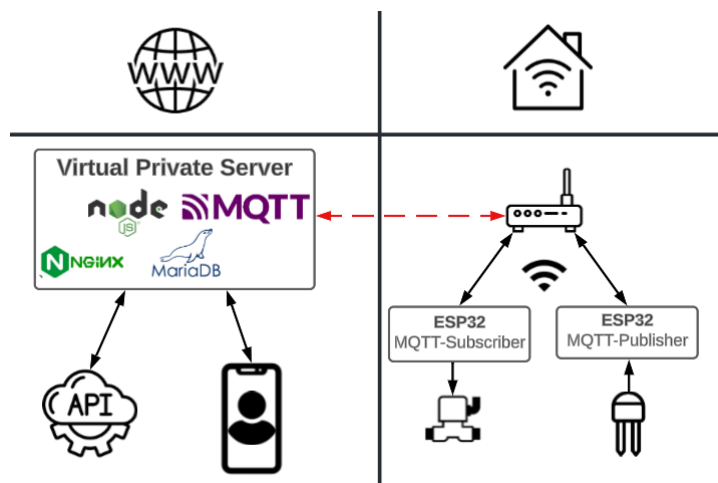


Abbildung 16 - Netzwerkzugehörigkeit der Komponenten, WWW (links), LAN (rechts)

chanismus bietet den signifikanten Vorteil, dass im Quellcode der ESP32-Mikrocontroller die Domain fest als Adresse für den MQTT-Broker integriert werden kann. Dadurch ist gewährleistet, dass die Clients stets eine Verbindung zum Server aufbauen können, unabhängig von etwaigen Änderungen der Server-IP.

Die Netzwerkzugehörigkeit der Komponenten ist in Abbildung ## 13 dargestellt. Zu beachten ist, dass der User auch aus dem Heimnetzwerk auf den VPS zugreifen kann.

Zum aktuellen Zeitpunkt erfordert die Kommunikation aller Module mit dem Server eine WLAN-Verbindung. Dies stellt eine grundlegende Vorausset-

zung für den Informationsaustausch und die Steuerung des Bewässerungssystems dar. Blickt man in die Zukunft, ist vorgesehen, dass nicht mehr alle Module zwingend eine WLAN-Verbindung benötigen.

Die Abdeckung eines ausgedehnten Areals mit einem stabilen WLAN-Signal stellt eine Herausforderung dar. Ein einzelner Router ist oftmals nicht ausreichend, um eine flächendeckende und zuverlässige Netzwerkverbindung zu garantieren. Aus diesem Grund wurde ein WLAN-Mesh-Netzwerk implementiert. Innerhalb dieses Netzwerks senden alle Router dasselbe WLAN-Signal aus und stehen in ständiger Kommunikation miteinander. Dies ermöglicht es verbundenen Geräten, nahtlos zwischen verschiedenen Routern zu wechseln, ohne die Netzwerkverbindung zu verlieren. Für Router, die keinen direkten Zugang zu einem Ethernet-Anschluss haben, bietet die Einbindung über Powerline Communication (PLC) eine Lösung. Hierbei wird das bestehende Stromnetz genutzt, um eine Netzwerkverbindung herzustellen.

In den Anfängen des Projekts wurde mit einem Arduino-Ethernet-Shield experimentiert. Es stellte sich jedoch heraus, dass die spezifischen Vorteile einer Ethernet-Verbindung von untergeordneter Bedeutung sind. Angesichts der Notwendigkeit, geringe Datenmengen zu übertragen und eine hohe Flexibilität in der Platzierung der Geräte zu gewährleisten, hat sich WLAN als die effizienteste Lösung erwiesen, um eine umfassende Netzwerkabdeckung auf großen Flächen zu realisieren. Das älteste Video zeigt die erste funktionierende Kommunikation, von einem PC zu einem Mikrocontroller über HTTP-API, und ist über <https://bit.ly/wv-ethernet> aufrufbar.

4.4.2 MQTT-Protokoll

MQTT (Message Queuing Telemetry Transport) ist ein leichtgewichtiges Nachrichtenprotokoll, das effiziente, zuverlässige und minimale Datentransfers ermöglicht, was es ideal für die Kommunikation zwischen Server und

Modulen in diesem IoT-System macht³⁴. MQTT operiert nach dem Prinzip des Publish-Subscribe-Modells, welches in Abbildung 16 visualisiert wird: Ein Publisher sendet Nachrichten (Payloads), indem er diese auf einem definier-

MQTT Publish / Subscribe

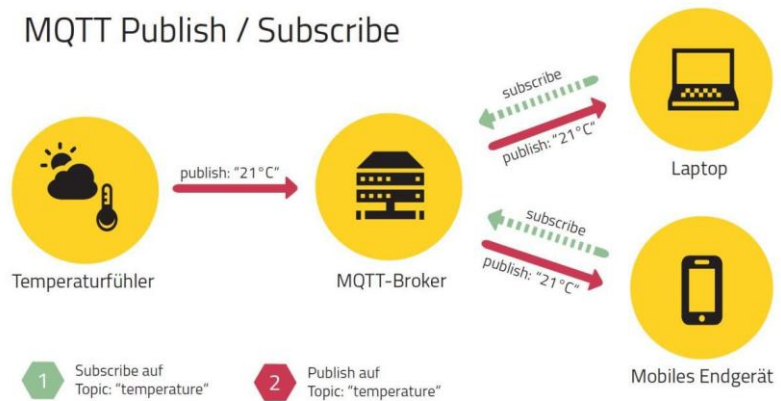


Abbildung 17 - Publish-Subscribe-Model dargestellt, Sensor veröffentlicht Temperaturwert Endgeräte empfangen diesen

ten Kanal (Topic) veröffentlicht. Diese Nachrichten werden an einen zentralen Broker gesendet. Sobald ein Subscriber ein bestimmtes Topic abonniert hat, erhält er die entsprechenden Nachrichten unmittelbar zugestellt. Clients, die das Topic nicht abonniert haben oder gerade nicht aktiv zuhören, erhalten diese Nachrichten nicht³⁵. Ein Video, was die erste erfolgreiche Verbindung von zwei ESP via MQTT zeigt, ist unter <https://bit.ly/wv-mqtt> anzusehen.

Ein wesentliches Merkmal von MQTT ist, dass Publisher und Subscriber anonym zueinander sind; sie müssen sich nicht gegenseitig kennen, um zu kommunizieren. Jeder Client benötigt lediglich eine einzige Verbindung zum Broker, was den Netzwerkverkehr und die Komplexität des Systems reduziert³⁶. Es ist theoretisch möglich, dass der Nachrichtenverkehr gespeichert wird und sich neue Subscriber den letztbekannten Payload erhalten, dies wurde nicht aktiviert, weil kein sinnvoller Anwendungsfall gefunden wurde. Die Quality of Service (QoS) in MQTT bestimmt die Garantieebene für die

³⁴ Lukas Schulz, "Datenübertragung mittels MQTT", Bachelorarbeit, Universität Ulm, 2017, S. 5-7

³⁵ Ebd., S. 7-9

³⁶ Ebd., S. 10

Zustellung einer Nachricht, von "höchstens einmal" (QoS 0) über "mindestens einmal" (QoS 1) bis hin zu "genau einmal" (QoS 2). Eigentlich sollte QoS 2 implementiert werden, die Arduino-Library unterstützt dies aber nicht.

Als Broker wird Aedes genutzt, welches ein schlankes, selbstkonfigurierbares und Open-Source MQTT-Broker-Projekt für Node.js ist. Es zeichnet sich durch seine "Barebone"-Architektur aus, wodurch es flexibel an eigene Anforderungen angepasst werden kann. Anfangs wurde versucht, Mosquitto als Broker zu implementieren. Da dieser nach Stunden nicht funktioniert hat, fiel die Entscheidung für Aedes, welcher beim ersten Versuch erfolgreich implementiert wurde.³⁷ Der Code des Brokers ist unter <https://bit.ly/wv-broker> aufrufbar.

Topics sind wie ein Dateisystem strukturiert, mit Ebenen, die durch Schrägstriche getrennt sind, das Response-Topic ist beispielsweise wie folgt aufgebaut: `/responseSensor/berlin`. Beim Veröffentlichen bedeutet das Weglassen weiterer Ebenen im Topic, dass eine Nachricht an alle Abonnenten des angegebenen Basis-Topics gesendet wird, unabhängig von deren spezifischen Unterthemen. Beim Empfangen können Ebenen durch den Rauten-Wildcard ersetzt werden, um alle Nachrichten zu empfangen, die mit dem vorangestellten Topic-Pfad beginnen, unabhängig von deren weiteren Unterthemen. Bei den implementierten Topics wird der Operator an die erste Stelle gestellt, sodass man gleichzeitig alle Clients erreichen kann. Beispielsweise abonniert das Skript, welches die Sensordaten in die Datenbank abspeichert, das Topic `/responseSensor/#` und published auf dem Topic `/requestSensor/`. Dadurch empfängt es die Sensordaten aller Module. Die Sensordaten werden der Datenbank-Tabelle durch den Modulnamen im Topic zugeordnet.

³⁷ GitHub Repository von aedes, <https://github.com/moscajs/aedes>, (Stand 08.03.2024)

MQTT, ursprünglich von IBM-Mitarbeitern entwickelt, ist in der Anwendungsschicht der TCP/IP-Protokollfamilie verortet, was seine breite Kompatibilität für verschiedene Netzwerke betont. Als OASIS-Standard erlangte es vor allem in IoT-Anwendungen Industrieakzeptanz, wo Effizienz bei Datentransfer entscheidend ist³⁸.

4.4.3 Netzwerksicherheit

Netzwerksicherheit ist ein wichtiger Aspekt, besonders wenn Server und Clients über das World Wide Web erreichbar sind. Es ist essenziell, sich auf diverse Bedrohungen vorzubereiten und die Sicherheitsmaßnahmen kontinuierlich zu aktualisieren, auch wenn absolute Sicherheit nie garantiert werden kann. Grundlegende Sicherheitsmaßnahmen sind implementiert, mit der Absicht, diese im Laufe der Zeit zu verbessern.

Die Kommunikation zwischen MQTT-Broker und Client erfolgt standardmäßig TLS-verschlüsselt, ergänzt durch eine Authentifizierung mittels Benutzername und Passwort, wobei die Passwörter hohe Sicherheitsstandards erfüllen. Die Datenübertragung der Website findet über das verschlüsselte Protokoll HTTPS statt. Die SSL-Verbindung zwischen dem Serveradmin und dem Server kann nur über den richtigen Private Key aufgebaut werden, der nur auf meinem Rechner abgespeichert ist.

Um potenzielle Schwachstellen zu minimieren, werden nur die notwendigen Server Ports geöffnet. Da DDoS-Attacken weiterhin eine Bedrohung darstellen, könnten bei Bedarf Netzwerksicherheitsdienste von Anbietern wie Cloudflare hinzugezogen werden. Zukünftige Überlegungen beinhalten die Nutzung einer IP-Whitelist, um die Serverkommunikation auf autorisierte

³⁸ Lukas Schulz, "Datenübertragung mittels MQTT", Bachelorarbeit, Universität Ulm, 2017, S. 5

Netzwerke zu beschränken, wobei mehr Sicherheit mit geringerem Komfort einhergeht.³⁹

4.5 Benutzeroberfläche

4.5.1 Konzept

Die Benutzeroberfläche stellt die essenzielle Schnittstelle zwischen dem Benutzer und dem gesamten Bewässerungssystem dar. Sie fungiert als umfassende Steuerzentrale, über die sämtliche Konfigurationen und Einstellungen vorgenommen werden können. Diese universelle Plattform bietet eine vollständige Übersicht aller verbundenen Module, ermöglicht die Überwachung des Server-Verbindungsstatus sowie den Abruf aktueller Sensorwerte und den Status der Ventile. Zudem können Benutzer sofortige Aktionen, wie beispielsweise einen Notstopp des Magnetventils, direkt über die Oberfläche auslösen.

Ein wesentlicher Bestandteil der Benutzeroberfläche ist die Möglichkeit, Bewässerungslinien zu erstellen und zu bearbeiten. Hierbei können Sensorik und Aktorik konfiguriert und miteinander verknüpft werden, um individuelle Bewässerungspläne zu erstellen. Benutzer können präferierte Kalibrierungswerte festlegen, die definieren, in welchen Intervallen die Bewässerung unter gewöhnlichen Bedingungen stattfinden soll, inklusive der Festlegung spezifischer Tage und Zeitfenster für die Bewässerung. Des Weiteren besteht die Möglichkeit, Bewässerungszeiträume anzugeben und zu bestimmen, ob Pflanzen überdacht sind und somit Wetterdaten in der Bewässerungssteuerung außer Acht gelassen werden sollen. Eine alternative Einstellung ermöglicht es, das System wie eine klassische Zeitschaltuhr zu nutzen, sodass Ventile zu festgelegten Zeiten automatisch geöffnet werden.

³⁹ Technische Systeme E2 bei Herr Stückrath

Da meine Expertise und Interessen primär in der technischen Realisierung und funktionalen Ausgestaltung des Systems liegen, ist die visuelle Gestaltung, die detaillierte Anpassung von Designelementen, nicht meine Stärke. Diese Tätigkeit, oft als Feinabstimmung der Benutzeroberfläche durch Anpassung von Farben und Formen bekannt, wird daher in der Prioritätenliste nach hinten verschoben. Die intensive Bearbeitung und Verbesserung des visuellen Designs ist für die Phase nach der Abgabe dieser Thesis vorgesehen, um sich vollkommen auf die funktionalen Aspekte und die Nutzerfreundlichkeit konzentrieren zu können.

4.5.2 React

React ist eine JavaScript-Bibliothek für die Entwicklung von Benutzeroberflächen, die für ihre Effizienz und Flexibilität in modernen Webentwicklungen geschätzt wird. Sie ermöglicht es Entwicklern, wiederverwendbare UI-Komponenten zu erstellen. Die Grundkonzepte umfassen JSX, eine Syntaxerweiterung, die HTML-ähnlichen Code in JavaScript ermöglicht, Komponenten als Bausteine der UI, State für die Datenhaltung innerhalb der Komponenten, und Props zur Datenübertragung zwischen Komponenten. React's deklarativer Ansatz vereinfacht das Erstellen interaktiver UIs. Gehostet wird der Build der React App über nginx.

Der DOM (Document Object Model) ist eine Programmierschnittstelle, die Webdokumente als Baumstruktur darstellt, wobei jedes Element als Objekt behandelt wird. Dies ermöglicht Entwicklern, Inhalt, Struktur und Stil von Webseiten dynamisch über Skriptsprachen wie JavaScript

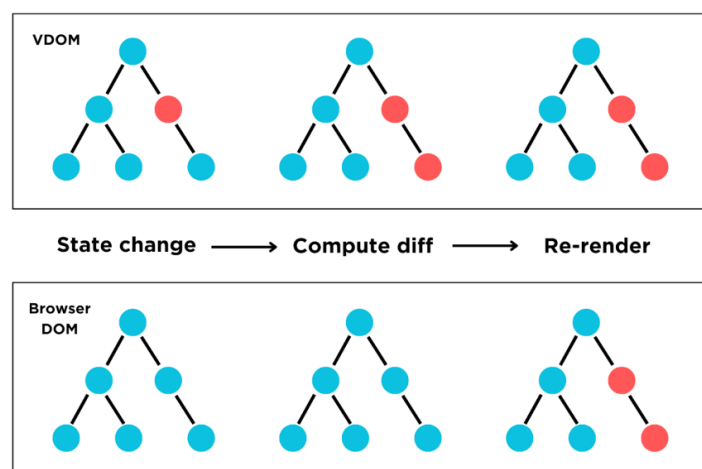


Abbildung 18 - Funktionsweise des Virtual DOM visualisiert

zu ändern. React führt das Konzept des Virtual DOM ein, eine leichtgewichtige Kopie des realen DOM im Speicher. Statt direkt im realen DOM Änderungen vorzunehmen, was performanceintensiv sein kann, führt React alle Änderungen zunächst im Virtual DOM durch. Anschließend vergleicht React den aktuellen Zustand des Virtual DOM mit einer vorherigen Version und berechnet die effizienteste Art, diese Änderungen im realen DOM umzusetzen, dies verbessert die Performance und Effizienz von Webanwendungen durch minimale, optimierte Änderungen im realen DOM. Dieses Verfahren ist in Abbildung ##15 visualisiert.⁴⁰

React wurde von Jordan Walke, einem Software-Ingenieur bei Facebook, entwickelt.⁴¹ Nach einer Onlineumfrage ist React 2023 das zweit-meistgenutzte JavaScript Framework.⁴² Neben Meta nutzen u.a. noch Netflix, Uber, Dropbox, AirBnB und Zendesk React für ihre Webanwendungen.⁴³

Anfangs wurde über getrennten HTML, CSS und JavaScript Code die ersten Weboberflächen implementiert. Diese wurden aber schnell unübersichtlich, weswegen Alternativen gesucht und React gefunden wurde.

4.5.3 WebSocket

Zur Echtzeitkommunikation zwischen Client und Server wird eine dauerhafte WebSocket-Verbindung genutzt. Diese Technik ermöglicht den bidirektionalen Datenaustausch, ohne dass die Verbindung nach jeder Übertragung geschlossen und neu aufgebaut werden muss. Im Vergleich zu her-

⁴⁰ Virtual DOM and Internals, <https://legacy.reactjs.org/docs/faq-internals.html>, (Stand 09.03.2024)

⁴¹ Website von React, <https://legacy.reactjs.org/community/team.html>, (Stand 09.03.2024)

⁴² Most used web frameworks among developers worldwide, as of 2023, <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>, (Stand 09.03.2024)

⁴³ Top 10 Big Companies Using React, <https://careerkarma.com/blog/companies-that-use-react/>, (Stand 09.03.2024)

kömmlichen HTTP-Anfragen, die einen Overhead durch wiederholte Handshakes und Header-Informationen verursachen, sind WebSockets effizienter, schneller und ideal für Echtzeitanwendungen. Es können beispielsweise Sensordaten sekundenaktuell übertragen werden, ohne bei jedem Datenaustausch eine neue Verbindung zu nutzen. Da die Website durch HTTPS verschlüsselt ist, wird die WebSocket-Verbindung zu einer verschlüsselten WebSocket Secure Verbindung aufgerüstet, sodass Daten verschlüsselt übertragen werden.⁴⁴ Jeder einzelne Client wird über seine Session-ID identifiziert, wodurch die Daten jeweils an den richtigen Client übermittelt werden können.⁴⁵

Auf der Backendseite der WebSocket-Verbindung wird die Verbindung zur Datenbank oder zum MQTT-Broker aufgebaut, sodass die Datenintegrität gewahrt bleibt, indem Anmeldedaten, wie beispielsweise zum MQTT-Broker, nicht an den Client ausgeliefert werden. Im WebSocket-Backend-Script sind spezifische Funktionen definiert, die durch Aktionsevents auf der Webseite ausgelöst werden. Diese Funktionen führen daraufhin eine bestimmte Aktion aus und senden ausschließlich das Resultat dieser Aktion zurück an den Client. Beispielsweise können über einen Button die aktuellen Sensorwerte angefordert werden. Der ActionListener hinter dem Button schickt eine Anfrage an das Backend, welches die MQTT Request Nachricht verschickt und nur die Antwort, also die Sensorwerte, an die Website übermittelt, welche die Daten anzeigt.

Es wird eine zentrale WebSocket-Komponente verwendet, die von anderen Komponenten importiert und genutzt wird, anstelle einer eigenen WebSocket-Verbindung für jede Komponente. Dieser Ansatz ermöglicht eine effiziente Verwaltung der WebSocket-Verbindung durch Wiederverwendung ei-

⁴⁴The complete guide to WebSockets with React, <https://ably.com/blog/websockets-react-tutorial>, (Stand 09.03.2024)

⁴⁵ Technische Informatik Q3 bei Herr Betz

ner einzigen Instanz, die über einen Kontext bereitgestellt wird. Diese Methode ähnelt dem Singleton-Prinzip in Java, bei dem eine einzige Instanz systemweit zugänglich gemacht wird, um Mehrfachinstanziierungen zu vermeiden.⁴⁶ Der Backend-WebSocket Code für die Showroom-App ist unter <https://bit.ly/wv-websocket> einsehbar.

4.6 Datenabfrage, -verwaltung und -verarbeitung

4.6.1 Bright Sky

In Deutschland stellt der Deutsche Wetterdienst (DWD) Wetterdaten gemäß dem Open Data Prinzip kostenlos zur Verfügung.⁴⁷ Allerdings gestaltet sich der direkte Zugriff auf diese Daten für Privatpersonen oft als kompliziert. Hier kommt die BrightSky-API ins Spiel, eine von einem Studenten entwickelte Schnittstelle, die speziell darauf ausgerichtet ist, Privatpersonen den Zugang zu den DWD-Daten zu erleichtern.⁴⁸

Die Abfrage der Wetterdaten erfolgt über eine API-Anfrage, bei der der gewünschte Zeitraum, nach dem ISO 8601 Standard, und die spezifische DWD-Messstation oder Koordinaten angegeben werden. Als Antwort erhält man ein JSON-Objekt in stündlichen Intervallen, das verschiedene Wetterparameter enthält. Derzeit wird nur die durchschnittliche Regenwahrscheinlichkeit der nächsten zwei Tage berechnet, das JavaScript ist aber modular aufgebaut, sodass es einfach um weitere Verarbeitungsmethoden ergänzt werden kann. Alle Verarbeitungsmethoden rufen dieselbe Datenbasis ab, sobald diese Datenbasis älter als eine Stunde ist, wird diese aktualisiert. Das JavaScript ist unter <https://bit.ly/wv-weather> aufrufbar.

Ein beispielhaftes JavaScript, dass die BrightSky-API für eine dreitägige Wetterprognose für Frankfurt am Main anfragt und die Antwort in einer

⁴⁶ Informationstechnologie Q2 bei Herr Endreß

⁴⁷ Open Data Bereich des Climate Data Center, <https://www.dwd.de/DE/leistungen/cdc/climate-data-center.html> (Stand 03.03.2024)

⁴⁸ Website von BrightSky, <https://brightsky.dev/> (Stand 03.03.2024)

JSON-Datei abspeichert, kann unter <https://bit.ly/wv-weather-json> angesehen werden. Ein JavaScript, das die JSON-Datei einliest und ausgewählte Parameter in einer MariaDB abspeichert, kann unter <https://bit.ly/wv-json-db> eingesehen werden. Dieses JavaScript funktioniert, wird aber nicht aktiv verwendet, weil es sich als effizienter erwiesen hat, die Wetterdaten bei Bedarf abzufragen, ohne sie abzuspeichern.

4.6.2 MariaDB

Als relationales Datenbankmanagementsystem (DBMS) kommt eine MariaDB zum Einsatz.⁴⁹ Diese Entscheidung basiert auf der Überlegung, eine weit verbreitete Open-Source-Lösung zu nutzen. MySQL wurde ebenfalls in Betracht gezogen, aber verworfen, weil MySQL nur für nicht-kommerzielle Projekte kostenlos einsetzbar ist. Um möglichen Lizenzkosten vorzubeugen, welche bei einer möglichen Kommerzialisierung von Wateringvision anfallen würden, ist die Wahl auf MariaDB gefallen, welches für jegliche Projekte kostenlos nutzbar ist.^{50 51}

In der Sensor-Datenbank hat jedes einzelne Sensorik Modul eine eigene Tabelle. In dieser Tabelle werden alle übertragenen Sensordaten und der Zeitpunkt der Abfrage abgespeichert. In der ##-Datenbank werden alle Daten zu einer Bewässerungslinie abgespeichert, wobei jede einzelne Bewässerungslinie eine eigene Tabelle besitzt. Abgespeichert wird die Bewässerungsdauer, Präferenzzeiten, Überdachungsstatus. Das JavaScript, um alle Sensorik-Module abzurufen und ihre übermittelten Daten in ihren Tabellen abzuspeichern, ist unter <https://bit.ly/wv-sensor-db> aufrufbar.

⁴⁹ Technische Informatik Q4 bei Herr Betz

⁵⁰ Website von MariaDB; <https://mariadb.org/de/>; (Stand 05.03.2024)

⁵¹ MySQL Produkte; <https://www.mysql.com/products/>; (Stand 05.03.2024)

Das JavaScript zum Einlesen einer JSON-Datei und abspeichern in einer MariaDB Datenbank ist fertig und war im Einsatz. Allerdings ist einfach leichtgewichtiger, die Wetterdaten bei Bedarf abzufragen. Langfristig sollen die Wetterdaten in einer MongoDB abgespeichert werden, was in ## genauer erklärt wird.

Für die Development-Einsicht in die Datenbanken wird das Tool phpMyAdmin eingesetzt, dies ist ebenfalls eine Open-Source-Lösung, die eine benutzerfreundliche, grafische und webbasierte Oberfläche zur Datenbankinteraktion bietet.⁵²

Zu Beginn des Projekts wurde mit SQLite gearbeitet, doch die Vorteile eines Server/Client-basierten DBMS wie MariaDB, insbesondere hinsichtlich Skalierbarkeit und Netzwerkkompatibilität, haben letztendlich überzeugt. Ein Video, was die erste erfolgreiche Implementierung einer Web-Datenbank-Verbindung zeigt, kann unter <https://bit.ly/wv-webdb> angesehen werden.

4.6.3 Event-basierte Bewässerung

Eine Event-basierte Bewässerung ist die erste automatische Bewässerungssteuerung, diese wurde absichtlich leichtgewichtig gehalten, weil dieser Teil des Systems langfristig komplett umgestalten werden soll, genaueres dazu wird in ## Ausblick KI erläutert.

Eine einfache, aber trotzdem effiziente Bewässerungssteuerung, ist konstant auf definierte Events zu überwachen. Dabei werden alle Bewässerungslinien aus der Datenbank geladen und einzeln verarbeitet. Zu jeder

⁵² Website von phpMyAdmin; <https://www.phpmyadmin.net/>; (Stand 05.03.2024)

Bewässerungslinie gehört ein Feuchtigkeitssensor, Ventil und ein Soll-Wert, dabei wird überprüft, ob der aktuell gemessene Feuchtigkeitswert der Erde unter dem Soll-Wert liegt. Wenn dies der Fall ist und die durchschnittliche Regenwahrscheinlichkeit der nächsten zwei Tage unter einem definierten Wert, derzeit 50% liegt, wird eine definierte Bewässerungsdauer, derzeit 90 Sekunden, veranlasst. Das Skript ist modular, skalierbar und konfigurierbar geschrieben. Es können einfach weitere Verarbeitungsmethoden hinzugefügt werden, es ist irrelevant, wie viele Bewässerungslinien angelegt sind und die Konstanten können einfach angepasst werden. Das JavaScript ist unter <https://bit.ly/wv-action> aufrufbar.

4.7 Wassermanagement

4.7.1 Wasserquellen

Regenwasser ist als Primärquelle für die Bewässerung vorgesehen. Diese Wahl ist nicht nur nachhaltig, sondern auch umweltschonend, da sie die natürlichen Wasserressourcen schont und die Abhängigkeit von der öffentlichen Wasserversorgung reduziert. Unser Haus besitzt eine Dachfläche von 60m², die zum Sammeln von Regenwasser geeignet ist. Das Wasser wird durch Regenrinnen und Fallrohre in eine Zisterne geleitet. Es ist jedoch zu beachten, dass das gesammelte Wasser auch für die Toilettenspülung genutzt wird. Da es insbesondere im Sommer, wenn der Bewässerungsbedarf am höchsten ist, oft nicht genug regnet, muss sie als Notlösung zeitweise mit Leitungswasser aufgefüllt werden.

4.7.2 Speicherung und Transport des Wassers

Die Nutzung einer Zisterne bietet mehrere Vorteile: Sie ermöglicht das Sammeln und Speichern von Regenwasser, was sowohl kosteneffizient als auch umweltfreundlich ist, da sie die Abhängigkeit von der öffentlichen Wasserversorgung reduziert und eine nachhaltige Ressourcennutzung fördert.

Die Zisterne umfasst ein Füllvolumen von 4.500 Litern. An dieser ist eine unterdruckgesteuerte Zisternenpumpe mit einer Förderleistung von 10 Bar angeschlossen. Durch den Druckabfall, der durch das Öffnen eines Magnetventils entsteht, wird die Pumpe aktiviert. Zur Stabilisierung des Drucks im System ist ein Druckausgleichsbehälter installiert, welcher Druckschwankungen auszugleicht.

Für die Wasserleitung vom Keller bis zur Gartenhütte wurden bereits vorhandene Leerrohre genutzt, durch die lediglich ein Schlauch gezogen werden musste. Von der Gartenhütte aus beginnen die einzelnen Bewässerungslinien, die verschiedene Teile des Gartens versorgen.

4.8 Modellhafte Darstellung / Showroom

Wenn Wateringvision gepitcht bzw. geprüft werden soll, ist es wichtig das System in kurzer Zeit jedermann verständlich zu machen. Anfangs sollte ein physisches Modell gebaut werden, welches ortsflexibel eingesetzt werden kann. Dieses physische Modell birgt zwei Nachteile. Einerseits ist die Logistik eines solchen Modells nicht zu unterschätzen. Der zweite, überwiegende Nachteil ist, dass es nur funktioniert, wenn man vor Ort ist.

Deswegen wurde sich ein digitaler Showroom ausgearbeitet. In diesem Showroom ist ein Livestream einer

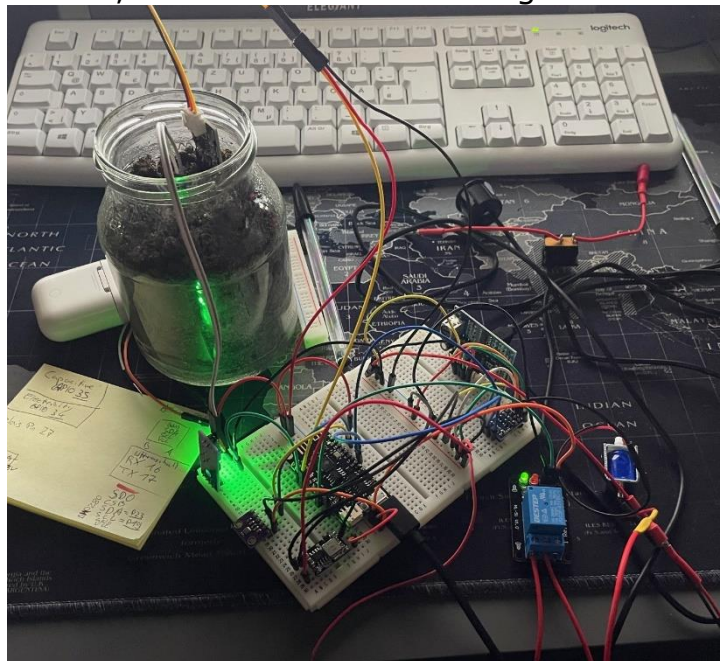


Abbildung 19 - Prototyp Aufbau des Showroom Moduls

Pflanze mit Feuchtesensoren, die Werte werden auf der Website ständig aktualisiert. Ebenfalls wird es möglich sein, selbstständig eine Bewässerung

zu starten. Dazu werden die Wetterdaten für den Nutzerstandort abgerufen und angezeigt. In einem sichtbaren Log sind alle Kommunikations- und Statusmeldungen einsehbar.

Der Code des Showrooms fasst die meisten Funktionen aller Module in einem zusammen. Die Firmware ist unter <https://bit.ly/wv-showroom> aufrufbar.

5 Ausblick

5.1 Mögliche Optimierungen

5.1.1 Weiterentwicklung der Website

Ein wesentliches Merkmal der Benutzeroberfläche ist die Integration und Anzeige aktueller Wetterdaten aus der Datenbank, die eine zentrale Rolle in der Planung und Durchführung effizienter Bewässerungsstrategien spielen. Darüber hinaus wird es möglich sein, die Kommunikation zu den Modulen zu monitoren, um die Zuverlässigkeit und Stabilität des Systems zu gewährleisten.

Um die Einrichtung von Bewässerungslinien zu vereinfachen, ist die Implementierung von Vorlagen vorgesehen. Diese ermöglichen es den Nutzern, schnell und effizient neue Bewässerungskonfigurationen anzulegen. In Verbindung mit der Einbindung einer Künstlichen Intelligenz zur Auswertung der Sensordaten können Benutzer Hinweise in natürlicher Sprache geben, die von der KI zur Optimierung der Bewässerungsprozesse genutzt werden.

Für die Wartung und das Management des Systems sind verschiedene Modi geplant, darunter ein Wartungsmodus, der es ermöglicht, systemweit alle Ventile zu schließen. Zusätzlich können individuelle Aktionen erstellt und bearbeitet werden, wie etwa die Automatisierung des Auffüllens der

Zisterne mit Frischwasser, sobald ein bestimmter Füllstand unterschritten wird.

Eine umfassende Einsicht in die Datenbank wird ebenfalls möglich sein, wobei Sensordatenschwankungen grafisch dargestellt werden können. Die Optimierung der Benutzeroberfläche für Hochkantgeräte ist ein weiteres geplantes Feature, um die Benutzerfreundlichkeit auf mobilen Endgeräten zu verbessern.

Mit der Einführung eines Accountsystems wird das Bewässerungssystem auch für die Nutzung durch mehrere Benutzer vorbereitet. Systemadministratoren erhalten dabei die Möglichkeit, Mitarbeiterkonten zu verwalten, zu warten und Berechtigungen zu managen, wobei jeder Unterbenutzer nur Zugriff auf seine relevanten Aufgabenbereiche hat.

5.1.2 Hardwaretechnische Weiterentwicklung der Module

In der aktuellen Phase befinden sich die vorgestellten Module noch im Prototypenstadium. Die einzelnen Komponenten werden momentan manuell auf einer Lochrasterplatine verdrahtet. Diese Herangehensweise ermöglicht zwar eine flexible Entwicklung und Erprobung verschiedener Konfigurationen, jedoch ist das Ziel für die nächste Generation, die Entwicklung einer gedruckten Leiterplatte (PCB) voranzutreiben. Eine PCB würde nicht nur den Zusammenbau der Module vereinfachen, sondern auch deren Professionalität und Zuverlässigkeit erhöhen.

Des Weiteren ist geplant, die Gehäuse der Module mithilfe von 3D-Drucktechnologien herzustellen. Dieser Ansatz eröffnet neue Möglichkeiten in Bezug auf das Design und Aussehen der Module, da 3D-Druck eine hohe Flexibilität in der Gestaltung ermöglicht. Darüber hinaus wird angestrebt,

die Module witterungsfest zu gestalten, sodass sie problemlos im gesamten Gartenbereich eingesetzt werden können, ohne dass ihre Funktionalität durch Umwelteinflüsse beeinträchtigt wird.

Ein weiterer Aspekt der zukünftigen Entwicklung betrifft die Integration zusätzlicher Sensoren in die Sensorikmodule. Überlegt wird, Messwerte wie Temperatur oder Luftfeuchtigkeit zu erfassen, um ein umfassenderes Bild der Gartenbedingungen zu erhalten. Parallel dazu wird die Möglichkeit untersucht, präzisere Sensortechnologien einzusetzen, um noch genauerer Daten für die Bewässerungssteuerung zu sammeln.

Ein wesentliches Ziel für die Zukunft ist es, einen autarken Betrieb der Module zu ermöglichen. Aktuell müssen alle Module über Kabel mit Strom versorgt werden. Geplant ist die Integration eines Akkus, der Energie speichert und somit einen temporären Betrieb auch bei Stromausfällen gewährleistet. Um eine vollständige Autarkie zu erreichen, wird die Eignung von Solarpaneelen geprüft, um den Strombedarf der Module zu decken. Dies würde eine kontinuierliche und umweltfreundliche Energieversorgung sicherstellen und den Installationsaufwand im Garten minimieren.

5.1.3 Softwaretechnische Weiterentwicklung der Module

Für die nächste Generation der Wateringvision-Software ist geplant, eine Funktion für Over-the-Air (OTA) Updates zu implementieren. Dies bedeutet, dass die ESP32-Module aktualisiert werden können, ohne eine direkte serielle USB-Verbindung zu einem Computer herstellen zu müssen. Stattdessen wird der ESP32 regelmäßig einen Update-Server abfragen, um zu überprüfen, ob eine neue Version der Firmware verfügbar ist. Bei Verfügbarkeit eines Updates wird dieses automatisch heruntergeladen und auf dem Modul installiert. Um diesen Prozess zu ermöglichen, muss der aktualisierte Code lediglich kompiliert und die resultierende Binärdatei auf dem Update-Server bereitgestellt werden. Diese Funktionalität zielt darauf ab,

Fernzugriffsmöglichkeiten zu schaffen, um etwaige Bugs effizient beheben zu können, ohne physischen Zugriff auf die Module zu benötigen.

Ein weiteres ambitioniertes Ziel für die Zukunft ist die Schaffung eines Ökosystems, in dem die Module untereinander kommunizieren können. Ein solches Netzwerk würde es den Modulen ermöglichen, bei der Erstinstallation WLAN-Zugangsdaten untereinander auszutauschen oder Daten in einem Art Ringsystem zu übermitteln. In diesem System müsste nur ein Modul des Rings eine aktive WLAN-Verbindung unterhalten, während die anderen Module Informationen austauschen könnten, ohne direkt mit dem Internet verbunden zu sein. Es wird erwartet, dass diese Vernetzung höchstwahrscheinlich über Bluetooth-Kommunikation realisiert wird, wobei dieses Thema in zukünftigen Entwicklungsphasen weiter erforscht werden soll. Der daraus resultierende Vorteil soll sein, dass man weniger WLAN Abdeckung benötigt, um die Module anzusprechen.

Die Nutzung der dualen Prozessorkerne der ESP32 bietet zudem die Möglichkeit einer nebenläufigen Programmierung. Das angestrebte Ziel ist es, einen Prozessorkern speziell für die Durchführung der Messungen zu verwenden, während der zweite Kern sich um andere Aufgaben wie WLAN-Verwaltung, MQTT-Kommunikation, OTA-Updates und ähnliche Funktionen kümmert. Diese Aufteilung der Verantwortlichkeiten soll eine effiziente und zuverlässige Funktion der Module gewährleisten, indem sie die gleichzeitige Ausführung verschiedener Aufgaben ohne gegenseitige Beeinträchtigung ermöglicht.

5.1.4 Bessere Netzwerksicherheit (geplant)

Da kommt noch mehr protection keine sorge

5.1.5 MongoDB

Als Ergänzung zu MariaDB soll als zweites DBMS MongoDB eingeführt werden. MongoDB nutzt das BSON (Binary JSON) Speichermodell, welches eine effiziente Speicherplatznutzung und Abfrageschnelligkeit ermöglichen.⁵³ MongoDB speichert Daten in einem dokumentenorientierten Format, welches sich von traditionellen relationalen Datenbanken unterscheidet. Diese Eigenschaft ermöglicht eine flexible Datenmodellierung und -speicherung, die besonders für Anwendungen mit umfangreichen JSON-Daten geeignet ist⁵⁴.

Zum Einsatz soll MongoDB kommen um die Daten von BrightSky langfristig abzuspeichern. Dabei wird jedes JSON-Objekt, welches alle Parameter für eine Stunde beinhaltet abgespeichert.

Es werden vorsorglich alle Wetterdatenparameter abgespeichert, auch diese, welche derzeit als nicht relevant eingeschätzt werden. Diese umfangreiche Datenmenge wird gesammelt, um eine möglich große Datenbasis zu haben, welche für ein mögliches Training einer künstlichen Intelligenz zu unterstützen. (## Verweis KI Ausblick)

Die direkte Anbindung von MongoDB an Node.js ermöglicht eine effiziente Datenverarbeitung und -speicherung, was es zu einer hervorragenden Wahl für die Entwicklung moderner Web-Applikationen macht.⁵⁵

⁵³ Deepak Vohra, "Using MongoDB Database", 2016, S. 167-200

⁵⁴ Ciprian-Octavian Truică, A. Boicea, Ionut Trifan, "CRUD Operations in MongoDB", 2013, S. 347-350

⁵⁵ Deepak Vohra, "Creating an Apache Hive Table with MongoDB", 2015, S. 405-426

5.1.6 Komplette Installation im Garten

Aktuell ist im Rahmen des Wateringvision-Projekts lediglich eine Bewässerungslinie in Betrieb. Diese Beschränkung ist primär auf zeitliche Restriktionen sowie die Wetterbedingungen im Winter und Frühjahr zurückzuführen, welche eine umfassende Implementierung vor der Deadline der BLL limitieren. Trotz dieser vorübergehenden Einschränkung ist es langfristig definitiv vorgesehen, den gesamten Garten mit der entwickelten Hardware auszurüsten.

5.1.7 Scalable Cloudhosting

Aktuell basiert die IT-Infrastruktur des Wateringvision-Projekts auf einem virtuellen privaten Server (VPS) unter Linux, welcher sich in der Praxis als funktional erwiesen hat. Trotz der aktuellen Funktionalität und Stabilität dieses Hosting-Ansatzes besteht ein ausgeprägtes persönliches Interesse daran, mich tiefergehend mit dem Thema des skalierbaren Cloud-Hostings auseinanderzusetzen. Ziel ist es, den Hosting-Prozess weiter zu optimieren und potenziell die Leistungsfähigkeit, Skalierbarkeit und Wartbarkeit des Gesamtsystems zu verbessern.

5.1.8 Künstliche Intelligenz zur Datenauswertung

Ein zentraler Aspekt des Wateringvision-Projekts, der derzeit besondere Aufmerksamkeit erfordert, ist der Autopilot-Algorithmus, welcher sich noch in einer frühen Beta-Phase befindet. Dieser Algorithmus spielt eine entscheidende Rolle, da er alle gesammelten Daten analysiert, um eine optimale Bewässerungsstrategie zu entwickeln. Momentan basiert der Algorithmus auf einem einfachen Skript, das zwar funktional ist, aber in Bezug auf Optimierung und Effizienz noch nicht den gewünschten Stand erreicht hat. Die Entwicklung ist hauptsächlich aufgrund zeitlicher Restriktionen noch nicht so weit fortgeschritten, wie ursprünglich angedacht.

Obwohl für die Auswertung der gesammelten Daten streng genommen keine künstliche Intelligenz erforderlich ist, besteht aufgrund persönlichen Interesses der Plan, sich zukünftig intensiver mit dem Einsatz von KI-Methoden zu befassen. Der Einsatz künstlicher Intelligenz könnte nicht nur die Präzision der Datenauswertung verbessern, sondern auch dazu beitragen, fortgeschrittenere und adaptivere Bewässerungsstrategien zu entwickeln, die auf komplexen Mustern und Prognosen basieren.

5.1.9 Überprüfung des Zisternenfüllstandes

Ein Skript soll automatisch den Zisternenfüllstand überprüfen, dabei wird in regelmäßigen Abständen der Ultraschall-Sensor abgerufen und der Ist-Wert mit einem flexibel einstellbaren Soll-Wert verglichen.

Falls der Füllstand der Zisterne kritisch niedrig ist, wird zunächst die Regenprognose konsultiert, um zu ermitteln, ob in naher Zukunft mit Regen zu rechnen ist. Diese effizienzorientierte Maßnahme dient dazu, Frischwasser nur dann zu nutzen, wenn es unbedingt notwendig ist. Sollte kein Regen erwartet werden und der Wasserstand kritisch niedrig sein, aktiviert das Skript das Ventil zur Frischwasserzufuhr. In diesem Fall wird die Zisterne jedoch nicht vollständig aufgefüllt. Stattdessen wird nur so viel Frischwasser zugeführt, wie unbedingt nötig ist.

5.1.10 Präventive Maßnahmen gegen einen Wasserschaden

Um die Zuverlässigkeit und Sicherheit des Wateringvision-Systems weiter zu erhöhen, ist die Implementierung von Kontrollmechanismen geplant, die einen fehlerhaften Wasseraustritt erkennen können. Ziel dieser Maßnahmen ist es, potenzielle Wasserschäden präventiv zu verhindern.

Der Kern dieser präventiven Strategie besteht darin, einen Wasserdurchflusssensor zu integrieren, der den realen Wasserdurchlauf kontinuierlich misst. Diese Messdaten werden dann mit den im System hinterlegten, geplanten Wassermengen abgeglichen. Bei der Detektion eines ungewöhnlich hohen Wasserdurchlaufs, der auf einen fehlerhaften Wasseraustritt hindeutet, soll umgehend ein Hauptventil aktiviert werden, das die komplette Wasserzufuhr automatisch stoppt.

5.1.11 Integration von Strommanagement

Die Integration eines Strommanagements in das Wateringvision-System stellt derzeit noch eine Vision in der Findungsphase dar. Die Konzeption und Umsetzung dieses Aspekts wird in den Fokus rücken, sobald das Projekt den schulischen Rahmen verlässt und in eine Phase der weiteren professionellen Entwicklung eintritt.

Das Ziel des Strommanagements ist es, den Energieverbrauch des Systems sowie die Energiegewinnung, beispielsweise durch Solarpaneele, präzise zu erfassen und zu überwachen. Diese Daten sollen nicht nur erfasst, sondern auch sinnvoll für den Nutzer aufbereitet werden. Damit würde das System nicht nur Informationen über den Wasserverbrauch und die Bewässerungseffizienz liefern, sondern auch Einblicke in die Energiebilanz geben. Nutzern soll dadurch ermöglicht werden, den Energieverbrauch ihres Bewässerungssystems zu optimieren und die Nutzung erneuerbarer Energiequellen effektiv zu integrieren und zu überwachen.

6 Fazit

-ich bin mit meiner Arbeit zufrieden.

-ich weiß, dass ich noch viele offene baustellen habe, aber mein oberstes ziel ist es eine funktionierendes gesamtsystem bis zum 21.03.2024 zu haben.

-ich habe unglaublich viel gelernt

-mir haben die vielen bugs das leben im letzten jahr nicht immer leicht gemacht, aber wenn ein bug, egal wie lange dieser einen zum verzweifeln gebracht hat, löst, ist man ein stück selbstsicherer (der selbe bug wird mich in zukunft nichtmehr diese zeit kosten)

-das themenfeld was mir am meisten freude bereitet hat ist „Meine Interpretation der Technischen Informatik im 21 Jahrhundert“ (den satz bitte drinne behalten)

-mich fasziniert dieser themenbereich der Informatik/Elektrotechnik und ich habe das wissen was ich in der schule gelernt habe auf ein modernes Internet of Things Projekt übertragen

-Ich habe meine aufgabe als „projekt developer“ gesehen, ich bin dafür verantwortlich, dass was ich mir ausdenke auch irgendwie umsetze, dabei ist der weg NICHT das ziel.

-zur technischen informatik gehört:

die module. Die entwicklung von mikrocontroller projekten, alles was dazu gehört, von dem prozess das man sich etwas im kopf ausdenkt, die ersten codes auf die esp zu laden, und der code stück für stück optimiert/verbessert/erweiter wird

-arduino sprache <3 (es sollte erwähnt werden, dass der „Arduino C++ Dialekt“ meine absolute Lieblings Programmiersprache ist, sie ist so genial für mikrocontroller und ich finde es schade das im Beruflichen Gymnasium so wenig mit dieser sprache gearbeitet wird.)

-das themenfeld „Embedded Linux Development“ fand ich ebenfalls sehr interessant und herausfordernd für mich, weil ich bisher bis auf in meinem praktikum ein paar tage, noch keine ernsthaften berührungspunkte mit der linux konsole hatte.

-auch mit javascript hatte ich keine Development-berührungspunkte, aber ich würde sagen, dass es eine gute wahl für mich war, sich auf eine programmiersprache für das backend festzulegen. In zukunft wird sich intensiver mit python auseinanderzusetzen.

-es gab auch arbeit welche erledigt werden muss, aber nicht wirklich spaß macht dazu zählt netzwerksicherheit oder datenbanken.

was für mich die mit abstand die größte herausforderung war und mir absolut garkein spaß gemacht hat, war das „wissenschaftliche“ arbeiten und dokumentieren. Für mich macht eine klassische wissenschaftliche arbeitsweise bei der entwicklung eines modernen it systems keinen sinn und aus der zeit gefallen.

-war teils sehr gestresst was vorallem an den bugs lag. Sie treten dann auf wenn man nicht mit ihnen rechnet und können einem den ganzen nachmittag klauen.

Die zentrale Lektion, die ich während der Entwicklung von Wateringvision gelernt habe, betrifft den Umgang mit der Angst vor dem Unbekannten. Besonders herausfordernd waren Momente, in denen ich innovative Ideen hatte, mir jedoch das technische Wissen für deren Umsetzung fehlte. Diese Erfahrungen lehrten mich, über meinen aktuellen Kenntnisstand hinaus zu denken, nicht nur in den Grenzen dessen zu verharren, was ich bereits beherrschte, sondern zu erkunden, was ich erlernen möchte.

Aktuell ist im Rahmen des Wateringvision-Projekts lediglich eine Bewässerungslinie in Betrieb. Diese Beschränkung ist primär auf zeitliche Restriktionen sowie die Wetterbedingungen im Winter und Frühjahr zurückzuführen, welche eine umfassende Implementierung vor der Deadline der BLL limitieren. Trotz dieser vorübergehenden Einschränkung ist es langfristig definitiv

vorgesehen, den gesamten Garten mit der entwickelten Hardware auszurüsten.

Obwohl für die Auswertung der gesammelten Daten streng genommen keine künstliche Intelligenz erforderlich ist, besteht aufgrund persönlichen Interesses der Plan, sich zukünftig intensiver mit dem Einsatz von KI-Methoden zu befassen. Der Einsatz künstlicher Intelligenz könnte nicht nur die Präzision der Datenauswertung verbessern, sondern auch dazu beitragen, fortgeschrittenere und adaptivere Bewässerungsstrategien zu entwickeln, die auf komplexen Mustern und Prognosen basieren.

Abbildungsquellenverzeichnis

Abb. 0, Hochtaunusschule Logo, https://hochtaunusschule.de/wp-content/uploads/2021/06/cropped-htscom_header_image_logo.png, (Stand 02.10.2023)

Abb. 1, Karte des Palmengartens, selbstständig fotografiert

Abb. 2, Zisternenanlage, selbstständig fotografiert

Abb. 3, Bewässerungsmethoden, selbstständig zusammengestellt, Icons von flaticon.com, erstellt von den Künstlern itim2101, srip und freepik

Abb. 4, DWD-Wetterstationen, https://www.dwd.de/DE/fachnutzer/landwirtschaft/appl/stationskarte/_node.html, (Stand 10.03.2024)

Abb. 5, SWE-Benchmark, Daten von <https://www.swebench.com/#>, Grafik aus <https://youtu.be/AgyJv2Qelwk?t=104>, (Stand 15.03.2024)

Abb. 6, Error-Meldung, selbstständig gescreenshotet

Abb. 7, Benutzerzugriff, selbstständig zusammengestellt, Icons von <https://nodejs.org/docs/v0.10.2/logos/>, <https://www.cleanpng.com/png-ubuntu-linux-distribution-long-term-support-canoni-1023184/>, <https://iconduck.com/icons/14184/raspberry-pi>, <https://www.debian.org/logos/index.de.html>, <https://1000logos.net/linux-logo/>, <https://flaticon.com/autoren/kerismaker> (Alle Links vom Stand 10.03.2024)

Abb. 8, Backen-Flowchart, selbstständig zusammengestellt, Icons von <https://www.dwd.de/EN/service/copyright/DWD-Logo.html>, <https://www.flaticon.com/autoren/smashicons>, <https://www.flaticon.com/autoren/freepik>, Icon vom ESP wurde von DALL-E generiert (Alle Links vom Stand 11.03.2024)

Abb. 9, PIN-Layout ESP32, <https://prilchen.de/belegungsplan-esp32-lolin32-lite/>, (Stand 01.03.2024)

Abb. 10, Technologieschema Sensorik, in Fritzing selbstständig zusammengestellt

Abb. 11, Sensordatenabfrage, selbstständig zusammengestellt, Server-Icon von <https://www.flaticon.com/autoren/pixel-perfect> (Stand 13.03.2024)

Abb. 12, Eichmessung Wasser, selbstständig fotografiert

Abb. 13, Technologieschema Aktorik, in Fritzing selbstständig zusammengestellt

Abb. 14, Ventilsteuerungslogik, selbstständig zusammengestellt, Server-Icon von <https://www.flaticon.com/autoren/pixel-perfect> (Stand 13.03.2024)

Abb. 15, Feuchtesensoren, selbstständig zusammengestellt, Abbildungen von <https://www.ebay.de/itm/354016225430>, <https://youtu.be/BKp4ib1UQj8>

(Thumbnail) (Stand 01.03.2024)

Abb. 16, Netzwerkzugehörigkeit, selbständig zusammengestellt, Icons von <https://de.vexels.com/png-svg/preview/141379/ubuntu-logo>, <https://mariadb.com/about-us/logos/>, <https://nodejs.org/docs/v0.10.2/logos/>, <https://medium.com/geekculture/what-is-nginx-2edfdad3722b>, <https://blog.wildix.com/de/einfuehrung-in-react-und-react-native/>, <https://www.flatironcon.com/de/autoren/freepik>, <https://www.solidblue.de/mqtt-kommunikation-zwischen-melarecon-und-solidblue/> (Alle Links vom Stand 10.03.2024)

Abb. 17, MQTT-Übersicht, Dennis Schröder am 13.08.2020, <https://schroeder-dennis.de/mqtt/das-protokoll-mqtt-einfach-erklaert-und-vorgestellt/>, (Stand 13.03.2024)

Abb. 18, Virtual DOM, Sean Blundell am 07.07.2023, <https://www.babel.com/en/magazine/build-your-own-react-episode-2>, (Stand 13.03.2024)

Abb. 19, Showroom Aufbau, selbstständig fotografiert

Eigenständigkeitserklärung

Ich versichere hiermit, dass ich die Besondere Lernleistung selbständig und ohne fremde Hilfe verfasst habe;

dass ich die Gedanken anderer Autoren und literarische Zitate, die ich direkt oder indirekt aus anderen Quellen übernommen habe, an den entsprechenden Stellen im Dokument deutlich gekennzeichnet habe;

dass ich oder eine andere Person meine Besondere Lernleistung oder Teile davon nicht zu anderen Prüfungen eingereicht habe.

Mir ist bekannt, dass eine falsche Erklärung rechtliche Konsequenzen nach sich ziehen wird.

Frankfurt am Main, 21.März 2024

Unterschrift

Anhang