**IoT-Projekt Bloombuddy**

**1. Allgemeine Informationen**

* **Projektname:** *Bloombuddy*
* **Ersteller:** Jerrit Schnaible
* **Datum:** *11.03.2025 bis 06.05.2025*

**2. Projektbeschreibung**

**Kurzbeschreibung:**  
Mein Projekt ist eine Smarte-Pflanzenbewässerung, wobei die Temperatur, die Luftfeuchtigkeit, die Helligkeit am Standort der Pflanze, sowie die Bodenfeuchtigkeit der Erde gemessen und automatisch eine Bewässerung gesteuert wird. Zusätzlich wird in meinem Projekt der Füllstand des Wassertanks gemessen und eine entsprechende Push-Benachrichtigung ausgegeben, wenn der Füllstand des Tanks unter den 20% Wert fällt. Die Werte werden über Node-Red visualisiert. Alle Messdaten werden in einer Datenbank gespeichert, um später darauf zurückgreifen zu können.

**3. Anforderungen und Funktionalitäten**

✅ **Sensorik**

**AHT21:** Dieser Sensor misst die Luftfeuchtigkeit und die Temperatur im Raum

**BH1750:** Dieser Sensor misst die Helligkeit am Standort der Pflanze

**VL53L0X TOF:** Dieser Sensor misst über TOF den aktuellen Füllstand des Wassertanks

**Kapazitiver Bodenfeuchtigkeitssensor:** Dieser Sensor misst die Bodenfeuchtigkeit in der Pflanzerde

✅ **Aktoren-Steuerung**

**AM325 Mini Pump:** Bewässerung der Pflanze, wenn die Pumpe angesteuert wird

✅ **Webinterface & Benutzerinteraktion**

Das Webinterface wird über Node-Red realisiert. Hier werden die aktuellen Messwerte angezeigt und für den Benutzer visualisiert. Der Benutzer kann also die aktuelle Helligkeit, die Luftfeuchtigkeit, die Temperatur, die Bodenfeuchtigkeit und den Füllstand des Wassertanks direkt ablesen. Außerdem bekommt der Benutzer eine Push-Benachrichtigung aufs Handy wenn der Füllstand des Wassertanks unter 20% fällt.

**4. Benötigte Komponenten**

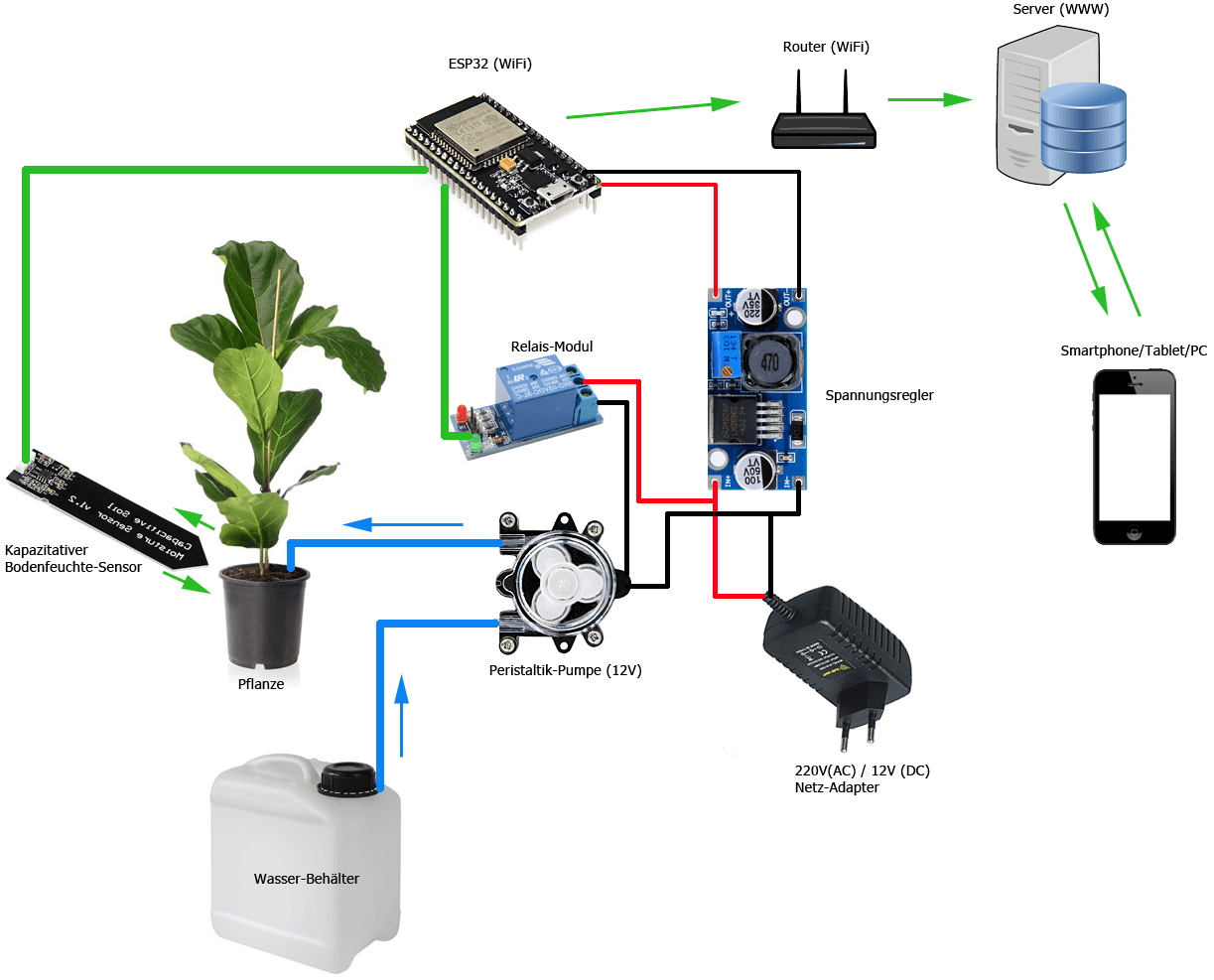
**4.1 Hardware**

| **Komponente** | **Modell/Typ** | **Funktion** |
| --- | --- | --- |
| **Mikrocontroller** | ESP32 S3 WROOM 1 | Sensoren auswerten und Pumpe ansteuern  Kommunikation mit Node-Red |
| **Sensor 1** | VL53L0X Time-of-Flight | Füllstand des Wassertanks überwachen |
| **Sensor 2** | AHT21 digitales Temperatur-Feuchtigkeitssensor-Messmodul mit I2C-Kommunikation | Messen der Temperatur und der Luftfeuchtigkeit im Raum |
| **Sensor 3** | Capacitive soil moisture sensor v2.0 kapazitiver Bodenfeuchtigkeitssensor | Messen der Bodenfeuchtigkeit in der Pflanzenerde |
| **Sensor 4** | BH1750 Lichtintensitätsbeleuchtungsmodul | Messen der Helligkeit am Standort der Pflanze |
| **Aktor 1** | AM325 Mini Pump  DC 3V-6V Peristaltische Dosierpumpe Selbstansaugend | Bewässerung der Pflanze |
| **Stromversorgung** | 5V Netzteil | Spannungsversorgung des ESPs, der Sensoren und der Pumpe |
| **Weitere Bauteile** | 3,3V 1 Kanal Relaismodul Low Level Trigger | Ansteuern der Pumpe über das Relais |
|  | Breadboard | Zum Aufbauen der Komponenten |
|  | Jumpercable in verschiedenen Längen | Zum Verbinden und anschließen der einzelnen Bauteile |
|  | PVC-Schlauch und Schlauchverbinder | Um das Wasser zur Pflanze zu führen |

**4.2 Software & Datenbank**

| **Komponente** | **Technologie** | **Funktion** |
| --- | --- | --- |
| **Microcontroller-Code** | Micropython | Speziell für Systeme mit begrenzten Ressourcen  Benötigt nur wenig Speicherplatz  Für ARM-basierte Chips |
| **Webinterface** | Node-Red | Visualisierung der Messwerte |
| **Datenbank** | MariaDB | Speicherung der Daten |

**5. Systemarchitektur**

****

Die Abbildung weicht leicht von meinem Projekt ab, ist aber dennoch sehr vergleichbar. Die Änderungen, die ich hinzufügen würde, wäre das ich noch einen TOF-Sensor über I2C-Bus an den ESP32 anschließen würde, um den Füllstand des Wassertanks zu messen. Zusätzlich noch einen AHT21-Sensor zum Messen der Luftfeuchtigkeit und der Temperatur im Raum, dieser ist ebenfalls über den I2C-Bus mit dem ESP32 verbunden. Außerdem verbaue ich einen BH1750 zum Messen der Helligkeit am Standort der Pflanze, dieser ist auch an dem I2C-Bus angeschlossen.

**6. Zeitplanung (Meilensteine)**

| **Datum** | **Aufgabe** |
| --- | --- |
| KW 11 | Festlegung der Projektziele  Ermittlung der Anforderungen an das System (z.B. Sensoren für Feuchtigkeit, Temperatur)  Erstellung eines Projektplans mit Meilensteinen und Zeitrahmen |
| KW 12 | Definition der notwendigen Sensoren und Aktoren  Auswahl der geeigneten Hardwarekomponenten (z.B. Mikrocontroller, Pumpe)  Erstellung eines technischen Designs für das Bewässerungssystem  Bestellung der notwendigen Komponenten (z.B. Sensoren, Mikrocontroller, Pumpen) |
| KW 13-14 | Entwicklung der Software  Implementierung der Bewässerungslogik mit Micropython  Integration der Sensordaten zur Steuerung der Bewässerung  Durchführung von Tests zur Sicherstellung der korrekten Funktion  Implementierung der Daten in Node-Red, Visualisierung in Node-Red  Ablegen der gemessen Werte in einer Datenbank |
| KW 15 | Zusammenbau des Bewässerungssystems mit allen Komponenten  Durchführung von umfassenden Tests des gesamten Systems unter realen Bedingungen |
| KW 16 | Anpassung der Parameter zur Verbesserung der Effizienz und Zuverlässigkeit  Kalibrierung der Sensoren und Anpassung der Aktoren |
| KW 17 | Erstellung einer detaillierten Dokumentation des Systems, inklusive Schaltpläne und genauer Funktionsbeschreibung unter realen Bedingungen |
| KW 18 | Durchführung der finalen Tests und Abnahme des Systems  Installation des Systems am Einsatzort (an der Pflanze) |
| KW 19 | Reserve zum Ausgleich auftretender Probleme oder zur Verbesserung des Systems |

**7. Offene Fragen & Herausforderungen**

**Batterie- und Stromprobleme:** Unzureichende Batterieleistung oder Stromausfälle können das System lahmlegen, zum Beispiel bei der Versorgung über eine Powerbank

**Über- oder Unterbewässerung:** Falsche Einstellungen, fehlerhafte Sensoren oder unvollständige Daten können zu Über- oder Unterbewässerung führen, was der Pflanze schädigen kann

**Ungleichmäßige Wasserverteilung:** Ungleichmäßige Wasserverteilung kann die Pflanze unterschiedlich stark bewässern

**Programmierung:** Analogwertverarbeitung kann komplex sein, zudem kann die Darstellung der Daten in Node-Red eine Schwierigkeit sein. Außerdem ist die Überwachung des Füllstands vom Wassertank durch einen TOF-Sensor bisher nicht getestet wurden.

**8. Fazit & Zielsetzung**

**Was sollte das Projekt am Ende können:**

Das Projekt sollte am Ende automatisch die Bodenfeuchtigkeit in der Pflanzerde messen, diese auswerte und falls diese zu trocken ist, automatisch eine Pumpe ansteuern, welche die Pflanze dann bewässert. Außerdem werden Messwerte wie Helligkeit, Temperatur, Luftfeuchtigkeit und der Füllstand des angeschlossenen Wassertanks gemessen und über Node-Red verarbeitet werden. Zusätzlich sollten die Messwerte auf einem Node-Red Dashboard dargestellt werden und in einer Datenbank gespeichert werden. Eine Besonderheit ist an diesem Projekt das wenn der Wassertank unter 20% Füllstand fällt, automatisch eine Push-Benachrichtigung, in diesem Fall per E-Mail, ausgelöst wird und den Empfänger darüber informiert das der Füllstand des Wassertanks unter 190mm liegt, was ungefähr 20% des verwendeten Wassertanks entspricht. Damit er genug Zeit hat den Wassertank wieder ausreichend zu füllen.

**Fazit:**

Die Umsetzung des Projekts hat gut funktioniert. Anfangs hat die Analogwertverarbeitung nicht perfekt funktioniert, da die Messwerte stark geschwankt haben, auch hat die Qualität der Spannungsversorgung einen Einfluss auf die Analogwertverarbeitung.