2025

Jerrit Schnaible

BZTG Oldenburg

6.5.2025

IoT-Projekt BloomBuddy

**Ein Bild, das Text, Schrift, Logo, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Inhaltsverzeichnis

[1. Projektbeschreibung 3](#_Toc197452068)

[Kurzbeschreibung: 3](#_Toc197452069)

[Link zum GitHub-Repository: 3](#_Toc197452070)

[https://github.com/jerrit151/300\_Projekt\_BloomBuddy 3](#_Toc197452071)

[2. Anforderungen und Funktionalitäten 3](#_Toc197452072)

[✅ Sensorik 3](#_Toc197452073)

[✅ Aktoren-Steuerung 4](#_Toc197452074)

[✅ Webinterface & Benutzerinteraktion 4](#_Toc197452075)

[3. Benötigte Komponenten 4](#_Toc197452076)

[3.1 Hardware 4](#_Toc197452077)

[3.2 Software & Datenbank 5](#_Toc197452078)

[3.3 Verwendete Software im Detail 5](#_Toc197452079)

[4. Systemarchitektur 5](#_Toc197452080)

[5. Zeitplanung (Meilensteine) 6](#_Toc197452081)

[6. Offene Fragen & Herausforderungen 7](#_Toc197452082)

[7.Technische Besonderheiten 7](#_Toc197452083)

[8. Zusatzfunktionen 7](#_Toc197452084)

[9. Fazit & Zielsetzung 8](#_Toc197452085)

[Was sollte das Projekt am Ende können: 8](#_Toc197452086)

[Persönliches Fazit: 8](#_Toc197452087)

[Bedienungsanleitung – Automatische Pflanzenbewässerung "BloomBuddy" 9](#_Toc197452088)

[1. Systemübersicht 9](#_Toc197452089)

[Funktionen: 9](#_Toc197452090)

[Vorbereitung & Inbetriebnahme 9](#_Toc197452091)

[3. Funktionen im Detail 10](#_Toc197452092)

[3.1 Sensorik & Messungen 10](#_Toc197452093)

[3.2 Automatische Pumpensteuerung 10](#_Toc197452094)

[3.3 Datenverarbeitung & -übertragung 10](#_Toc197452095)

[3.4 Push-Benachrichtigung & Alarm 10](#_Toc197452096)

[4. Bedienung im Alltag 11](#_Toc197452097)

[Automatikbetrieb: 11](#_Toc197452098)

[Manuelle Steuerung über das Node-Red Dashboard 11](#_Toc197452099)

[Statuskontrolle: 11](#_Toc197452100)

[5. Wartung & Tipps 11](#_Toc197452101)

[6. Störungen & Fehlersuche 11](#_Toc197452102)

[7. Beschreibung des Node-RED-Flows 12](#_Toc197452103)

[Node-Red-Flow: 12](#_Toc197452104)

[Node-RED-Dashboard: 12](#_Toc197452105)

[Zusammenfassung 14](#_Toc197452106)

[Installationsanleitung für das BloomBuddy Microcontroller-Projekt 15](#_Toc197452107)

[1. Übersicht der Komponenten 15](#_Toc197452108)

[Benötigte Hardware 15](#_Toc197452109)

[Benötigte Software 15](#_Toc197452110)

[Benötigte Bibliotheken in Micropython 15](#_Toc197452111)

[Wofür die Software benötigt wird 16](#_Toc197452112)

[Aufbau der Datenbank und Beispielmesswerte 17](#_Toc197452113)

[2. Anschluss der Sensoren 18](#_Toc197452114)

[3. Hinweise zur Verkabelung 18](#_Toc197452115)

[4. Inbetriebnahme 19](#_Toc197452116)

[5. Funktionsweise 19](#_Toc197452117)

[6. Sicherheitshinweise 19](#_Toc197452118)

[7. Fehlerbehandlung 19](#_Toc197452119)

## 

## 1. Projektbeschreibung

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

## Link zum GitHub-Repository:

### <https://github.com/jerrit151/300_Projekt_BloomBuddy>

## 2. 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 die aktuelle Helligkeit, Luftfeuchtigkeit, Temperatur, Bodenfeuchtigkeit und den Füllstand des Wassertanks direkt ablesen. Bei einem Füllstand unter 20 % erhält der Benutzer eine Push-Benachrichtigung aufs Handy.

## 

## 3. Benötigte Komponenten

### 3.1 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 |
| **Aktor 2** | 5V 1 Kanal Relaismodul Low Level Trigger | Ansteuern der Pumpe über das Relais |
| **Stromversorgung** | 5V 2A Steckernetzteil | Spannungsversorgung des ESPs, der Sensoren und der Pumpe |
| **Weitere Bauteile** | 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 |

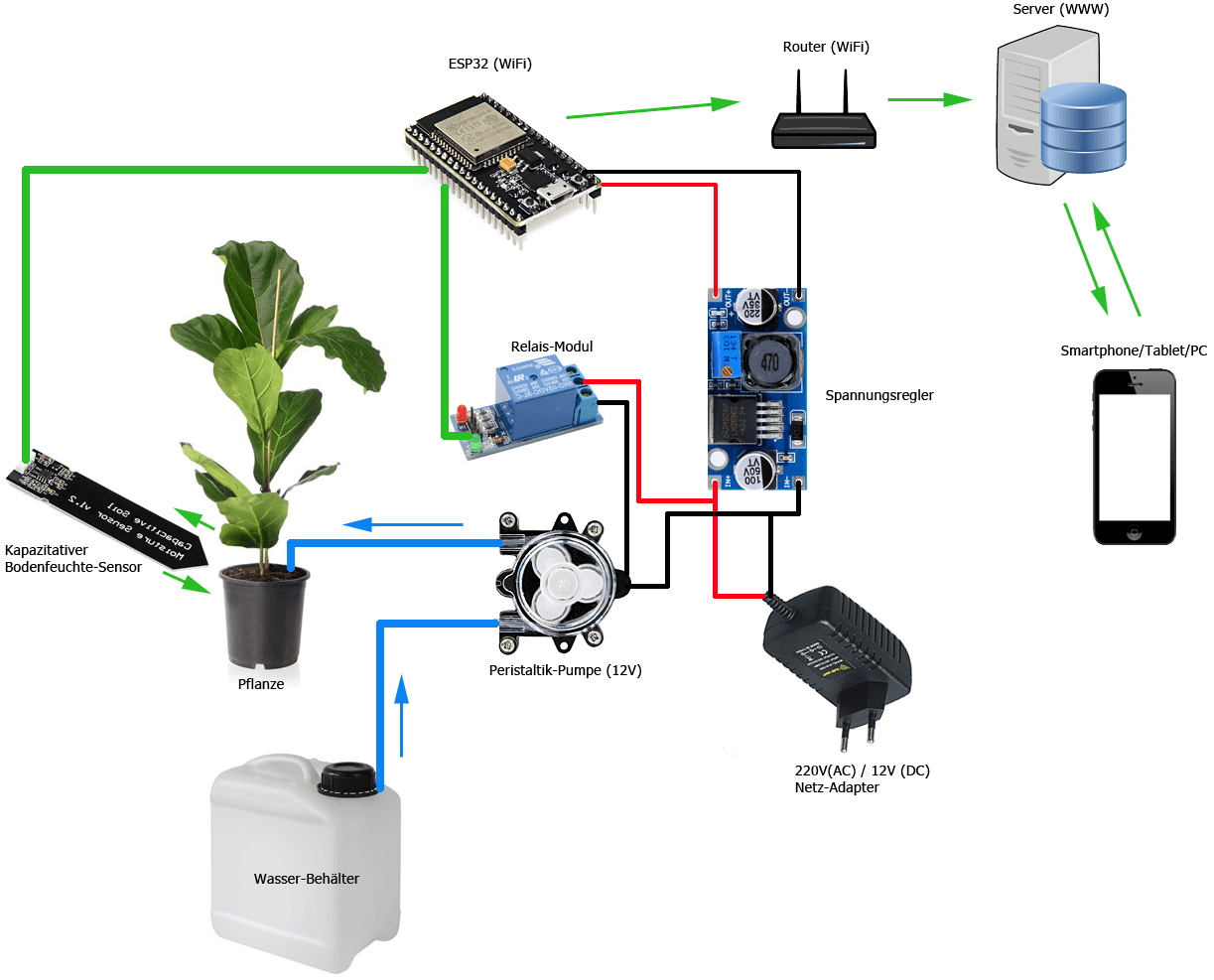
### 3.2 Software & Datenbank

| **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 |

### 3.3 Verwendete Software im Detail

* Node-RED (inkl. node-red-dashboard und node.js)
* Thonny IDE
* HeidiSQL
* Mosquitto MQTT-Broker
* Micropython für Thonny IDE
* MicroPython Firmware für den ESP32
* MariaDB

## 4. Systemarchitektur



Die Abbildung weicht leicht von meinem Projekt ab, ist aber dennoch sehr vergleichbar. Ergänzend habe ich einen TOF-Sensor über den I2C-Bus an den ESP32 angeschlossen, um den Füllstand des Wassertanks zu messen. Zusätzlich kommt ein AHT21-Sensor zum Messen der Luftfeuchtigkeit und Temperatur zum Einsatz, ebenfalls über I2C. Auch der BH1750 zur Helligkeitsmessung ist am I2C-Bus angeschlossen.

## 

## 5. 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, Pumpe) |
| 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 |

## 6. Offene Fragen & Herausforderungen

Batterie- und Stromprobleme: Unzureichende Batterieleistung oder Stromausfälle können das System lahmlegen, z. B. bei Versorgung über eine Powerbank.

Über- oder Unterbewässerung: Falsche Einstellungen, fehlerhafte Sensoren oder unvollständige Daten können zu Über- oder Unterbewässerung führen und der Pflanze schaden.

Programmierung: Die Analogwertverarbeitung kann komplex sein, zudem kann die Darstellung der Daten in Node-RED eine Herausforderung darstellen. Die Überwachung des Füllstands durch den TOF-Sensor wurde bisher nicht getestet.

## 7.Technische Besonderheiten

Die technischen Besonderheiten an meinem Projekt sind zum einen die Push-Benachrichtigung, wenn der Füllstand des Wassertanks unter 20 % (260 mm) fällt. Die Push-Benachrichtigung ist in diesem Projekt eine E-Mail, die über Node-RED an den Empfänger verschickt wird. Eine weitere Besonderheit ist die Analogwertverarbeitung mit dem ESP32. Dazu wurde der Analogausgang des kapazitiven Bodenfeuchtesensors auf einen Analogeingang des ESP32 verdrahtet und die Auswertung des Sensors programmiert, um am Ende einen richtigen Wert zu erhalten. Der umgerechnete Wert wird dann im Programm weiterverarbeitet, um über ein Relais eine Pumpe zur Bewässerung anzusteuern. Zudem kann die Pumpe jederzeit manuell über das Node-RED Dashboard ein- und ausgeschaltet werden.

## 8. Zusatzfunktionen

Zusätzlich zum Automatikbetrieb wurde eine manuelle Steuerung der Pumpe implementiert. Diese erfolgt über einen Schalter im Node-RED-Dashboard und hat jederzeit Vorrang vor dem Automatikbetrieb. Die Pumpe kann beispielsweise auch eingeschaltet werden, wenn die Bodenfeuchtigkeit 100 % beträgt, etwa zum Spülen des Schlauchs. Die manuelle Steuerung wird nicht automatisch abgeschaltet, sondern erst, wenn der Schalter auf dem Dashboard auf „OFF“ gestellt wird. In diesem Fall übernimmt das System sofort wieder den Automatikbetrieb.

## 9. Fazit & Zielsetzung

### Was sollte das Projekt am Ende können:

Das Projekt soll automatisch die Bodenfeuchtigkeit der Pflanzerde messen und auswerten. Ist die Erde zu trocken, wird automatisch eine Pumpe angesteuert, die die Pflanze bewässert. Zusätzlich werden Helligkeit, Temperatur, Luftfeuchtigkeit und der Füllstand des Wassertanks gemessen und über Node-RED verarbeitet. Die Messwerte werden auf einem Node-RED-Dashboard dargestellt und in einer Datenbank gespeichert. Eine Besonderheit ist die Push-Benachrichtigung per E-Mail, wenn der Wassertank unter 20 % Füllstand (260 mm) fällt. So bleibt genug Zeit, den Tank wieder aufzufüllen. Die manuelle Steuerung der Pumpe über das Dashboard hat Vorrang vor dem Automatikbetrieb und bleibt aktiv, bis der Schalter wieder auf „OFF“ steht. Alle genannten Funktionen wurden umgesetzt, um die Funktionalität zu gewährleisten. Auch die Kommunikation per MQTT funktioniert einwandfrei. Die Daten werden in Echtzeit auf dem Dashboard angezeigt und können eine Woche zurückverfolgt werden. Zusätzlich wurde programmiert, dass sich die Pumpe nicht mehr einschalten lässt, wenn der Füllstand des Wassertanks unter 280 mm liegt. Da der Füllstand von oben nach unten gemessen wird, gilt: Je kleiner der Messwert, desto höher der Füllstand.

### Persönliches Fazit:

Die Umsetzung des Projekts hat gut funktioniert. Anfangs gab es Probleme mit der Analogwertverarbeitung, da die Messwerte stark schwankten. Auch die Qualität der Spannungsversorgung hatte Einfluss auf die Messungen. Nach korrekter Einstellung funktionierte alles zuverlässig. Eine Herausforderung war die Befestigung des TOF-Sensors am Wassertank, damit dieser den Füllstand zuverlässig messen kann. Auch der Versand der Push-Benachrichtigung funktionierte anfangs nicht, da falsche Einstellungen im E-Mail-Konto vorlagen. Nachdem diese behoben waren, lief alles einwandfrei. Die Messwerte des TOF-Sensors mussten in Node-RED so verarbeitet werden, dass nur eine E-Mail versendet wird, wenn der Füllstand unter 260 mm liegt. Auch dies wurde gelöst. Insgesamt hat die Umsetzung Spaß gemacht und das Ergebnis ist eine autonome, funktionierende Pflanzenbewässerung für Zuhause, die sich leicht auf weitere Pflanzen erweitern lässt – es sind lediglich kleinere Softwareanpassungen und zusätzliche Bodenfeuchtesensoren nötig.

# Bedienungsanleitung – Automatische Pflanzenbewässerung "BloomBuddy"

Willkommen zur Bedienungsanleitung für das BloomBuddy-Bewässerungssystem! Dieses System dient zur automatisierten Bewässerung Ihrer Pflanze mit Wasser und informiert Sie zuverlässig über wichtige Umgebungsdaten. Diese Anleitung führt Sie Schritt für Schritt durch die Funktionalitäten und den Ablauf der Anlage.

## 1. Systemübersicht

### Funktionen:

* Automatische Bewässerung der Pflanze anhand der Bodenfeuchtigkeit
* Messung und Überwachung von:
  + Bodenfeuchtigkeit
  + Temperatur und Luftfeuchtigkeit
  + Helligkeit
  + Wasserfüllstand (Entfernungssensor)
* Datenübertragung per MQTT an eine zentrale Steuerung/Visualisierung mit Node-RED
* Push-Benachrichtigungen bei niedrigem Tankfüllstand
* Speicherung aller Sensordaten in einer Datenbank
* Manuelle Steuerung der Pumpe über Node-RED-Dashboard

### Vorbereitung & Inbetriebnahme

* Schließen Sie alle Sensoren und das Relais gemäß der Hardwareliste und des Schaltplans an die vorgesehenen Pins des ESP32 an.
* Stellen Sie sicher, dass der WLAN-Zugang (SSID & Passwort) korrekt im Code hinterlegt ist.
* Schalten Sie das System ein, indem Sie den ESP32 und die Pumpe mit Strom versorgen.
* Verlegen Sie die Schläuche der Pumpe so, dass ein Schlauch vom Wassertank zur Pumpe und der andere von der Pumpe zur Pflanze führt. Achten Sie dabei unbedingt auf die Flussrichtung der Pumpe, damit das Wasser korrekt gepumpt wird.
* Stecken Sie den kapazitiven Bodenfeuchtesensor bis zur Markierung auf dem Sensor in die Pflanzerde. Nur so kann eine zuverlässige Messung der Bodenfeuchte erfolgen.
* Platzieren Sie das Schlauchende der Pumpe mit ausreichend Abstand zum Sensor in der Erde, damit das Wasser nicht direkt auf den Sensor trifft. So werden Messfehler durch direkte Benetzung des Sensors vermieden.
* Und schon ist ihr BloomBuddy System erfolgreich installiert und nimmt seine Arbeit auf.

## 3. Funktionen im Detail

### 3.1 Sensorik & Messungen

* Bodenfeuchtigkeit:  
  Der Sensor misst den Feuchtegrad im Boden. Die Werte werden zwischen "sehr trocken" (z. B. 3070) und "sehr nass" (z. B. 1700) kalibriert.
* Temperatur und Luftfeuchtigkeit:  
  Gemessen mit dem AHT21 und als Mittelwert aus mehreren Messungen ermittelt.
* Helligkeit:  
  Gemessen mit dem BH1750; Mittelwertbildung zur Genauigkeit.
* Wasserfüllstand:  
  Der VL53L0X misst die Entfernung zur Wasseroberfläche und erkennt so den Tankfüllstand.

### 3.2 Automatische Pumpensteuerung

* Ist der Boden zu trocken (≤ 40%), wird die Pumpe automatisch für 25 Sekunden eingeschaltet, um die Pflanze zu bewässern.
* Nach Ablauf der 25 Sekunden schaltet sich die Pumpe automatisch wieder ab.
* Ist die Bodenfeuchtigkeit hoch genug (> 40%), bleibt die Pumpe aus.
* Die Pumpe lässt sich nicht mehr einschalten wenn der Füllstand des Wassertanks > 280mm ist.

### 3.3 Datenverarbeitung & -übertragung

* Jeder Messzyklus berechnet gemittelte Sensordaten und entfernt Ausreißer zur Verbesserung der Genauigkeit.
* Die Ergebnisse werden als JSON-Objekt verpackt und über MQTT an das Netzwerk (Node-RED) gesendet.
* In Node-RED werden die Daten empfangen, visualisiert und in eine Datenbank (MariaDB) gespeichert

### 3.4 Push-Benachrichtigung & Alarm

* Sinkt der Füllstand des Wassertanks unter 20% (260mm), wird eine Push-Benachrichtigung ausgelöst (E-Mail).

## 4. Bedienung im Alltag

### Automatikbetrieb:

* Nach Anschluss startet das System automatisch.
* Die Sensoren messen regelmäßig alle Werte, regeln die Bewässerung und senden die Umgebungsinformationen an das Netzwerk.
* Sie müssen lediglich sicherstellen, dass der Wassertank regelmäßig nachgefüllt wird.

### Manuelle Steuerung über das Node-Red Dashboard

Die Pumpe kann jederzeit manuell über das Node-Red Dashboard gesteuert werden. Die manuelle Steuerung hat Vorrang vor dem Automatikbetrieb:

* Ist der Schalter im Dashboard auf „ON“ gestellt, läuft die Pumpe dauerhaft und der Automatikbetrieb ist deaktiviert. Dies ist beispielsweise nützlich, um den Schlauch zu spülen oder die Pumpe gezielt zu testen.
* Ist der Schalter auf „OFF“ gestellt, übernimmt das System wieder die automatische Steuerung der Bewässerung entsprechend der gemessenen Bodenfeuchtigkeit.
* Solange der Schalter auf „ON“ bleibt, bleibt die Anlage im Handbetrieb und geht nicht in den Automatikmodus über.

Die Steuerung erfolgt über einen Schalter im Node-RED Dashboard, der per MQTT mit dem System verbunden ist.

### Statuskontrolle:

* Die aktuellen Werte für Bodenfeuchte, Temperatur, Luftfeuchtigkeit, Helligkeit und Tankfüllstand können Sie jederzeit über die Benutzeroberfläche in Node-RED ablesen.
* Bei Problemen mit dem WLAN oder MQTT informiert das System über die Konsole.

## 5. Wartung & Tipps

* Sensor-Kalibrierung: Die Referenzwerte für „trocken“ und „nass“ sollten einmalig für Ihre Pflanze ermittelt und im Code angepasst werden.
* Wassertank: Halten Sie den Tank stets gefüllt, damit das System zuverlässig arbeiten kann.

## 6. Störungen & Fehlersuche

Meldet das System „WLAN-Verbindung fehlgeschlagen!“, prüfen Sie Zugangsdaten und Router.

Reagiert die Pumpe nicht, prüfen Sie das Relais oder die Verkabelung.

Bei falschen Sensordaten: Sensoren auf korrekten Sitz und Verschmutzung prüfen.

## 7. Beschreibung des Node-RED-Flows

Übersicht der Funktionen im Node-RED-Flow für das BloomBuddy-Projekt

### Node-Red-Flow:

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

### 

### Node-RED-Dashboard:

Ein Bild, das Text, Screenshot, Multimedia-Software, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

MQTT-Empfang der Sensordaten:

Node: mqtt in („Sensorwerte“)

Funktion: Empfängt die vom ESP32 gesendeten Sensordaten (Füllstand, Temperatur, Luftfeuchtigkeit, Helligkeit, Bodenfeuchtigkeit) über das Topic Zuhause/Wohnung/BloomBuddy.

Aufbereitung und Visualisierung der Einzelwerte:

Function-Nodes extrahieren jeweils einen Wert aus der empfangenen Nachricht.

UI-Gauges & Charts zeigen jeden Wert als Messinstrument (Gauge) und im Verlauf (Chart) an.

Warnung bei niedrigem Füllstand:

Switch-Node prüft, ob der Füllstand unter 260 mm (ca. 20 %) liegt.

Function-Node erstellt eine Warnmeldung, wenn der Wert unterschritten wird.

E-Mail-Node versendet diese Warnung als E-Mail an eine festgelegte Adresse.

Delay-Node begrenzt die E-Mail-Frequenz, um Spam zu vermeiden.

Speicherung der Sensordaten in der Datenbank:

Function-Node bereitet die Sensordaten und einen Zeitstempel für den SQL-Insert vor.

MySQL-Node schreibt die Daten in die Tabelle sensorwerte\_wohnung der Datenbank.

Manuelle Pumpensteuerung:

UI-Switch ermöglicht das manuelle Ein- und Ausschalten der Pumpe über das Dashboard.

MQTT-Out-Node sendet den Schaltbefehl als MQTT-Nachricht an das ESP32-Programm.

Debugging und Überwachung:

Debug-Nodes geben Nachrichten und Werte zur Kontrolle in der Node-RED-Seitenleiste aus.

## Zusammenfassung

Das BloomBuddy-System arbeitet weitgehend autark mit automatischer Steuerung und Übertragung aller relevanten Daten. Ihr Eingriff beschränkt sich auf das Nachfüllen des Tanks und die Kontrolle der Sensorfunktion über Node-RED – den Rest erledigt die Anlage selbstständig. Zusätzlich können Sie die Pumpe jederzeit manuell über das Node-RED-Dashboard steuern. Die manuelle Steuerung hat dabei Vorrang vor dem Automatikbetrieb: Ist der Schalter im Dashboard auf „ON“ gestellt, läuft die Pumpe dauerhaft und der Automatikmodus ist deaktiviert – dies kann beispielsweise zum Spülen des Schlauchs genutzt werden. Erst wenn der Schalter wieder auf „OFF“ steht, übernimmt das System wieder automatisch die Bewässerung entsprechend der gemessenen Bodenfeuchtigkeit.

# Installationsanleitung für das BloomBuddy Microcontroller-Projekt

Diese Anleitung erklärt Schritt für Schritt, wie die Hardware korrekt angeschlossen und das System in Betrieb genommen wird.

## 1. Übersicht der Komponenten

### Benötigte Hardware

* ESP32 S3
* Capacitive Soil Moisture Sensor V2.0.0 (analog)
* AHT21 (Temperatur & Luftfeuchtigkeit, I2C)
* BH1750 (Helligkeit, I2C)
* VL53L0X (ToF Entfernungssensor, I2C)
* 5V Relaismodul
* AM325 Mini Pumpe
* 5V 2A Steckernetzteil
* Diverse Jumper-Kabel

### Benötigte Software

* Node-RED (inkl. node-red-dashboard und node.js)
* Thonny IDE
* HeidiSQL
* Mosquitto MQTT-Broker
* Micropython für Thonny IDE
* MicroPython Firmware für den ESP32
* MariaDB

### Benötigte Bibliotheken in Micropython

* aht.py
* bh1750.py
* VL53L0X.py

### Wofür die Software benötigt wird

1. Node-RED

Zur Visualisierung, Steuerung und Automatisierung der Sensordaten und Pumpensteuerung. Node-RED dient als Flow-basierte Entwicklungsumgebung, um MQTT-Nachrichten zu empfangen, Daten zu verarbeiten, in eine Datenbank zu schreiben und Benachrichtigungen zu generieren.

2. Thonny IDE

Zum Schreiben, Hochladen und Debuggen von MicroPython-Code auf dem ESP32. Die Thonny IDE ist besonders für Einsteiger geeignet und bietet eine einfache Möglichkeit, MicroPython-Code direkt auf das Board zu übertragen und auszuführen.

3. HeidiSQL

Zur Verwaltung und Überwachung der MariaDB- oder MySQL-Datenbank, in die Node-RED die Sensordaten schreibt. HeidiSQL bietet eine grafische Oberfläche, um Tabellen zu erstellen, Abfragen auszuführen und die Datenbankstruktur zu verwalten.

4. MQTT-Broker (z. B. Mosquitto)

Der MQTT-Broker ist das Bindeglied zwischen dem ESP32 (der die Sensordaten publiziert) und Node-RED (das die Daten abonniert und weiterverarbeitet). Mosquitto ist ein weit verbreiteter, kostenloser MQTT-Broker, der sich einfach unter Windows installieren lässt.

5. Python (inkl. MicroPython für ESP32)

Python wird auf dem Laptop für die Thonny IDE benötigt. MicroPython muss auf dem ESP32 installiert werden, damit der bereitgestellte Code ausgeführt werden kann.

6. Treiber für den ESP32

Damit der Windows-Laptop den ESP32 erkennt, werden die passenden USB-Treiber (meist CP210x oder CH340, je nach Board) benötigt.

7. MariaDB

Die eigentliche Datenbank, in die Node-RED die Sensordaten schreibt.

### Aufbau der Datenbank und Beispielmesswerte

Ein Bild, das Text, Screenshot, Software, Display enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Display, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

## 2. Anschluss der Sensoren

a) Sensoren (AHT21, BH1750, VL53L0X)

* VCC aller Sensoren an +5V (ESP32 5V-Pin (auf Breadboard))
* GND aller Sensoren an GND (ESP32 GND-Pin (auf Breadboard))
* SDA aller Sensoren an GPIO 5
* SCL aller Sensoren an GPIO 4
* Als Busabschluss wurden für SCL & SDA jeweils 4,7kΩ Widerstände auf+5V verdrahtet

b) Bodenfeuchtesensor (kapazitiver Bodenfeuchtesensor V2.0.0)

* VCC an +5V
* GND an GND
* Signal an GPIO 15

c) Relaismodul (5V, 1-Kanal)

* DC+ (VCC) an +5V (ESP32 5V-Pin)
* DC- (GND) an GND (ESP32 GND-Pin)
* IN1 an GPIO 8 (ESP32, wie im Code: Pin(8))
* Relais schaltet bei High-Signal, siehe Jumper

d) Pumpe am Relaisausgang

* COM am Relais an +5V des externen Netzteils
* NO (Normally Open) am Relais an Plusleitung der Pumpe
* NC bleibt frei
* Minusleitung der Pumpe an GND des externen Netzteils

## 3. Hinweise zur Verkabelung

* Alle GND-Leitungen (ESP32, Sensoren, Relais, Pumpe) müssen verbunden sein.
* Achtung: Arbeite immer spannungsfrei.
* 5V-Relais: Der Steuerstrom für das Relais kommt vom ESP32. Die Last (Pumpe) wird über die Relaiskontakte geschaltet.

## 

## 4. Inbetriebnahme

1. **Verkabelung kontrollieren** (siehe oben).
2. **Code auf den ESP32 flashen** (mit Thonny) .
3. **ESP32 mit Strom versorgen** (über USB Typ C per Powerbank oder ähnliches).
4. **Pumpe mit Strom versorgen** (über 5V 2A Steckernetzeil)
5. **MQTT-Server bereitstellen** (wie im Code, z.B. Mosquitto auf 192.168.33.79).
6. **Node-Red verwenden**, um Nachrichten zu senden:
   * {"Schalter1":"ON"} schaltet die Pumpe ein (Relais HIGH)
   * {"Schalter1":"OFF"} schaltet die Pumpe aus (Relais LOW)

Wenn alle diese Punkte abgearbeitet sind, sollte die Anlage sofort mit der Arbeit beginnen. Die Steuerung über Node-RED ist optional; die Anlage funktioniert auch ohne manuelle Steuerung der Pumpe.

## 5. Funktionsweise

* **Automatikbetrieb:** Die Pumpe schaltet automatisch ein, wenn die Bodenfeuchtigkeit ≤ 40 % beträgt, der Wassertank ausreichend gefüllt ist (< 280mm) und kein Handbetrieb aktiv ist. Sie läuft dann 25 Sekunden. Danach erfolgt eine erneute Messung
* **Manueller Betrieb:** Über MQTT kann die Pumpe jederzeit ein- und ausgeschaltet werden. Der Handbetrieb hat Vorrang.
* **Sensorwerte** werden regelmäßig gemessen und per MQTT publiziert.

## 6. Sicherheitshinweise

* **Kein Netzstrom:** Das System ist für 5V-Komponenten ausgelegt.
* **Relaismodul:** Achte darauf, dass der Steuerstrom des Relais den ESP32 nicht überlastet.
* **Wasser & Elektronik:** Sorge für Spritzwasserschutz und sichere Montage.

## 7. Fehlerbehandlung

* **Pumpe schaltet nicht:** Prüfe, ob das Relais beim Schalten klickt. Miss die Spannung am Relaisausgang.
* **Sensorwerte fehlen:** Prüfe I2C-Verbindungen (SDA/SCL) und Versorgungsspannung.
* **MQTT funktioniert nicht:** Prüfe WLAN, Broker-IP und MQTT-Topic.
* **ESP32 startet nicht:** Prüfe, ob die 5V-Versorgung stabil ist.