

[Dieses Bild wurde von Künstlicher Intelligenz generiert](#)

## Glassbox

**Thema:** Glassbox Warnungen für Stickigkeit und Temperatur

### Dokumentinformationen

Dateiname: Projekt\_Dokumentationen\_Glassbox  
Speicherdatum: 13.12.2024

### Autoreninformationen

Autor: Nepomuk Crhonek, Inaam Ahmed, Eleonora Vynogradova

**Inhaltsverzeichnis**

1	Einleitung .....	3
1.1	Sinn und Zweck .....	3
1.2	Ziel.....	3
1.3	Produkt Beschreibung .....	3
1.4	Gruppen Aufteilung.....	3
2	Checkliste und Steckplan .....	4
2.1	Steckplan Bilder .....	6
2.2	Sicherheitsvorkehrungen.....	7
3	Code.....	8
3.1	Überblick .....	8
3.2	Hauptfunktionen.....	8
3.3	Hardware.....	8
3.4	Einstellungen .....	8
3.5	LED-Anzeigen.....	8
3.6	MQTT-Themen .....	9
3.7	Bildschirmanzeige.....	9
3.8	Programmablauf .....	9
3.9	Fehlerbehebung.....	9
4	Node Red .....	11
4.1	Flowcharts.....	11
4.2	UI.....	12
5	Testing .....	13
5.1	Testprotokoll .....	13
6	Schlussfolgerungen .....	14
6.1	Reflexion .....	14

## 1 Einleitung

### 1.1 Sinn und Zweck

Das vorliegende Dokument ist die Dokumentation von unserem Glassbox Projekt vom üK 216. Hier ist alles vorhanden von den Hardwareteilen bis zur Software.

### 1.2 Ziel

Das Ziel unseres Projekts ist, dass es die Temperatur und Feuchtigkeit von der Glassbox berechnet. Dazu haben wir verschiedene Features gemacht. Wie zum Beispiel gibt es eine Ampel, die anzeigt, ob alles gut ist oder nicht. Wenn die Ampel Rot leuchtet, soll unser Produkt einen Ton machen. Dazu soll es auf Node-RED bei UI Grafisch und mit einem Wert dargestellt werden.

### 1.3 Produkt Beschreibung

Dieses Produkt ist das beste Produkt von allen. Es heisst Baum. Wenn du in der Glassbox bist, kannst du unser Produkt benutzen damit es die Temperatur und Feuchtigkeit misst. Unser Produkt sagt dir, wenn du am besten die Glassbox verlassen sollst. Wenn es zu warm und feucht wird tut die Ampel rot leuchten und einen Alarm ton machen. Dazu hast du eine Message auf dem Display. Du kannst deine Einstellungen in UI beliebig auswählen. Auch noch kannst du auf dem Monitor sehen wie viel Grad es ist, und wie viel Prozent es feucht ist. Benutz unser Produkt und du wirst eine gute Umgebung im Glassbox haben.

Wichtig zu erwähnen sind noch die Sachen, die unser Produkt nicht kann:

- Es kann keine Musik abspielen lassen
- Es kann nicht mit Bluetooth verbunden werden
- Es kann kein CO<sub>2</sub> messen

### 1.4 Gruppen Aufteilung

**Nepomuk:** Zuständig für Coding, MQTT, Node Red und Dokumentation über den Code





**Inaam:** Zuständig für den Steckplan, Dokumentation vom Steckplan, Sicherheitsvorkehrungen und die Schlussfolgerung

**Eleonora:** Zuständig für die Dokumentation Node Red, Testing, Grafiken und der Einleitung





## 2 Checkliste und Steckplan

Wir haben uns entschieden, ein Temperatur-Feuchtigkeit Sensor zu benutzen. Um das richtig zu stecken, haben wir recherchiert, wie man die Hardware einstecken muss. Für den Temperatur-Feuchtigkeit-Sensor haben wir die Dokumentation 00 verwendet, da dort der Steckplan war. Bei der Ampel konnten wir unser Vorwissen verwenden, da wir es schon mal eingesteckt haben. Doch bei dem Buzzer mussten wir recherchieren, da wir noch nie mit dem gearbeitet haben.

Beim Stecken hatten wir einen Fehler, da wir nicht einberechnet haben, dass der Strom nicht ganz durchgeht, sondern in der Mitte stoppt. Aber durch kleine umsteckenden ging es. Dadurch können wir jetzt mit dem Code anfangen.

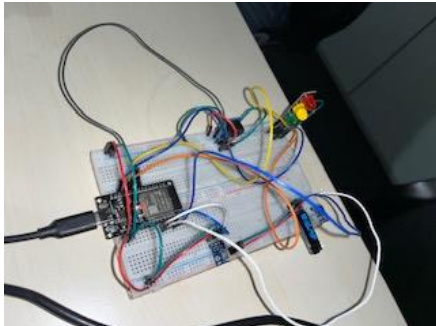
Grafik	Hardware	Menge
	Ampel	x1
	OLED-Display	x1
	Feuchtigkeit und Temperatur Sensor	x1
	Buzzer	x1

## Glassbox Warnungen für Stickigkeit und Temperatur

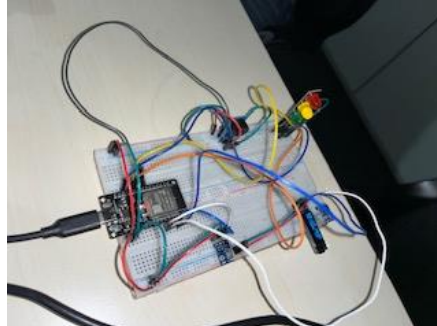
	ESP 32	x1
	Männliche Kabel	x20
	USB-C Kabel	x1
	Steckplatine	x1

## 2.1 Steckplan Bilder

Alter Steckplan:



Neuer Steckplan:



Hardware	Pin	GPIO-Pins
OLED-Display	VCC	3v3
OLED-Display	GND	GND
OLED-Display	SCK	D22
OLED-Display	SDA	D21
AHT10	VIN	3v3
AHT10	GND	GND
AHT10	SCL	D22
AHT10	SDA	D21
Ampel	GND	GND
Ampel	R	D27
Ampel	Y	D14
Ampel	G	D13
Buzzer	GND	GND
Buzzer	I/O	D25
Buzzer	VCC	3v3



## 2.2 Sicherheitsvorkehrungen

Wir verwenden den MQTT-Server der Noser Young. Man hat Zugriff auf den Server durch die IP-Adresse. Es ist jedoch sicherer, ein Passwort zu erstellen. So kann man nicht nur mit der IP-Adresse Zugriff auf MQTT erhalten.

Bei Node-Red können Benutzer nur mit dem Benutzernamen «admin» und dem Passwort «KxtWffSx» Zugriff erhalten. Man könnte die Sicherheit jedoch deutlich erhöhen, indem man den Benutzernamen und das Passwort komplexer und länger gestaltet. Andernfalls könnten Personen mit Bösen Absichten diese leicht erraten.

Zusätzlich kann man Node-Red durch ein Zertifikat schützen. Mit einem Zertifikat wird die Kommunikation verschlüsselt, sodass niemand sie abhören kann. Zudem wird verhindert, dass Personen ohne Berechtigung auf Node-Red zugreifen.

### 3 Code

#### 3.1 Überblick

Dieses Programm steuert einen ESP32 Mikrocontroller, der Temperatur und Luftfeuchtigkeit in einem Raum überwacht. Es zeigt die Werte auf einem kleinen Bildschirm an und sendet sie über WLAN an einen MQTT-Server.

#### 3.2 Hauptfunktionen

- Misst Temperatur und Luftfeuchtigkeit
- Zeigt die Messwerte auf einem OLED-Bildschirm an
- Zeigt den Status mit drei LEDs an (rot, gelb, grün)
- Gibt einen Warnton aus, wenn die Werte zu schlecht sind
- Sendet die Messwerte über MQTT
- Kann Einstellungen über MQTT empfangen

#### 3.3 Hardware

- ESP32 Mikrocontroller
- AHTX0 Temperatur- und Feuchtigkeitssensor
- SSD1306 OLED-Bildschirm
- Drei LEDs (rot, gelb, grün)
- Summer für Warnton

#### 3.4 Einstellungen

##### 3.4.1 Grundeinstellungen

- Temperatur: mindestens 27°C, höchstens 50°C
- Luftfeuchtigkeit: mindestens 23%, höchstens 50%

##### 3.4.2 WLAN und MQTT

- Verbindet sich mit "GuestWLANPortal"
- Sendet Daten an MQTT-Server (IP: 10.10.2.127)
- Gerätenamen: "ESP32-ukteam-1-Glassbox"

##### 3.4.3 Update

- Mit [Web Flasher](#) haben wir den ESP32 auf den neusten Stand gebracht

#### 3.5 LED-Anzeigen

- Grünes Licht: Alle Werte sind gut
- Gelbes Licht: Entweder Temperatur oder Luftfeuchtigkeit ist außerhalb der Grenzen
- Rotes Licht: Beide Werte sind außerhalb der Grenzen (mit Warnton)



### 3.6 MQTT-Themen

#### 3.6.1 Messwerte senden

- Temperatur: "zuerich/glasbox/oliv/temp"
- Luftfeuchtigkeit: "zuerich/glasbox/oliv/humidity"

#### 3.6.2 Einstellungen empfangen

- Maximale Temperatur: "zuerich/glasbox/oliv/settings/tempMax"
- Minimale Temperatur: "zuerich/glasbox/oliv/settings/tempMin"
- Maximale Luftfeuchtigkeit: "zuerich/glasbox/oliv/settings/humidityMax"
- Minimale Luftfeuchtigkeit: "zuerich/glasbox/oliv/settings/humidityMin"

### 3.7 Bildschirmanzeige

Der OLED-Bildschirm zeigt:

- Aktuelle Temperatur in °C
- Aktuelle Luftfeuchtigkeit in %
- Aktueller Status ("great", "leave soon" oder "leave now")

### 3.8 Programmablauf

#### 1. Start:

- Startet alle Sensoren und den Bildschirm
- Verbindet sich mit WLAN
- Verbindet sich mit MQTT-Server

#### 2. Hauptschleife:

- Prüft WLAN- und MQTT-Verbindung
- Liest Temperatur und Luftfeuchtigkeit
- Aktualisiert LEDs und Bildschirm
- Sendet Daten an MQTT-Server
- Wartet 200 Millisekunden
- Wiederholt die Schleife

### 3.9 Fehlerbehebung

- Bei WLAN-Verlust: Automatische Wiederverbindung
- Bei MQTT-Verlust: Automatische Wiederverbindung
- Bei Sensorfehler: Programm stoppt mit Fehlermeldung
- Bei Bildschirmfehler: Programm stoppt mit Fehlermeldung

Weitere Dokumentation zum Code sind als Kommentare im Code vermerkt.

### 3.10 Code Quellen

Den meisten Teil des Codes haben wir von den offiziellen Kursunterlagen entnommen. Bei der Verbindung von dem ESP32 und dem Buzzer haben wir diese [Webseite](#) verwendet. Bei der subscribe Funktion haben wir diese [Webseite](#) verwendet.

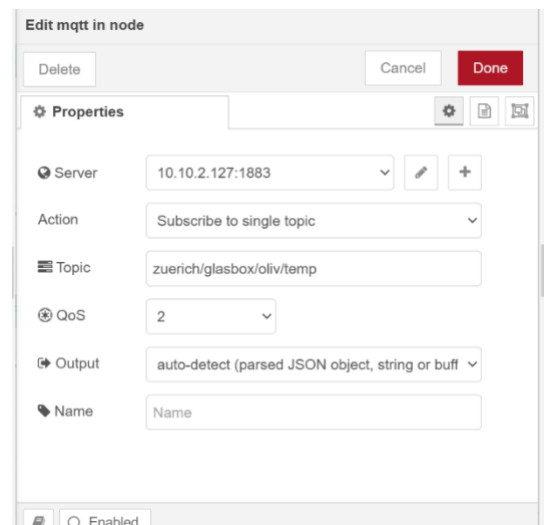
## 4 Node-RED

### 4.1 Flowcharts

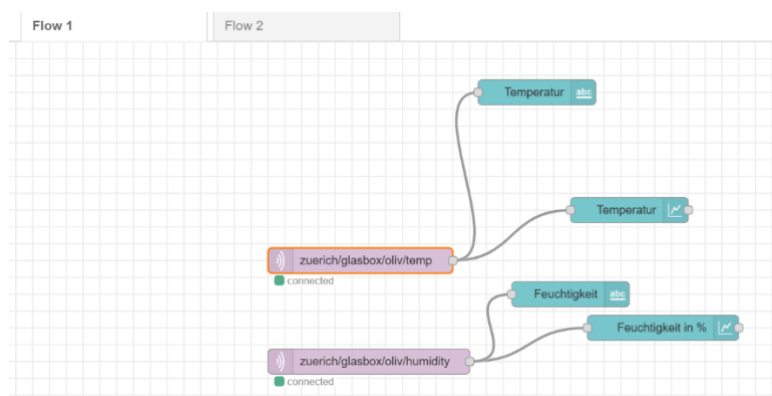
Zuerst mussten wir ins Node-RED einloggen, dafür haben wir ein Passwort und einen Benutzernamen erhalten. Danach haben wir mit dem MQTT-Server verbunden. Um zu verbinden, mussten wir beim Server in Properties die IP-Adresse **10.10.2.127** und noch den Port **1883** dazu. Danach haben wir einen Topic dazu geschrieben der heisst **zuerich/glasbox/oliv/temp**.

Danach haben wir zu den Nodes Verbindungen gemacht damit die Daten als Graph und Nummer dargestellt werden. Diese mussten wir noch zu Gruppen einordnen.

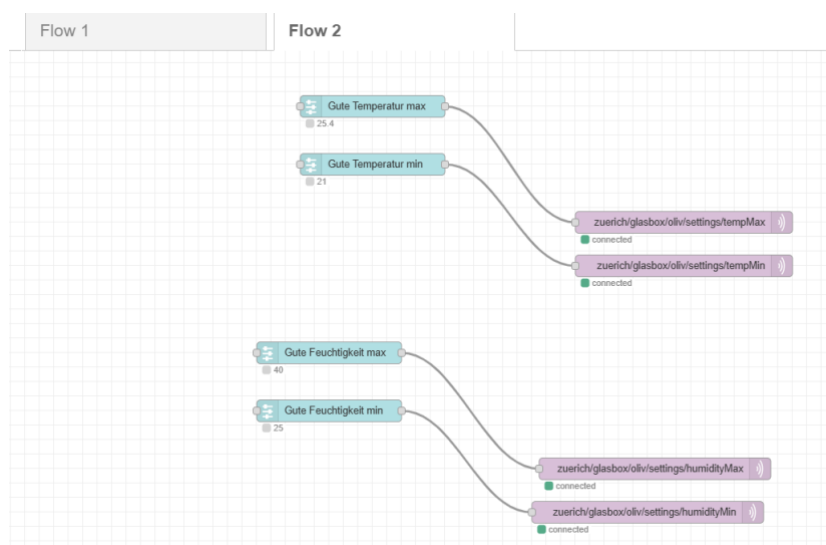
Nach dem erstem Flow haben wir ein zweites dazu gemacht. Dort haben wir die Settings erstellt damit man die Einstellungen ändern kann.



#### Flow 1



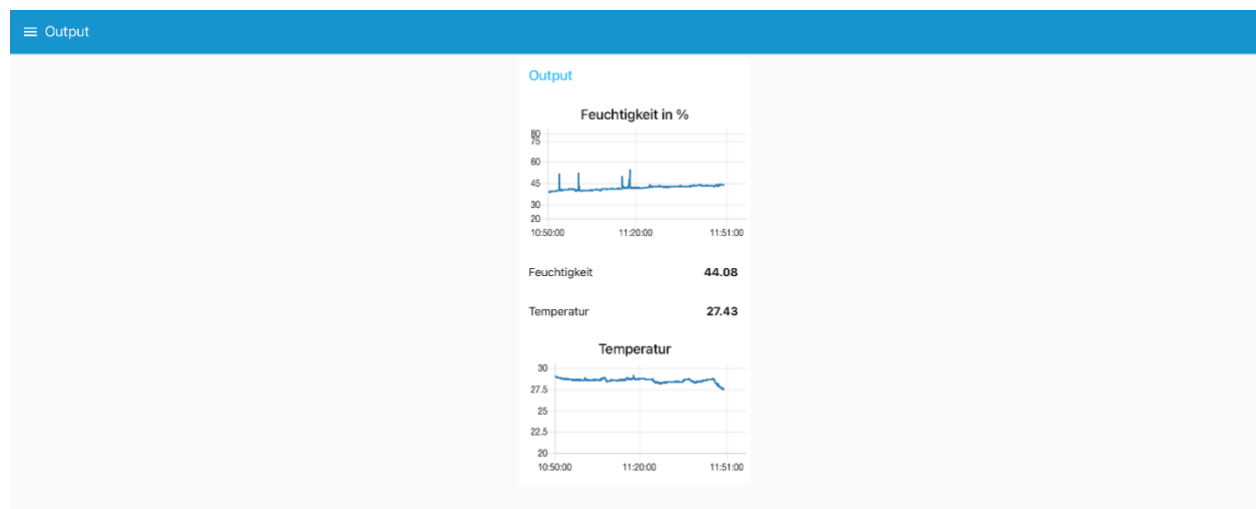
#### Flow 2



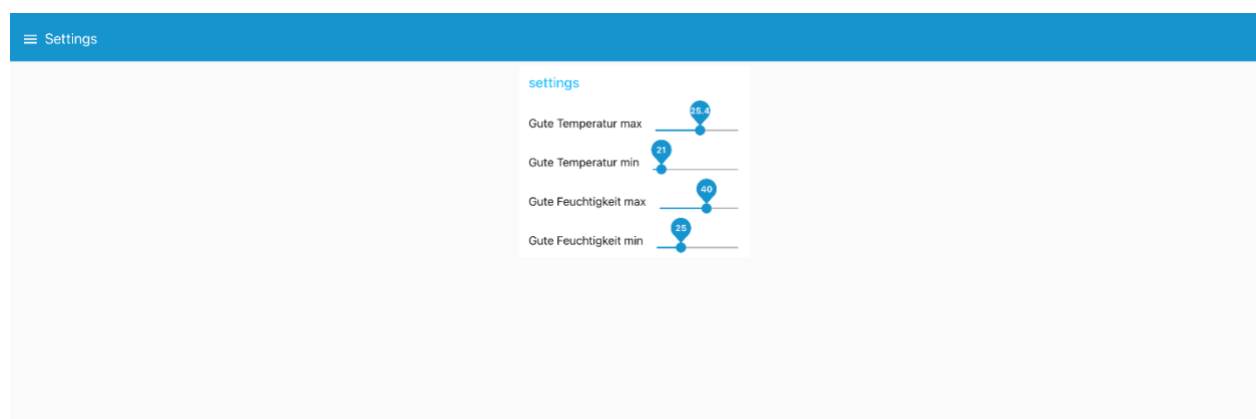
## 4.2 UI

Nachdem wir mit dem ganzen fertig gewesen sind, haben wir den UI geöffnet zum Sehen, ob alles so aussieht wie wir es möchten. Es sah dann gut aus. Wir haben am Schluss die Unterteilung zwischen Output und Settings gehabt. Beim Output sieht man die Graphen und die genaue Anzahl der Daten, währenddessen kann man bei den Settings die Einstellungen so anpassen wie man will.

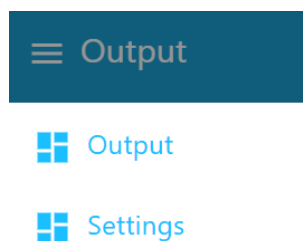
### Output



### Settings



### Balken



## 5 Testing

### 5.1 Testprotokoll

Nr.	Test	Eingabe	Erwartet	Ergebnis	Status	Tester
1.	OLED-Display zeigt die Temperatur an	Der Temperatur Sensor soll vorhanden sein	Temperatur soll angezeigt werden	Temperatur wird angezeigt	✓	Nepomuk
2.	OLED-Display zeigt die Feuchtigkeit an	Der Temperatur Sensor soll vorhanden sein	Feuchtigkeit soll angezeigt werden	Feuchtigkeit wird angezeigt	✓	Nepomuk
3.	OLED-Display zeigt den Status an	Display ist vorhanden	Der entsprechende Status soll angezeigt werden	Der richtige Status wird angezeigt	✓	Nepomuk
4.	Ampel tut leuchten	Ampel ist vorhanden	Die Ampel soll leuchten	Die Ampel leuchtet	✓	Nepomuk
5.	Ampel tut entsprechend leuchten	Der Temperatur Sensor soll vorhanden sein	Die Ampel tut entsprechend zu den Daten leuchten	Die Ampel leuchtet wie es soll	✓	Eleonora
6.	Beim roten Licht gibt es ein Ton	Temperatur Sensor berühren	Während das Licht rot ist, gibt es ein Ton	Es gibt ein Ton während es rot ist	✓	Inaam
7.	Bei Node-RED werden die Daten als Graph angezeigt	Graph ist im Flow vorhanden	Die Daten sollen als Graph dargestellt werden	Die Daten werden als Graph dargestellt	✓	Eleonora
8.	Bei Node-RED werden die Daten als Nummer angezeigt	Text ist im Flow vorhanden	Die Daten sollen als Nummer dargestellt werden	Die Daten werden als Nummer dargestellt	✓	Inaam
9.	Es gibt eine Auswahl zwischen Settings und Output	Oben links anklicken und beide Varianten anklicken können	Man soll auf der linken Seite oben eine Auswahl sehen	Es gibt links oben eine Auswahl	✓	Eleonora
10.	Bei Settings kann man die Einstellungen ändern	Einstellungen ändern	Man soll die Einstellungen ändern können	Man kann es ändern	✓	Eleonora

**Glassbox Warnungen für Stickigkeit und Temperatur**


---

11.	Die entsprechenden Einstellungen werden vom ESP32 übernommen	Einstellungen ändern	Die Einstellungen sollen entsprechend übernommen werden	Die Einstellung wird übernommen	✓	Inaam
12.	Alles tut zusammenarbeiten: Teil 1	Der Temperatur Sensor soll vorhanden sein	Bei über 26 C° und über 40% humidity soll die Ampel rot werden & ton ausgegeben werden	Die Ampel wird rot, und es wird ein Ton ausgegeben	✓	Eleonora
13.	Alles tut zusammenarbeiten: Teil 2	Der Temperatur Sensor soll vorhanden sein	Bei OLED-Status auf Verlassen anzeigen, daten auf Anzeigen und Graph + Nummer bei Node-RED angezeigt werden	Der Status wird als Verlassen angezeigt und die Daten werden als Nummer + Graph angezeigt	✓	Eleonora

## 6 Schlussfolgerungen

### 6.1 Reflexion

Die Teamarbeit war sehr gut, da wir alle miteinander gearbeitet haben. Wir haben uns alle gegenseitig geholfen und uns Informationen über Hardwareteile, MQTT wie auch Node Red ausgetauscht. Dies half uns bei der Durchführung des Projektes. Durch das konnten wir unser Projekt schon am zweiten Tag abschliessen.

Bei der technischen Umsetzung lief fast alles gut. Am Anfang hatten wir Probleme mit dem Steckplan, aber wir haben das schnell gefixt. Nachdem wir den Steckplan korrigiert hatten, hat alles super funktioniert - der Temperatur-Sensor, die Ampel und der Buzzer hat alles gut zusammen funktioniert.

Unser Testing hat gezeigt, dass alles läuft. Das haben wir auch in der Testliste aufgeschrieben.

Was wir besser machen könnten: Die Sicherheit könnte stärker sein. Wie wir geschrieben haben, sollten die Passwörter für Node-RED schwieriger sein und wir könnten ein Zertifikat benutzen, damit die Verbindung sicherer ist.

Am Ende haben wir alles erreicht, was wir wollten. Wir haben viel über Teamarbeit und IoT-Sachen gelernt. Weil wir früh fertig waren, konnten wir auch alles gut testen und verbessern.