

Dokumentation Wetterstation

Projektbezeichnung	Projektarbeit Mikrocomputertechnik
Projektleiter	Noah Canadea / Milan Bursac
Erstellt am	20.12.2021
Letzte Änderung am	10.01.2022
Status	abgeschlossen
Aktuelle Version	1.0

Änderungsverlauf

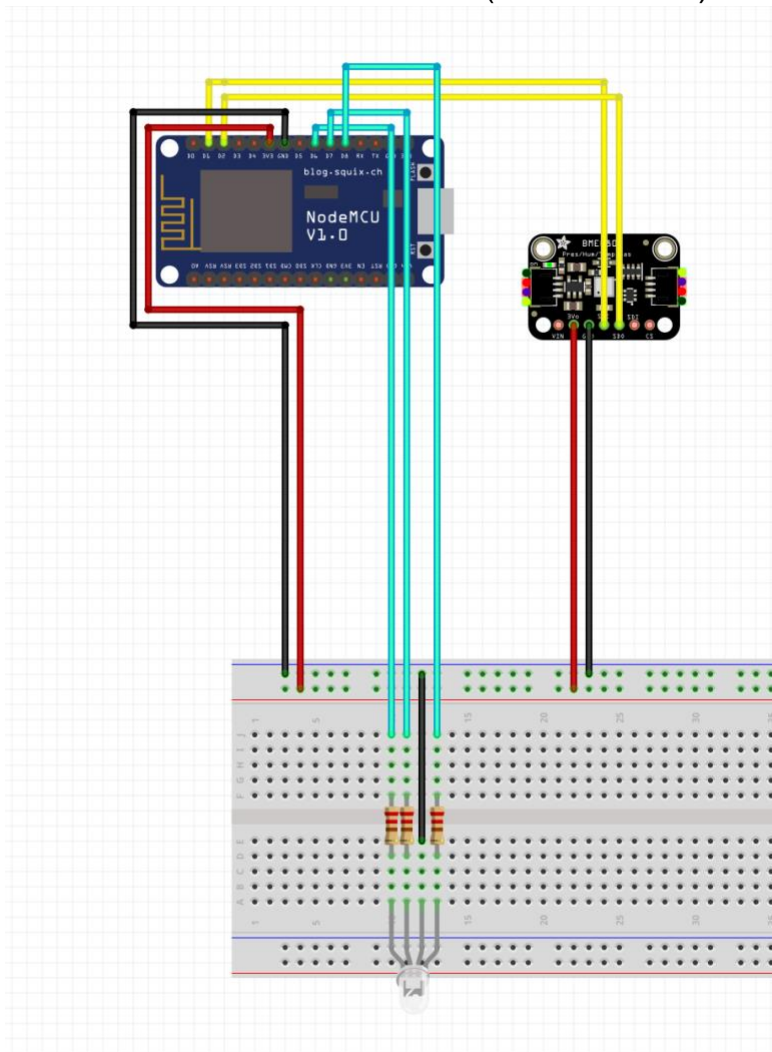
Nr.	Datum	Version	Geänderte Kapitel	Art der Änderung	Autor	Status
1	20.12.2021	0.5	Alle	Erstellung	Noah Canadea	ok
2	03.01.2022	0.6	6, 7, 8	Ergänzung	Noah Canadea	Ok
3	08.01.2022	0.7	Alle	Anpassungen	Noah Canadea	Ok
4	09.01.2022	1	Alle	Anpassungen und Ergänzungen	Milan Bursac	Ok

Inhalt

1	Schemas	3
1.1	Elektronik Schema ESP (Wetterstation)	3
1.2	Aktivitätsdiagramme	4
1.2.1	ESP Code (Wetterstation)	4
1.2.2	NodeRed (Backend)	6
1.3	MySQL Schema	7
1.4	Prinzipschema Cloud	7
2	Soll – Ist Abgleich	8
3	Testprotokoll	10
4	Zeitplan	11
5	Verwendete Produkte und Librarys	12
5.1	Backend / Frontend	12
5.2	ESP8266 Wetterstation Librarys	12
5.3	ESP8266 Wetterstation Hardware	12
6	Dashboards	12
7	Anhang	12

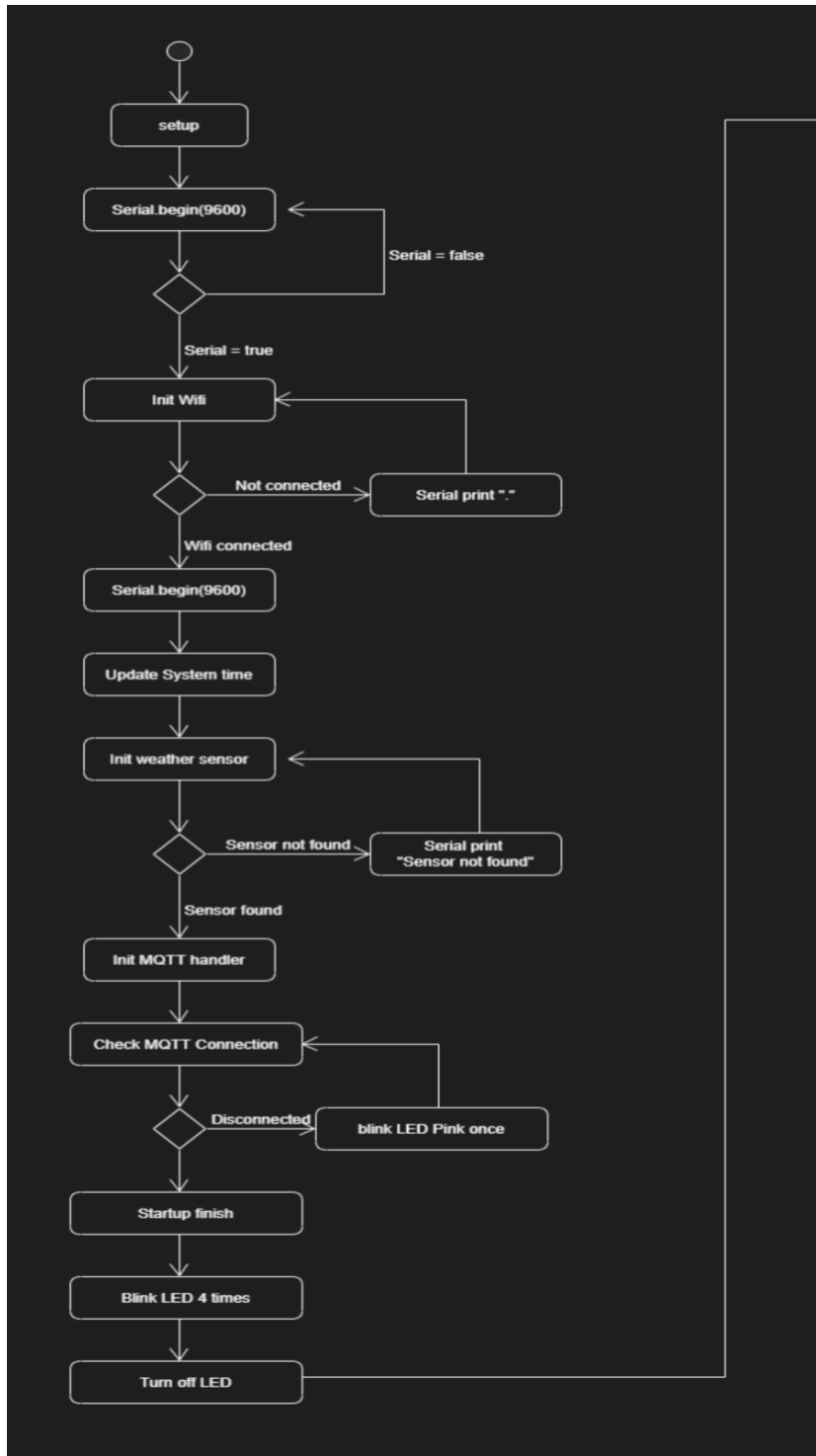
1 Schemas

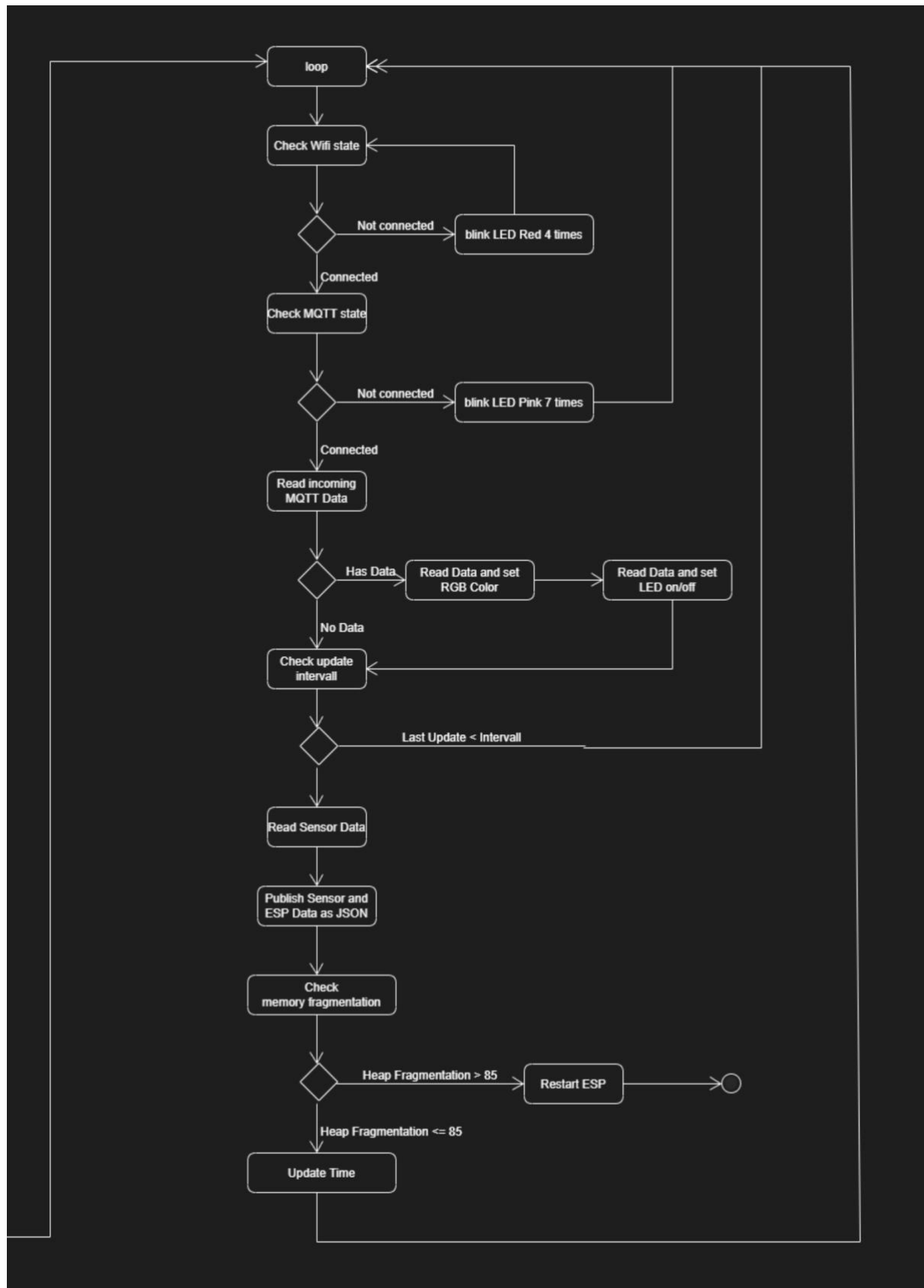
1.1 Elektronik Schema ESP (Wetterstation)



1.2 Aktivitätsdiagramme

1.2.1 ESP Code (Wetterstation)

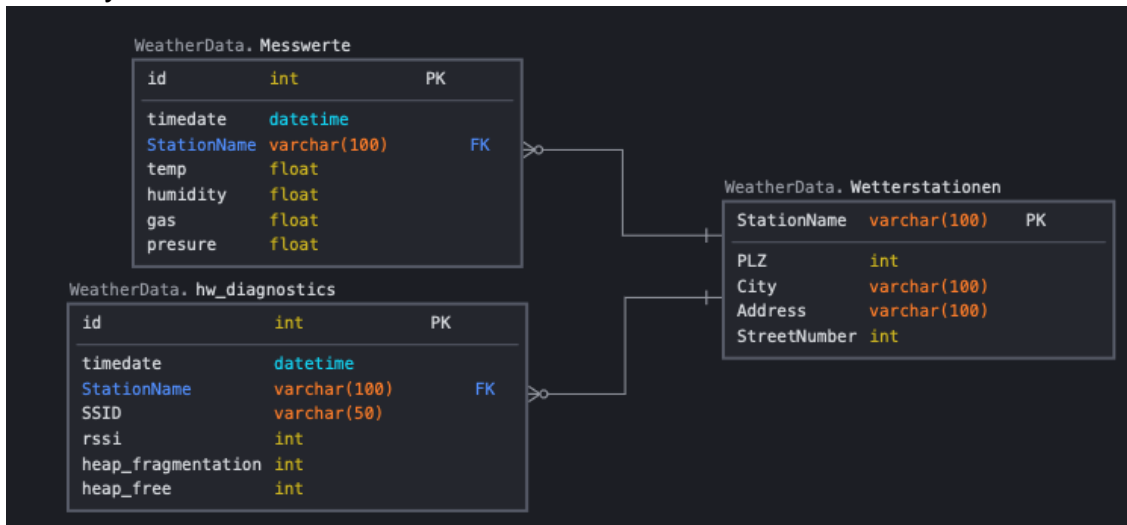




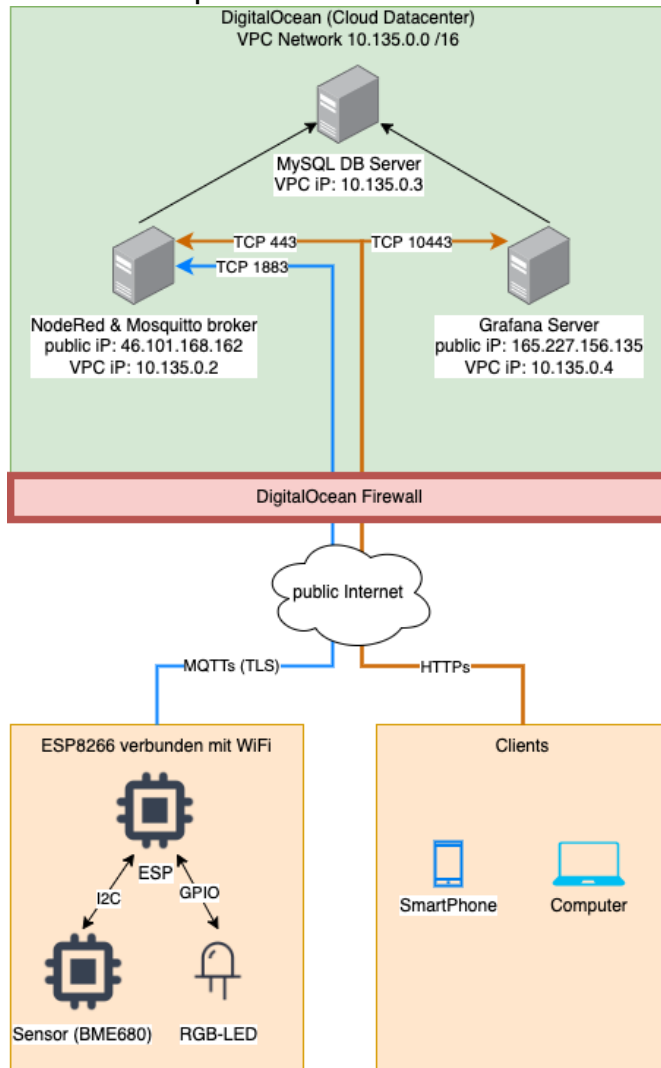
Live Data Handler



1.3 MySQL Schema



1.4 Prinzipschema Cloud



2 Soll – Ist Abgleich

Soll	Ist	Erfüllt
4.1 Erheben folgender Messwerte: <ul style="list-style-type: none"> - Temperatur - Luftfeuchtigkeit - Luftqualität - Luftdruck 	Temperatur, Luftfeuchtigkeit, Luftqualität und Luftdruck werden aufgezeichnet.	JA
4.2 Erheben diverser Metriken der Wetterstation: <ul style="list-style-type: none"> - iP Adresse - Hostname - Systemzustand - Letztes Update der Station usw. 	Die Station sendet diverse Metadaten an den NodeRed Server: <ul style="list-style-type: none"> - Hostname - Verbundene SSID - WiFi Stärke in RSSI - iP Adresse - Heap Fragmentierung - Freier Heap Speicher 	JA
4.3 Persistente Speicherung der Messwerte in der Cloud: Sämtliche, aufgezeichneten Messwerte sollen auf einem zentralen Server in der Cloud für einen definierten Zeitraum gespeichert werden. Hierzu soll eine DB wie Bspw. MySQL verwendet werden, auf welche zu einem späteren Zeitpunkt weitere Systeme für die Auswertung angebunden werden können.	Die Messwerte werden von der Wetterstation als JSON Payload per MQTTs an den NodeRed gesendet. Dieser speichert die Werte anschließend in einem MySQL Server.	JA
4.4 Fehlererkennung: Die Wetterstation soll die Möglichkeit haben, im Falle eines Fehlers (Server Verbindung getrennt, keine Netzwerkverbindung usw.), dies mithilfe einer LED oder eines Displays zu signalisieren.	Verliert die Station die MQTT oder WLAN-Verbindung, wird dies mittels einer RGB LED signalisiert. Je nach Fehler, wird eine andere Farbe verwendet: Rot blinkend: Kein WLAN. Orange blinkend: MQTT Fehler	JA
5.1 Technologien Für die Entwicklung sollen Produkte und Technologien nach aktuellem Stand der Technik eingesetzt werden. Gibt es für eine spezifische Anforderung bereits eine Library, kann diese sofern ausgewiesen verwendet werden.	Sämtlicher Code wurde nach bestem Wissen und Stand der Technik entwickelt. Die verwendeten Module können im Punkt 3 dieses Dokuments entnommen werden.	JA
5.2 Sicherheit Sämtliche Systeme sollen nach aktuellem Stand der Technik abgesichert werden. Verbindungen, welche über das Internet aufgebaut werden, müssen mit Hilfe von SSL/TLS und gültigen Zertifikaten verschlüsselt werden, um ein Mitlesen oder Manipulieren der Daten seitens Dritter zu erschweren.	Das NodeRed UI und Grafana sind über HTTPs mit einem gültiges SSL Zertifikat gesichert. Sämtliche MQTT Kommunikation zwischen Wetterstation und NodeRed (Mosquitto) ist per TLS gesichert. Lediglich mit der Zertifikatsprüfung gibt es noch Probleme, weshalb diese zum aktuellen Zeitpunkt deaktiviert ist.	Teilweise
5.3 Hosting Weil der Kunde die Server nicht lokal betreiben möchte, sollen diese bei einem großen Cloud-Provider gehostet werden. Die Auswahl des Anbieters ist hierbei dem Entwickler überlassen.	Der MySQL Server, Grafana Server und Node-Red Server werden als separate Ubuntu VMs bei DigitalOcean gehostet.	JA

<p>5.4 Erweiterbarkeit</p> <p>Um Erweiterungen zu einem späteren Zeitpunkt zu ermöglichen, müssen sämtliche Software-Module möglichst modular und erweiterbar konzipiert werden.</p>	<p>Der Source-Code der Wetterstation wurde möglichst modular aufgebaut. Für verschiedene Aufgaben wurden eigene Klassen erstellt (Bspw. für das MQTT Handling).</p> <p>Die DB wurde so konzipiert, dass mehrere Wetterstationen gleichzeitig Daten aufzeichnen können.</p>	<p>JA</p>
---	--	-----------

3 Testprotokoll

Test	Erwartetes Ergebnis	Tatsächliches Ergebnis	Erfüllt
Simulieren einer fehlenden WLAN-Verbindung der Wetterstation.	Die LED Blinkt einige Male Rot und pausiert anschliessend. Dies geschieht so lange, bis die WLAN-Verbindung wieder aufgebaut ist.	Die LED Blinkt einige Male Rot und pausiert anschliessend. Dies geschieht so lange, bis die WLAN-Verbindung wieder aufgebaut ist.	Ja
Simulieren einer fehlenden MQTT-Verbindung der Wetterstation.	Die LED Blinkt einige Male Violett und pausiert anschliessend. Dies geschieht so lange, bis die MQTT-Verbindung wieder aufgebaut ist. Auf dem NodeRed Dashboard sollte nach überschreiten des Updateintervall um 20s eine Meldung erscheinen, dass die Station offline ist.	Die LED Blinkt einige Male Violett und pausiert anschliessend. Dies passiert so lange, bis die MQTT-Verbindung wieder hergestellt wurde. Auf dem NodeRed Dashboard ist nach 50s eine Warnmeldung erschienen, dass die Station keine Daten mehr sendet.	Ja
Betreiben der Station für einen längeren Zeitraum, um sicherzustellen, dass diese Stabil läuft.	Die Station sollte ohne Probleme über einen längeren Zeitraum betrieben werden können. Die aufgezeichneten Daten werden hierbei mit dem definierten Updateintervall an den NodeRed gesendet.	Die Station ist Stand 08.01.2022 seit über einer Woche kontinuierlich in Betrieb. Bis jetzt konnten keine Probleme festgestellt werden. Auch das Updateintervall wird zeitlich eingehalten.	Ja
Validieren der Genauigkeit der Messdaten.	Die von der Wetterstation gesammelten Daten sollten relativ genau sein.	Wir haben die vom BME 680 Sensor aufgezeichneten Daten mit einem anderen Produkt (Netatmo) abgeglichen. Die Messwerte sind hierbei ziemlich genau. Bei der Temperatur beträgt der Unterschied Bspw. höchstens 0.2 – 0.4°C.	JA

4 Zeitplan

	06.12.21	13.12.21	20.12.21	27.12.21	03.01.22	09.01.22
Sensor bestellen	- M	-	-	-	-	-
Aktivitätsdiagramm	M	M	M	M	-	-
Hardwarediagramm	M	M	-	-	-	M
Software schreiben	- -	M N	M N	M N	M N	-
Node-RED konfigurieren	- N	N	N	N	N	-
Digital Ocean Server aufsetzen	N -	-	-	-	-	-
Mosquitto konfigurieren	N -	-	-	-	-	-
MySQL konfigurieren	N -	-	-	-	-	-
Grafana konfigurieren	N -	N	N	N	N	-
Dokumentation schreiben	M M	M M	M N	M N	M N	M -
Pflichtenheft	- -	-	-	- N	- N	- N
Präsentation	- -	-	-	-	M -	M -
Testprotokoll	- -	-	-	-	M N	M N
Korrekturen	- -	-	-	-	M N	M N

5 Verwendete Produkte und Librarys

5.1 Backend / Frontend

- [Node-Red](#): Backend und Frontend UI.
 - o [Node-red-dashboard](#): Diverse UI Elemente.
 - o [Node-red-node-mysql](#): Verbindungen zum MySQL Server.
 - o [Node-red-contrib-counter](#): Baustein welcher einen Counter ausgibt.
- [MySQL](#): DB Server zur Speicherung der Messwerte.
- [Grafana](#): Visualisierung der MySQL Daten.
 - o [DynamicText](#): Modul zur darstellung von dynamischem Text basierend auf SQL Querys.

5.2 ESP8266 Wetterstation Librarys

- [Adafruit BME680](#): Library zum Auslesen der Sensorwerte.
- [ESP8266WiFi](#): Library für WLAN und TLS Verbindungen.
- [PubSubClient](#): Library für MQTT Verbindungen zwischen Node-Red und Wetterstation.
- [NTPClient](#): Library für Zeitsync per NTP.
- [ArduinoJson](#): Library zum parsen von Daten zu Json Objekten.

5.3 ESP8266 Wetterstation Hardware

- [NodeMCU ESP8266](#): Hauptmodul
- [RGB LED](#): Status LED
- [BME 680](#): Wettersensor

6 Dashboards Links

- [Grafana Dashboard](#)
- [Node-Red Dashboard](#)

7 Anhang

- Plattform IO Projekt der Wetterstation
- NodeRed Konfiguration
- Grafana Konfiguration
- SQL Statements zur Erstellung der DB
- Pflichtenheft