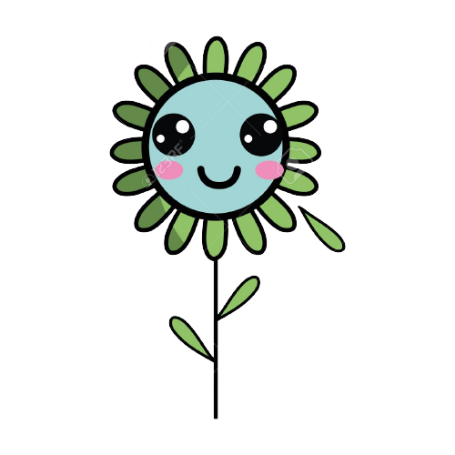
Projektdokumentation

Überbetrieblicher Kurs

Modul 216

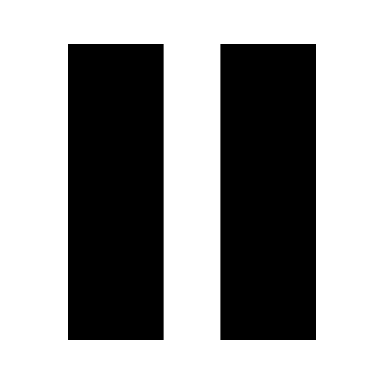
Gruppe 6





Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



Inhalt

[Einstieg 3](#_Toc88825019)

[Anforderungen 3](#_Toc88825020)

[Optionale Anforderungen 3](#_Toc88825021)

[Ressourcen 3](#_Toc88825022)

[Datenfluss 4](#_Toc88825023)

[4](#_Toc88825024)

[4](#_Toc88825025)

[Ablauf 4](#_Toc88825026)

[Realisierung 4](#_Toc88825027)

[Planung und Entscheidung 4](#_Toc88825028)

[Grundaufbau 5](#_Toc88825029)

[LED 5](#_Toc88825030)

[Switch 6](#_Toc88825031)

[Tests 6](#_Toc88825032)

[Testkonzept 6](#_Toc88825033)

[Testdrehbuch 6](#_Toc88825034)

[Probleme 8](#_Toc88825035)

[Sensoren Wert 8](#_Toc88825036)

[Sensor Rost 8](#_Toc88825037)

[Node RED Port 8](#_Toc88825038)

[Buffer Overflow 8](#_Toc88825039)

[RGB LED 9](#_Toc88825040)

[Keine Pflanze mit Erde 9](#_Toc88825041)

[Reflexion 9](#_Toc88825042)

[Team 9](#_Toc88825043)

[Quellen 9](#_Toc88825044)

# Einstieg

Viele Menschen besitzen Zuhause oder im Büro eine oder mehrere Pflanzen.

Dabei kommt es hin und wieder vor, dass das Bewässern vergessen wird und die Pflanze deshalb verwelkt.

In diesem Projektauftrag soll nun ein IoE Messgerät entwickelt werden, welches die Feuchtigkeit einer Pflanze (Topferde) misst und einem signalisiert, dass die Pflanze Wasser braucht.

# Anforderungen

Für das Projekt haben wir uns Anforderungen gesetzt, welche wir bis zur Abgabe erfüllt haben wollen.

* Umsetzung gemäss Bewertungsraster (siehe Dokument «Bewertung» in Microsoft Teams)
* Zeitlicher Verlauf
* Die Feuchtigkeit der Erde soll gemessen werden und bei vorbestimmten Werten soll ein LED-Lämpchen die Farbe wechseln.
* Werte der Pflanze sichtbar machen (Node-RED)

# Optionale Anforderungen

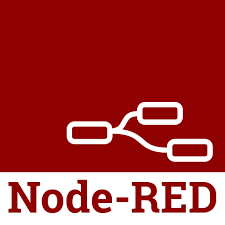
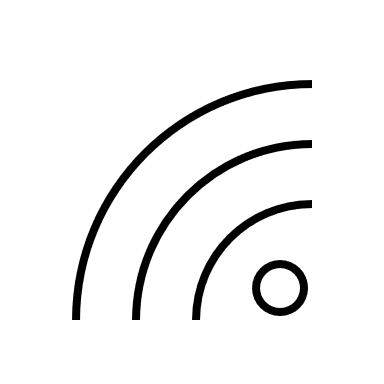
* (Optional bei genügend Zeit) Möglichkeit zur Benachrichtigung über ein Kommunikationskanal (E-Mail, SMS…)
* (Optional bei genügend Zeit) Der Zustand der Pflanze soll über einen OLED Bildschirm veranschaulicht werden.

# Ressourcen

|  |  |
| --- | --- |
| Ressource: | Anzahl: |
| ESP 32 Mikrocontroller | 1x |
| Hygrometer Sensor | 1x |
| Jumper Kabel (female-male) | 7x |
| RGB LED | 1x |
| Breadboard (klein) | 1x |
| HW 103 | 1x |
| Mikro USB-Kabel | 1x |
| Arduino (Software) |  |

# Datenfluss

Ein Bild, das Text, Elektronik, Schaltkreis enthält.

Automatisch generierte Beschreibung

Sensordaten Sensordaten

# 

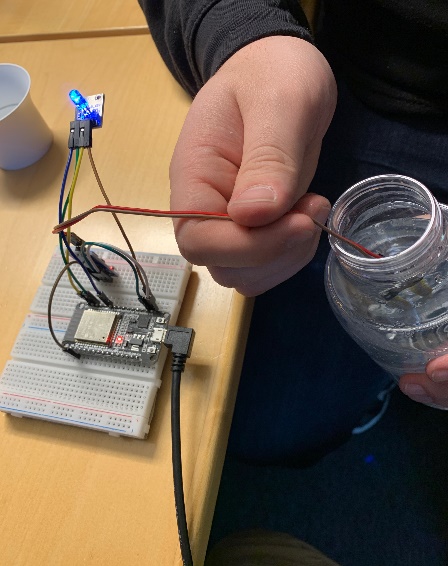
# Drahtlos SilhouetteEin Bild, das Text, Elektronik, Schaltkreis enthält. Automatisch generierte Beschreibung

## Ablauf

Unser Sensor misst die Feuchtigkeit der Erde durch die Leitfähigkeit. Diese Werte werden dann über die Jumper Kabel erst dem HW 103 und dann dem ESP 32 weitergeleitet. Der ESP 32 schickt dann diese Werte über das WLAN unserer Node-RED Instanz weiter. In unserer Instanz werden dann die Werte in einem Graphen umgewandelt. Auch kann uns eine Funktion in der Instanz ab einem gewissen Wert eine E-Mail schicken, welche den Zustand der Pflanze beschreibt. Zusätzlich können mit einer LED anzeigen lassen, dass sich der ESP 32 mit dem WLAN verbindet und was der Zustand der Pflanze ist. Zusätzlich können wir über einen Switch in unserer Instanz den Sensor aus/an -stellen.

# Realisierung

## Planung und Entscheidung

Vor unserer Realisierung haben wir mit dem Planen und Entscheiden angefangen. Als erstes haben wir uns also zusammengesetzt und entschieden, welches Thema wir als Projekt behandeln möchten. Die Entscheidung lag vor allem bei Mikka, da er schon im Vorhinein die Idee hatte, ein IoT Gerät zu programmieren, welches ihm den Zustand seiner Pflanzen nennt. Zusätzlich war es einer der Vorschläge, zu was wir ein Projekt führen sollten. Nun gingen wir zur Planungsphase über. Wir begannen damit zusammenzutragen, welche Ressourcen wir für unser Projekt benötigen. Auf jeden Fall bräuchten wir einen ESP 32 Mikrocontroller, ein Breadboard und einen Feuchtigkeitssensor. Wir haben uns auch noch vorgestellt, dass wir den Zustand der Pflanze direkt anzeigen lassen können durch ein LED-Lämpchen. Eventuell wollten wir auch noch bei genügend Zeit ein OLED Bildschirm einbinden, welcher durch Emojis den Zustand der Pflanze zusätzlich verdeutlicht. Auch bräuchten wir auf jeden Fall noch ein paar Jumper Kabel, um alles miteinander zu verkabeln.

## Grundaufbau

Nach dem wir die Ressourcen einmal aufgeschrieben und unseren Projektauftrag verdeutlich haben, sammelten wir die ersten Ressourcen zusammen und begannen damit, alles zu verkabeln was wir bisher benötigten. Erst verbanden wir den Sensor mit dem HW 103 Modul, welcher dann mit dem ESP 32 angeschlossen wurde. Bevor wir unsere LED verbinden, wollten wir erst ein Programmbeispiel finden und uns davon Ausschnitte für unser Programm holen. Um nun erste Daten zu bekommen, benutzten wir eine Wasserflasche, in welche wir den Sensor reinlegten.

Über den Serial Monitor bekamen wir nun einige Messdaten des Sensors. Allerdings waren diese Werte in einem negativen Bereich und sobald der Sensor das Wasser berührte, stieg der Negative Wert von -512, auf -304. Für uns war dieser Merkwürdige Wertebereich ein Dorn im Auge. Dafür haben wir uns erstmal angeschaut, welche Zahlen der Sensor ohne Berechnungen versendet. Ohne Berechnung im Programm, erhielten wir einen Konstanten Wert von 4095. Sobald nun der Sensor ins Wasser geriet, sank der Wert drastisch. Um nun logische Werte zu bekommen, benutzten wir eine Mathematische Formel, welche uns den Wert 4095 auf 0% setzte und sobald der Wert durch das Wasser sank, stieg unser Endwert auf über 60%. Somit hatten wir nun eine Logische Skala, 0% ist trocken und 60%+ ist nass. Damit wir feststellen können, ob unsere Daten überhaupt nützlich angewendet werden können auf eine Pflanze, haben wir die Sensoren über Nacht laufen lassen. Damit konnten wir sehen, dass die Feuchtigkeit abgenommen hat und wir so eine ungefähre Idee haben, welche Werte eine Pflanze generiert.

## LED

Mit den Werten konnten wir nun unsere LED einstellen. Dafür verbanden wir diese mit dem ESP 32 über das Breadboard. Nun schrieben wir im Programm das, sobald der Sensor einen Gewissen Prozent Wert hat, sich die Farbe der LED ändert. Schlussendlich leuchtet unsere LED rot für Trocken, bis einem Feuchtigkeitswert von 5%. Ab 5% bis 55% leuchtet die LED grün, wobei sich mit 4 Stufen, das Blau intensiviert.

Ein Bild, das Text, Elektronik enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Elektronik enthält.

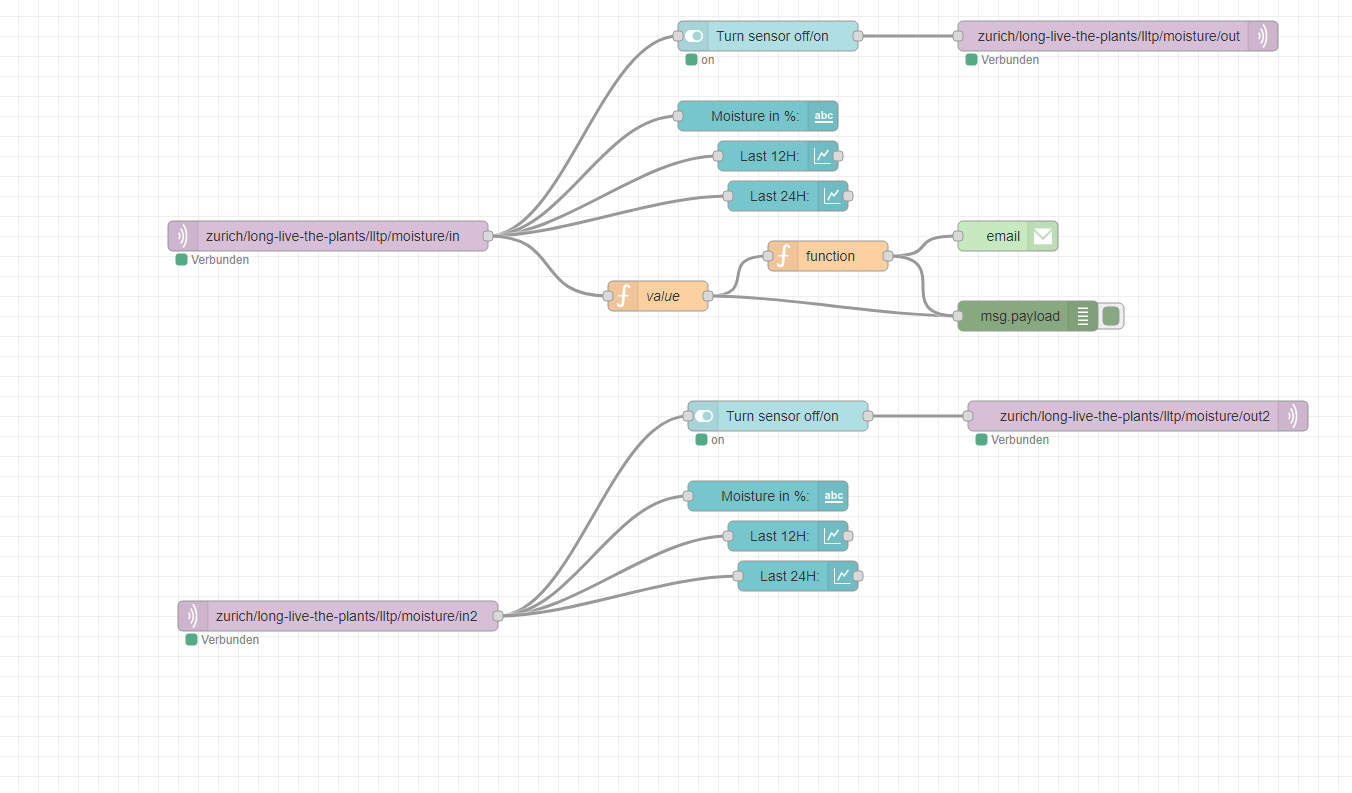
Automatisch generierte BeschreibungEin Bild, das Text, Elektronik enthält.

Automatisch generierte BeschreibungAb einem Feuchtigkeitsprozent von über 55% wird die LED blau.

Nachdem die LED eingestellt war, fingen wir an, Node RED zu verbinden. Dafür senden wir die Daten über das WLAN. Unsere Node RED Instanz war anfangs sehr einfach gehalten. Wir haben unsere Werte simpel in einem Graph wiedergegeben. Um nun einige Punkte bei der Evaluation zu integrieren, haben wir für unsere MQTT Verbindung ein Passwort gesetzt. Auch schicken wir nun dem Sensor einen Wert zurück, welcher nun die LED auf dem HW 103 aufblinken lässt, sobald der ESP 32 kein WLAN hat oder die Verbindung sucht.

## Switch

Unsere nächste Idee war, den Sensor über Node-RED deaktivieren zu können. Dafür haben wir in unserer Instanz einen Switch eingebaut welcher nun einen True/False Wert zurückgibt. Dieser Wert soll nun die loop Funktion steuern. Solange unser Switch in Node-RED auf True ist, läuft innerhalb der loop Funktion unser Programm zum Sensor und zum Senden der Daten.



# Tests

## Testkonzept

Während des Verlaufs unseres Projektes haben wir immer wieder kleinere Test zu den Funktionen gemacht. Unsere Tests basierten dabei immer darauf, dass wir kontrollieren, ob uns der Sensor Daten sendet, da sonst der Rest unserer Test keinen Sinn gehabt hätte. Nach dem Punkt hatten unsere Tests jeweils andere Ziele, die wir erreichen wollten.

## Testdrehbuch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testfall-Nr | | | 1 | | | |
| Testfall-Bezeichnung | | | Datensignaltest | | | |
| Anforderungs-Nr. | | | 1 | | | |
| Testumgebung | | | Noser Young  ESP 32, Hygrometer, Serial Monitor, Jumper Kabel, Wasserflasche | | | |
| Zu testende Funktionalität | | | Das Empfangen von Werten testen, ob diese brauchbar sind. | | | |
| Datum der Testdurchführung | | | 24.11.2021 | | | |
| Tester | | | Cedric Zollinger, Mikka Kummer | | | |
| Testschritte: | | | | | | |
| **Nr.** | **Aktion** | **Erwartetes Ergebnis** | | **Effektives Ergebnis** | **Erfüllt** | **Kommentar** |
| 1 | Sensor mit Beispiel-programm ins Wasser gehalten | Ein Wert zwischen  300 und 526 | | Werte lagen im  Minusbereich, bei  weniger Feuchtigkeit stieg der Wert an | Nein | Unser Werte hätten laut Beispielcode im erwartenden Bereich liegen müssen |
| 2 | Sensor mit angepasstem Programm ins Wasser gehalten | Ein Wert zwischen über 4000 | | Der höchste Wert lag immer bei 4095 | Ja | Mit nachrechnen der umgekehrten Formel erwarteten wir einen Wert über 4000 |
| 3 | Rechnungsformel im Programm anpassen für logischere Werte | Werte welche in einem Prozentbereich zwischen 0-100 liegen | | Der Trockenste Wert war 65% und der Feuchteste 0% | Jein | Wir erhielten Werte im erwartenden Bereich, aber waren falsch herum aufgelistet |
| 4 | Rechnungsformel nochmals angepasst | Bei einem trocknen Sensor ist der Wert auf 0%, wird der Sensor ins Wasser gehalten sollte dieser steigen | | Der trockenwert lag bei 0%. Der Nässeste Wert lag bei über 65% | Ja | Wir hatten uns eher erhofft Werte zwischen 0-100%, aber mit 0-65% konnten wir auch verkraften und damit abreiten. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testfall-Nr | | | 2 | | | |
| Testfall-Bezeichnung | | | Werte in Node RED | | | |
| Anforderungs-Nr. | | | 2 | | | |
| Testumgebung | | | Noser Young  ESP 32, Hygrometer, Serial Monitor, Jumper Kabel, Pflanze mit Erde | | | |
| Zu testende Funktionalität | | | Ist Node RED richtig verbunden und gibt uns der Graph Werte aus | | | |
| Datum der Testdurchführung | | | 24.11.2021 | | | |
| Tester | | | Cedric Zollinger, Mikka Kummer | | | |
| Testschritte: | | | | | | |
| **Nr.** | **Aktion** | **Erwartetes Ergebnis** | | **Effektives Ergebnis** | **Erfüllt** | **Kommentar** |
| 1 | Erste Verbindung aufbauen und einfachen Liniengraph benutzen | Node RED soll die gleichen Werte wie der Serial Monitor erhalten und uns diese im Graph anzeigen | | Node RED lieferte keine Angaben. | Nein | Nach dem Missglück entdeckten wir, dass unsere Ports im MQTT Explorer und im Code falsch eingestellt waren |
| 2 | Verbindung wird versucht aufzubauen | Node RED soll die gleichen Werte wie der Serial Monitor erhalten und uns diese im Graph anzeigen | | Node RED erhielt die Werte unseres Sensors und stellte uns diese in einem Liniengraph dar | Ja | Der Graph zeigte uns die gleichen Werte wie im Serial Monitor an |
| 3 | Sensor in eine Pflanze stecken und den ESP 32 über eine Powerbank befeuern | Der Sensor sollte uns Werte der Pflanze senden und uns diese weiterhin in Node RED darstellen | | Der Sensor lieferte weiterhin Daten und konnte diese Node RED zuschicken | Ja | Wie erwartet lieferte der Sensor weiterhin Daten |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testfall-Nr | | | 3 | | | |
| Testfall-Bezeichnung | | | LED | | | |
| Anforderungs-Nr. | | | 3 | | | |
| Testumgebung | | | Noser Young  ESP 32, Hygrometer, Serial Monitor, Jumper Kabel, Pflanze mit Erde, LED | | | |
| Zu testende Funktionalität | | | Die LED sollte bei gewissen Werten die Farbe wechseln. | | | |
| Datum der Testdurchführung | | | 24.11.2021 | | | |
| Tester | | | Cedric Zollinger, Mikka Kummer | | | |
| Testschritte: | | | | | | |
| **Nr.** | **Aktion** | **Erwartetes Ergebnis** | | **Effektives Ergebnis** | **Erfüllt** | **Kommentar** |
| 1 | LED mit ESP 32 verbinden und im Programm vorgegebenen Werten, Farben zuweisen | Die LED sollte bei 0-5% Rot leuchten, dann bis 60% stufenweise vom Grünen ins Blaue wechseln und ab 60% blau werden. | | Die LED veränderte die Farbstufen und wurde blau ab 60%, jedoch gab es kein rot unter 5% | Nein | Für kurze Momente erhielten wir eine weisse LED und bei gewissen Werten war die LED aus. |
| 2 | Im Programm die Bedingungen der Farben ändern und jeweils eine zweite Bedingung einfügen | Die LED sollte bei 0-5% Rot leuchten, dann bis 55% stufenweise vom Grünen ins Blaue wechseln und ab 60% blau werden. | | Die LED lieferte uns die Erwünschten Farbwerte, aber bei unter 5% erleuchtete auch blau. Bei über 55% flackerte dann blau | Nein | Es stellte sich heraus, dass wir für den blau Anschluss den inkorrekten Steckplatz auf dem ESP 32 benutzt haben |
| 3 | Steckplätze der Verbindungen nochmals überprüfen | Die LED sollte bei 0-5% Rot leuchten, dann bis 55% stufenweise vom Grünen ins Blaue wechseln und ab 60% blau werden. | | Die LED funktionierte nun genau so wie wir es wollten und veränderte die Farben ab dem gewünschten Wert | Ja | Die LED verursachte schon bei kürzerem Anschauen Probleme mit den Augen und Kopfschmerzen |

# Probleme

## Sensoren Wert

Unser Sensor lieferte uns merkwürdige Werte, obwohl uns der Beispielcode Werte im Bereich zwischen 300-526 liefern sollte, lieferte er uns einen Wert im negativen Bereich zwischen 500 und 700.

## Sensor Rost

Aufgrund der Konstruktion des Sensors ist er anfällig auf Rost. Bei den Tests, bei welchen wir eine Wasserflasche genommen haben mit Kohlensäure, sank der Wert des Sensors stetig. So mussten wir stetig die Kontakte abtrocknen, um so die Dünne Rostschicht zu entfernen und wieder korrekte Werte zu erhalten.

## Node RED Port

Jedem Team wurde eine Node RED Instanz zugewiesen. In Code haben wir also darum den Port des MQTT gewechselt zu unserem Node RED Port. Dadurch konnte der ESP 32 keine Daten senden.

## Buffer Overflow

Unser Char Array hat sich durch die Sensor Daten überfüllt, da dieser zu wenig Speicherplatz für sich freihalten konnte. Dadurch ist das Programm abgestürzt und der ESP 32 hat sich immer wieder rebootet.

## RGB LED

Unsere LED wurde anfangs falsch eingesteckt. Anfangs funktionierte sie dennoch wie geplant und leuchtete mit den korrekten Farben. Als wir aber den Wertebereich angepasst haben, meldete sich die blaue Farbe schon bei 0-5%, obwohl dort nur rot leuchten sollte. Nicht nur leuchtete die Farbe, sondern sie flackerte auch, was sehr unangenehm war in den Augen. Aber nachdem wir die Kontakte richtig miteinander verbunden haben, lief die LED ohne Zwischenfälle in den korrekten Farben.

## Keine Pflanze mit Erde

Eines der grössten Probleme, welches wir nicht durch Programmieren oder Umstecken lösen konnten war, dass wir um Büro keine Pflanze gefunden haben, die wie eine normale Topfpflanze in Erde gepflanzt ist. Schlussendlich haben wir aber noch eine gefunden, welche Tatsächlich noch brauchbar gewesen ist, um unseren Sensor hineinstecken zu können.

# Reflexion

## Team

Wir haben uns als Team schnell gefunden. Wir haben oft miteinander kommuniziert und uns zusammen Ziele gesetzt. Auch bei den Aufteilungen wer welche Arbeit übernimmt, haben wir ohne grosse Konversationen entscheiden können. Auch selbst haben wir im Team nie Konflikte geführt. Als Team konnten wir so sehr effizient arbeiten, auch weil wir uns nicht gegeneinander in die Quere gekommen sind. Beiden hat das Projekt Spass gemacht. Ein Punkt, welchen wir aber sehr stressig fanden, waren die Journale, da wir zurzeit 5 führen müssen inklusive dieser Projektdokumentation.

## Mikka

Dieses Projekt, war sehr spannend. Ich konnte sehr viel lernen, was ich auch zuhause anwenden kann/werde.

Es hat Spass gemacht an einem physischen Objekt zu arbeiten, da man seinen Code Live umgesetzt sieht und nachverfolgen kann was passiert.

Unsere Teamdynamik war sehr angenehm und wir kamen effizient voran, da wir keine Kommunikationsprobleme hatten und eine klare Arbeitsaufteilung.

Ein paar Informationen bezüglich MQTT und Node-RED wurden etwas dürftig uns gegenüber kommuniziert. Beispielsweise das man den Port bei 1883 lassen muss und nicht auf seine Instanz anpassen muss. Diese Info hätte uns einiges an Zeit gespart.

Alles in allem war das Projekt sehr lehrreich und interessant. Ich habe mit viel Freude und Elan daran gearbeitet.

## Cedric

Ich hatte während des Projekts sehr viel Spass. Ich konnte vieles lernen und sehen nun einige «Smart» Produkte aus einem anderen Blickwinkel. Interessant war, wie viele neue Möglichkeiten man sieht, nützliche Produkte zu erstellen, welche man Zuhause verwenden könnte, um einiges zu erleichtern. Unser Team funktionierte sehr gut. Wir haben beide unsere Stärken am passenden Ort einsetzten können um so unsere Leistung als Team zu ergänzen. Es gab auch Situationen in denen wir ein bisschen frustriert waren, warum etwas nicht lief, wie es sollte, aber diese Sequenzen konnten wir zusammen gut lösen.

# Quellen

Abbildung 1: ESP 32 Mikrocontroller

<https://media.digikey.com/photos/Schtoeta-Engineering/MFG_ESP32-DEVKITC-32E.jpg>

Abbildung 2: Lachende Pflanze <https://previews.123rf.com/images/stockgiu/stockgiu1706/stockgiu170600557/79584203-kawaii-gl%C3%BCckliche-blume-pflanze-mit-bl%C3%A4ttern-und-bl%C3%BCtenbl%C3%A4tter-vektor-illustration.jpg>

Abbildung 3: <https://i2.wp.com/www.esp8266learning.com/wp-content/uploads/2017/06/Soil-Moisture-Sensor.jpg?resize=355%2C355>

Abbildung 4: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQWgVuppfVqS6ikt0a1vGdYon8IxsL377Le-Q&usqp=CAU>

Abbildung 5: Node-RED

<https://upload.wikimedia.org/wikipedia/commons/2/2b/Node-red-icon.png>

Abbildung 6:

<https://res.cloudinary.com/rsc/image/upload/b_rgb:FFFFFF,c_pad,dpr_1.0,f_auto,h_843,q_auto,w_1500/c_pad,h_843,w_1500/F2286004-01?pgw=1&pgwact=1>