

Lernjournal

Thema: Lernjournal für den üK 216

Dokumentinformationen

Dateiname: Lernjournal_uk_216_OsDo.docx
Speicherdatum: 25.11.2022

Autoreninformationen

Autor: Oscar Doell
E-Mail: oscar.doell@noseryoung.ch
Tel: +41 31 917 53 70

Inhaltsverzeichnis

1	Tag 1	3
1.1	Tagesablauf	3
1.2	Wissenssammlung.....	3
1.3	Reflexion.....	5
2	Tag 2	5
2.1	Tagesablauf	5
2.2	Wissenssammlung.....	5
2.3	Reflexion.....	8
3	Tag 3	9
3.1	Tagesrückblick.....	9
3.2	Wissenssammlung.....	9
3.3	Reflexion.....	14
4	Tag 4	14
4.1	Tagesablauf	14
4.2	Wissenssammlung.....	14
4.3	Reflexion.....	16
5	Tag 5	16
5.1	Tagesablauf	16
5.2	Wissenssammlung.....	16
5.3	Reflexion.....	17
6	Tag 6	17
6.1	Tagesablauf	17
6.2	Wissenssammlung.....	18
6.3	Reflexion.....	18
7	Quellenverzeichnis	18

1 Tag 1

1.1 Tagesablauf

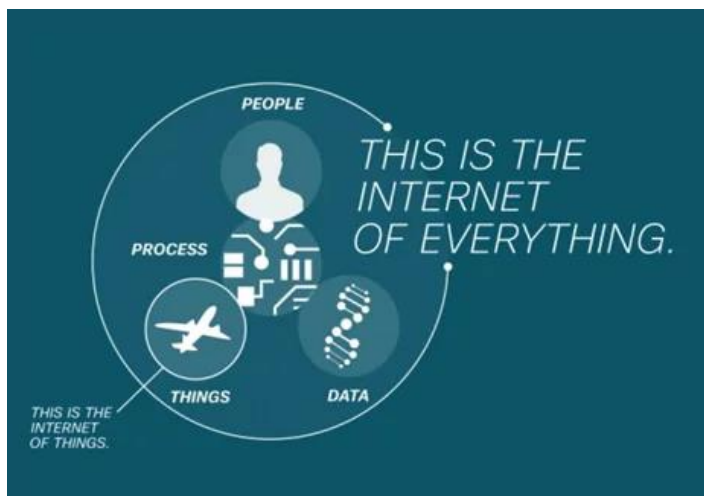
Tagesablauf von Mittwoch, 16.11.2022

- Mit einem Input den üK 216 gestartet
- Lerngruppen gebildet
- PDF 01_modul-216-ku-grundlagen-ioe-iot gelesen und wichtige Informationen notiert
- Spike über Datensicherheit von IoT Geräte und ADC, DAC gehabt
- Mit Lerngruppen das gelesene PDF zusammengefasst
- Lernjournal angefangen

1.2 Wissenssammlung

Unterschied zwischen IoT and IoE:

IoE ist ein Superset von IoT. Das heißt das IoE ein Überbegriff für IoT ist, denn IoT fokussiert nur auf die Verbindung von Dingen mit dem Internet, aber IoE fokussiert auch auf die Kommunikation zwischen Menschen und Maschine und Maschine und Maschine.



[Quelle](#)

Bsp:

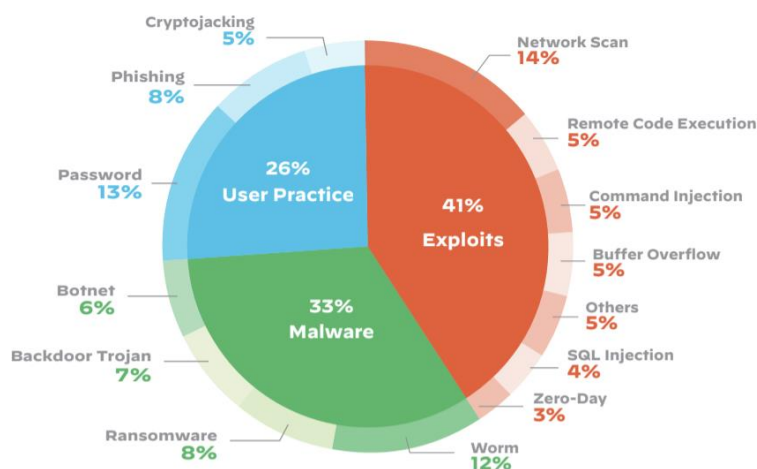
IoT Gerät: Eine Smart-Lampe, die man mit dem Handy anschalten, abschalten und die Farbe ändern kann.

IoE Gerät: Eine Smart-Lampe, der alles machen kann, was oben genannt wurde, aber es lernt z. B. wann du nach Hause kommst und schaltet automatisch das Licht an. Die Smart-Lampe hat auch eine Kamera drin, die Bewegungen merkt und dir auf dem Handy, egal wo du bist, eine Nachricht schicken kann, dass jemand zu Hause ist.

Was ist IIoT:

IIoT steht für Industrial Internet of Things und ist hauptsächlich geschäftlich ausgerichtet. Das heißt die Komponente müssen sehr zuverlässig sein, über längere Zeiten benutzbar sein und sie müssen auch sehr präzise sein.

Sicherheitsrisiken bei IoE und IoT Geräte



Quelle

In dieser Abbildung sieht man die Gründe für Cyberattacken auf IoE oder IoT Geräte. Man sieht hier 26 % der Cyberattacken wegen Fehler vom User passieren z.B. schlechte Passwörter, Phishing oder Cryptojacking. Die NCSC hat ein Dokument geschrieben, worauf steht, welche Maßnahmen man ergreifen soll, wenn man ein IoE oder IoT Gerät aufstellt.

Die wichtigsten Maßnahmen sind:

- Keine voreingestellte Benutzernamen / Passwörter benutzen
- Starke Passwörter benutzen (mindestens 12 Zeichen, Zahlen und Buchstaben sowie Sonderzeichen)
- Falls man das Gerät nicht mehr benutzt, dann sollte man das Gerät vom Internet trennen
- Keine Standardports verwenden (z. B. 23 – Telnet, 443 – HTTPS, usw.), sondern einen high-port (z. B. 2323 anstelle von 23, 43443 anstelle von 443 usw.)

Wenn man bereits Opfer zu einem Angriff geworden ist, sollte man einen Factory Reset machen und anschließend alle Maßnahmen bei der Wiederherstellung befolgen.

<https://www.ncsc.admin.ch/ncsc/de/home/infos-fuer/infos-unternehmen/aktuelle-themen/massnahmen-schutz-iot.html>

1.3 Reflexion

Ich habe heute vieles neues gelernt. Ich habe mir vor dem üK die Zusammenfassung auf Modulbaukasten angeschaut und wusste nicht, was die Hälfte bedeutet. Aber jetzt, nur nach einem Tag dieser üK, verstehe ich fast alles in dieser Zusammenfassung. Ich hatte am Anfang Mühe, weil ich nicht so ganz verstanden habe, was der Unterschied von IoT und IoE war. Es wurde mir dann aber erklärt und nach noch ein kleines bisschen Selbstrecherche kenne ich jetzt denn Unterschied. Ich konnte, nachdem ich den Unterschied von IoT und IoE gekannt hatte, die Begriffe und alles andere viel besser verstehen.

2 Tag 2

2.1 Tagesablauf

Tagesablauf von Donnerstag, 17.11.2022

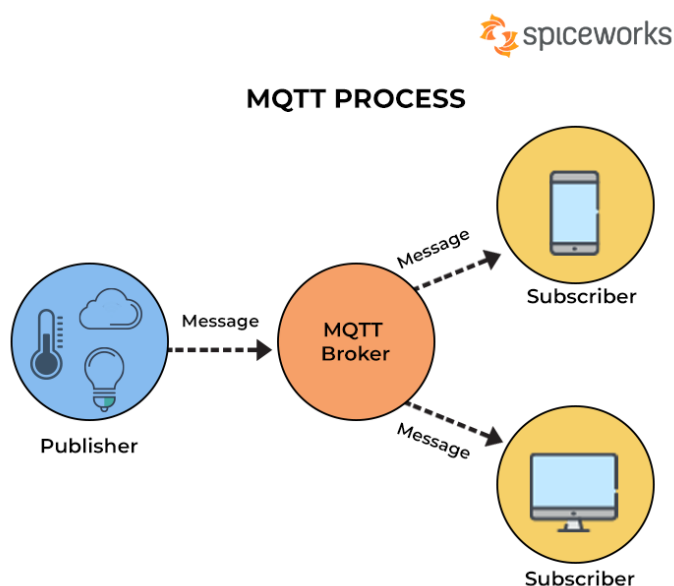
- Dokument 02, Kapitel 2 und 3 über MQTT und Node-RED gelesen
- Gelinkte Videos vom Dokument 02 über MQTT geschaut
- Dokument 00 ab Kapitel 6 bearbeitet
- Lernjournal geschrieben
- Dokument 02 ab Kapitel 4 bearbeitet

2.2 Wissenssammlung

MQTT:

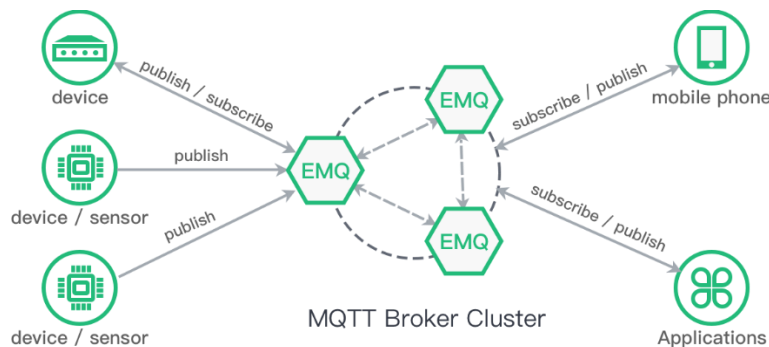
MQTT steht MQ Telemetry Transport

Viele Quellen geben, an dass es für Message Queue Telemetry Transport steht, aber das MQ kommt von einem Produkt namens MQ IBM. MQTT ist ein Publisher/Subscriber Model statt dem standardisierten Client/Server Model. Das heißt, dass das Gerät welche Informationen erhalten will, vom sogenannten Publisher muss zuerst zu den Nachrichten und Daten, die sie erhalten wollen, abonnieren (Subscriber). Solche Daten und Nachrichten heißen Topics.



Quelle

In dem Mittelpunkt dieser Publishers und Subscribers sitzt der sogenannte Broker. Der Broker ist ein Server, der die Daten vom Publisher bekommt und es den Subscriber zuschickt. MQTT ist aber «bi-directional», das heisst dass es auch von den Subscriber Daten bekommen kann, die es dem Publisher zuschickt. Weil dieser Broker im Mittelpunkt steht, kann es zu einem SPOF (Single Point of Failure) kommen. Wenn also der Broker Offline geht, können die Publisher keine Daten verschicken und die Subscriber können keine Daten erhalten. Moderne Broker umgehen dieses Problem mit Cluster.



Quelle

In einem Broker Cluster befinden sich mehrere Server, das heisst, dass wenn ein Server offline ist, die Daten immer noch von einem anderen Server geschickt und erhalten werden können.

Wenn man ein neues Gerät mit einem MQTT-Broker verbinden will, muss man ein Connect Package an dem Broker schicken, der Informationen enthält über dem Benutzer.

MQTT-Packet: CONNECT	
contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60

Quelle

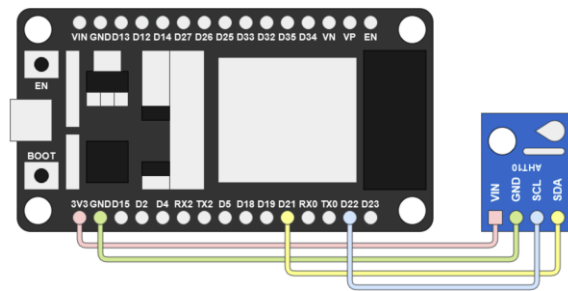
Eine cleanSession ist es dann, wenn man die Verbindung mit dem Broker schließt, wenn man das Gerät abschaltet.

Wenn alles richtig ist, dann schickt der Broker ein connack package zurück.

Das Gleiche Prinzip gilt auch, wenn man zu einem Topic subscriben will oder wenn man einen Publisher hinzufügen will. Dann gibt es aber beim Publisher publish und puback packages und beim subscriben subscribe und suback packages.

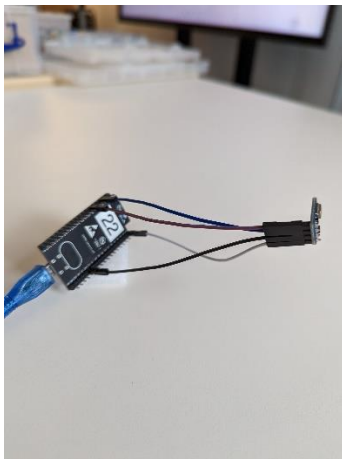
Erste on hand erfahrungen mit ESP-32:

Ich durfte heute zum ersten Mal richtig mit den ESP-32 und viele Sensoren arbeiten. Für unsere erste Aufgabe durften wir ein Temperatur Sensor an unserem ESP-32 verbinden.



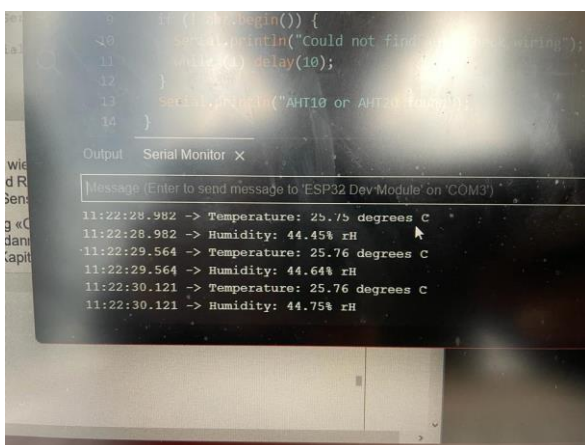
Quelle: Noser Young 00_modul-216-ku-getting-started_v1-4.pdf

Der Sensor muss so angeschlossen werden. Es muss eine Verbindung zum Ground zum 3.3V zum Anschluss D21 (auf unserem ESP-32 nur 21) und zum Anschluss D22(auf unserem ESP-32 nur 22).



Quelle

Danach haben wir einen Demo-Code von dem Hersteller vom Temperatursensors benutzt, um es zu testen. Diesen Code kann man benutzen, wenn man zuerst die Library «Adafruit AHTX0» installiert unter dem Menu Tools->Manage Libraries. Danach kann man unter File → Examples → Adafruit AHTX0 → adafruit_ah_test den Demo Code auswählen. Ein neues Fenster wird geöffnet und man muss dann nur noch den Code zum ESP-32 uploaden.



Quelle

Probleme:

Problem	Begründung	Lösung
Kabel im falschen Pin eingesteckt	Da unser ESP-32 Board andere Namen hat für ihre Anschlüsse, gab es ein Problem als ich einen Code vom Internet kopierte, da es für den Pin einen anderen Namen hatte.	Pin auf dem ESP-32 gewechselt und Namen im Code gewechselt.
Falsche Baudrate	Ein weiteres Problem vom Kopieren vom Internet war, dass im Code eine andere Baudrate eingegeben wurde.	Baudrate von 9600 zu 115200 gewechselt.
Topic Name nicht überein mit code	Im Demo Code vom Dokument 02 habe ich vergessen meine eigenen Topic Namen im Code einzufügen, dies führte dazu, dass es im Node-RED angezeigt hat, dass es verbunden ist, aber dennoch keine Daten bekommen hat.	Eigene Topic Namen im Demo Code einfügen

2.3 Reflexion

Heute konnte ich zum ersten Mal ein ESP-32 richtig benutzen. Wir hatten zwar am Anfang vom Tag ziemlich viel Theorie, aber es hat sich für mich gelohnt, da ich, als ich mit dem ESP-32 startete, auch verstanden habe, wieso ich es genau so macht und wie alles funktioniert. Es hat sehr geholfen, dass ich jetzt den Theorieteil, den ich gestern gehabt habe, einsetzen konnte, mit den ESP-32 Boards. Es hat mir geholfen, den Theorieteil von gestern und heute besser zu verstehen. Die Videos für MQTT habe ich auch sehr gut gefunden, weil ich glaube, das ich mehr Aufmerksam bin, wenn ich ein Video schaue. Nachdem ich das dritte Problem in der Tabelle gelöst habe, habe ich bemerkt, dass es für mich doch nicht so kompliziert ist und ich auch Spaß habe daran zu schaffen.

3 Tag 3

3.1 Tagesrückblick

Tagesrückblick vom Freitag, 18.11.2022
<ul style="list-style-type: none">• Tageseinstieg mit Andrew• Hilfe Dokument zum Fachgespräch gelesen und gewisse Themen recherchiert• Experimentiert mit ESP-32 und verschiedene Sensoren• Lernjournal geschrieben

3.2 Wissenssammlung

Heute haben wir den ESP-32 mit einer LED und einem Knopf verknüpft und haben es so programmiert, dass der LED eine zufällige Farbe zeigt, wenn man den Knopf drückt.

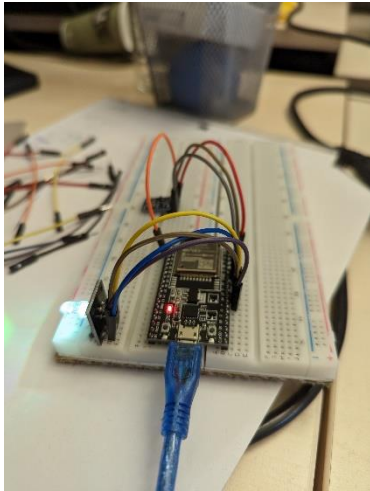
Zuerst muss man die LED auf dem Breadboard verbinden. Danach kann man diesen Code kopieren und in der Arduino IDE einsetzen.

```
1  int red = 11;
2  int green = 10;
3  int blue = 9;
4
5  void setup() {
6    pinMode(red, OUTPUT);
7    pinMode(green, OUTPUT);
8    pinMode(blue, OUTPUT);
9  }
10
11 void loop() {
12   analogWrite(red, random(0, 255));
13   analogWrite(green, random(0, 255));
14   analogWrite(blue, random(0, 255));
15   delay(1000);
16 }
```

Quelle

Quelle:<https://create.arduino.cc/projecthub/gol73/innovation-lab-4-rgb-led-random-color-change-0bc371>

Wenn man den Code uploadet, sollte das Licht anschalten und sollte jede Sekunde eine neue Farbe anzeigen.



Quelle

Damit man die RGB-Zahlen per MQTT verschicken kann, um es auf Node-RED zu sehen. Muss man den Code vom PDF [02_modul-216-ku-esp32-node-red-mqtt_v1-4.pdf](#) Kapitel 4.1.2 bis Mitte Seite 12 zum vorherigen Code hinzufügen.

Ganz oben unter dem topic1 char sollte noch diese topics eongefügt werden:

```
const char* topic3 = "m216demo/ RGB/Red/in";
const char* topic4 = "m216demo/RGB/Green/in";
const char* topic5 = "m216demo//RGB/Blue/in";

//Wenn man den LED in andere Pins eingesteckt hat dann diese Zahl hier eingeben
const int redPin = 16;
const int greenPin = 17;
const int bluePin = 18;
```

Jetzt muss nur noch die Loop Funktion mit diesem Code ersetzt werden:

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  sensors_event_t humidity, temp;
  aht.getEvent(&humidity, &temp);
  char tempBuffer[10];
  sprintf(tempBuffer, "%f", temp.temperature);
  client.publish(topic1, tempBuffer);

  int red = random(0, 255);
  int green = random(0, 255);
  int blue = random(0, 255);

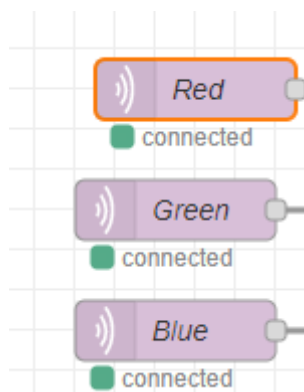
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

```
char red2[4];
char green2[4];
char blue2[4];

sprintf(red2, "%d", red);
sprintf(green2, "%d", green);
sprintf(blue2, "%d", blue);

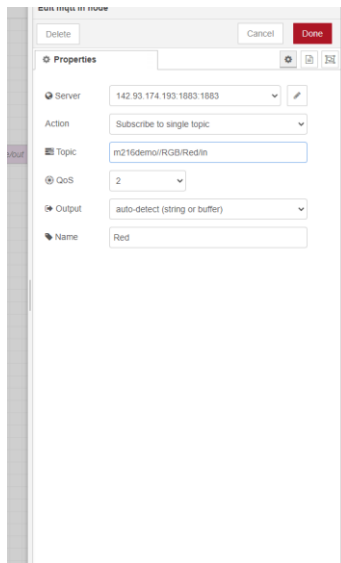
client.publish(topic3, red2);
client.publish(topic4, green2);
client.publish(topic5, blue2);
delay(100);
client.loop();
}
```

Wenn man diesen Code kopiert hat, sollte man zu Node-RED wechseln und dort 3 neue MQTT in Blöcke ins Field ziehen.



[Quelle](#)

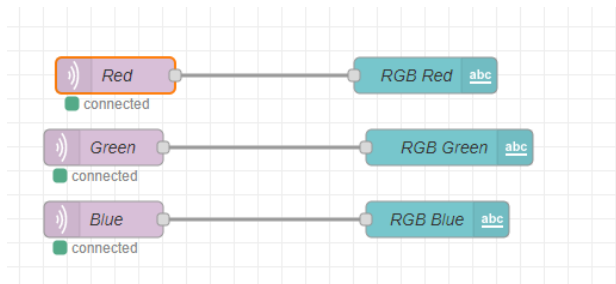
Wenn man diese Blöcke 2-mal anklickt, kann man die Einstellungen ändern und man kann im Topic Feld denn Topic Namen vom Code eingeben.



Quelle

Das sollte man mit den anderen 2 Topics auch machen, aber man muss aufpassen, dass man den richtigen Topic Namen eingibt. Diese findet man im Code unter topic3(Red), topic4(Green) und topic5(Blue).

Damit es im Node-RED UI einen Output gibt, muss man noch 3 Textfelder im Field ziehen. Die Einstellungen kann man frei ändern. Es ist egal, welchen Namen man ihm gibt.



Quelle

Jetzt sollten die RGB-Zahlen in der UI angezeigt werden.

Man kann jetzt den Knopf am Breadboard verknüpfen. Um den Knopf zum Funktionieren bringen, muss man folgende Code Abschnitte an gewisse Stellen einfügen:

```
const int buttonPin = 15;
```

Diesen Code Abschnitt kann man unter den redPin, greenPin etc. hintun. Die Zahle gleicht wieder die Zahle des Ports, an dem der Knopf verbunden ist.

```
int buttonState = digitalRead(buttonPin);
```

Das kann am Anfang von der Loop Funktion eingesetzt werden.

```
pinMode(buttonPin, INPUT_PULLUP);
```

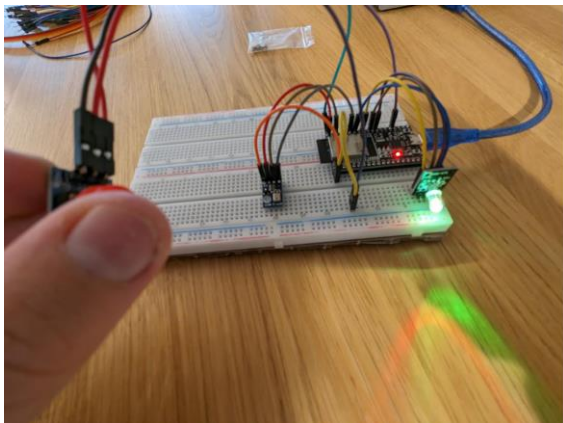
Dieser Code kann in der Setup Funktion eingefügt werden.

```
if (buttonState == HIGH) {
    analogWrite(redPin, red);
}
```

```
analogWrite(greenPin, green);  
analogWrite(bluePin, blue);  
  
char red2[4];  
char green2[4];  
char blue2[4];  
  
sprintf(red2, "%d", red);  
sprintf(green2, "%d", green);  
sprintf(blue2, "%d", blue);  
  
client.publish(topic3, red2);  
client.publish(topic4, green2);  
client.publish(topic5, blue2);  
}
```

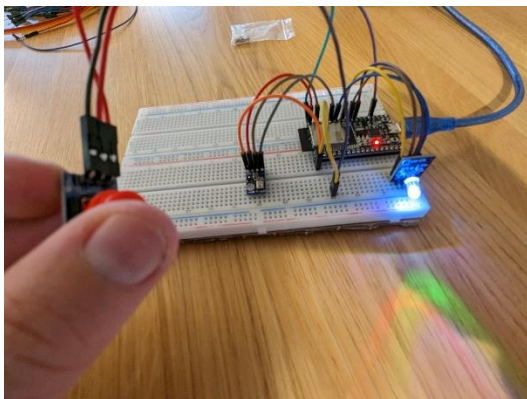
Zum Schluss kann man diesen Code-Abschnitt, mit dem alten, der gleich aussieht in der Loop Funktion ersetzt werden.

Wenn man alles richtig gemacht hat und den Code zum ESP-32 uploadet, wird es funktionieren und man kann mit dem Knopf die Farbe der LEDs wechseln.



Quelle

Vor dem Knopfdruck



Quelle

Nach dem Knopfdruck

Probleme:

Fehler	Begründung	Lösung
client.publish kann keine Ints übertragen	Client.publish kann nur Strings/char Arrays als Argument haben, also als ich probiert habe einen Int als Argument anzugeben, kam eine Fehlermeldung	Mit sprintf int zu char Array gewechselt
Fehler mit sprintf	Es wurden falsche Daten via MQTT gesendet. Es hat sich herausgestellt, dass ich denn char Array zu klein deklariert habe, sodass es denn \0 nicht drin haben konnte.	Char Array Größe von 3 auf 4 wechseln.

3.3 Reflexion

Heute war bisher den lustigsten Tag von dem üK, weil wir den ganzen Tag mit den verschiedenen Sensoren experimentieren konnten. Weil ich ohne bestimmte Aufträge arbeiten durfte, konnte ich viel mehr ausprobieren, was dazu führte, dass ich mehr lernen konnte und das, was ich schon gelernt habe, besser verstehen konnte. Es gab zwar immer wieder Probleme, aber als ich diese lösen konnte, habe ich immer mehr gelernt. Es war auch für mich spannend zu erfahren, welche Sensoren was machen. Ich freue mich sehr endlich mit dem Projekt anzufangen und hoffe, dass ich das, was ich heute gemacht habe, im Projekt anwenden kann.

4 Tag 4

4.1 Tagesablauf

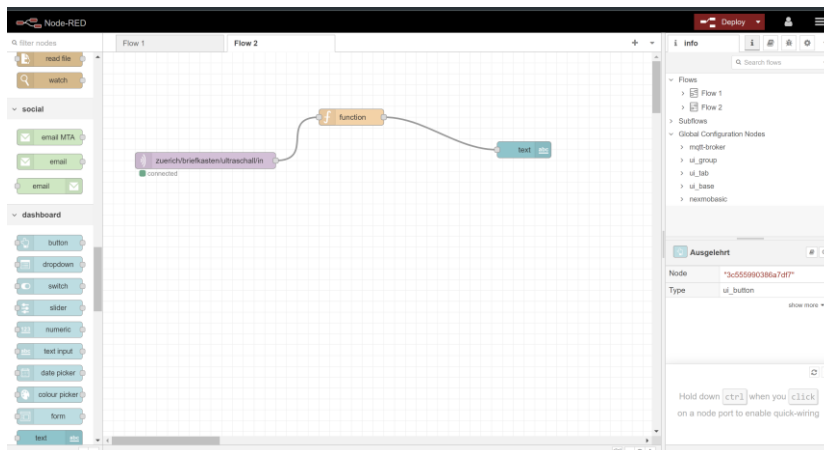
Tagesablauf vom Mittwoch, 23.11.2022
<ul style="list-style-type: none"> • Input mit Andrew zur Austeilung der Aufgaben • Den Auftrag «Briefkasten» gewählt • Informationen über das Thema gesammelt • Erste Demos auf dem Breadboard zusammengestellt • MQTT-Verbindung mit dem esp-32 hergestellt • Mit Node-RED eine UI gestaltet für das Projekt

4.2 Wissenssammlung

Im Node-RED eine Variable um eins erhöhen, wenn ein Sensor abgeht:

Da ich für dieses spezifische Thema sehr viel Zeit verbraucht habe, möchte ich hier eine Anleitung machen, damit ich es besser verstehe.

Wenn man im C oder in Java eine Variable um eins erhöhen will, dann würde man normalerweise eine Klassen- oder eine globale Variable definieren und diese dann immer um 1 erhöhen mit ++ oder +=1. Im Node-RED ist es aber ein bisschen komplizierter eine globale Variable zu definieren.



Quelle

Der Flow sieht zwar sehr simple aus, aber der Funktionsblock ist der Kern der Komplikation. Wenn man in der Funktion nämlich nur eine Variable definiert und sie nachher immer um 1 erhöht, wenn etwas ausgelöst wird, etwa so:

Name

Setup

On Start

On Message

On Stop

```

1 var input = msg.payload;
2 var output = 0;
3 if(input < 10 || input > 1100){
4   output++;
5 }
6
7 return {payload : output};

```

Quelle

dann wird es, jedes Mal, wenn eine neue Message vom MQTT gelesen wird, den Wert output auf 0 zurücksetzen. Um das zu verhindern, braucht man dann eine globale Variable. Die Funktion mit einer globalen Variable sollte so aussehen:

```

1 var input = msg.payload;
2 var count=global.get('count') || 0;
3 if(input < 10 || input > 1100){
4   count +=1;
5 }
6 msg.payload=count;
7 global.set('count',count);
8 return msg;
9

```

Quelle

In dieser Funktion wird die Variable count am Anfang nur auf 0 gesetzt, wenn es keine bereits definierte globale Variable namens count gibt. Wenn schon, dann wird count zu diesem Wert initialisiert. Und so kann es sich immer um 1 erhöhen.

Probleme:

Problem	Begründung	Lösung
Variable um 1 erhöhen	Siehe Wissenssammlung	Siehe Wissenssammlung
Arduino IDE konnte keine Verbindung zum esp-32 herstellen	Als ich probiert habe meinen Code auf dem esp-32 hochzuladen kam im CMD immer, dass es keine Verbindung herstellen konnte.	Arduino IDE neugestartet.

4.3 Reflexion

Heute haben wir mit unserem Projekt angefangen. Wir haben uns für den Briefkasten Projekt entschieden und obwohl ich am Anfang nicht so glücklich war mit dem Projekt, finde ich es jetzt doch noch ein großartiges Projekt. Ich habe mich im Projekt mit Node-RED beschäftigt und ich habe es sehr mühsam gefunden. Der Grund dafür ist das alles im Node-RED für mich 10x komplizierter scheint als z.B. im C oder Java. Ein Beispiel ist, dass ich, um eine Variable immer um 1 zu erhöhen, etwa 1.5 Stunden gebraucht habe, um es zu herausfinden. Ich glaube aber, dass alle andere Optionen ausser Node-RED wahrscheinlich schlimmer sein würden.

5 Tag 5

5.1 Tagesablauf

Tagesablauf vom Donnerstag, 24.11.2022
<ul style="list-style-type: none"> • Für das Fachgespräch gelernt • Erster Test beim Briefkasten Noser Young gemacht • Fachgespräch gehabt • Zweiter Test beim Briefkasten Noser Young gemacht • Projektdokumentation geschrieben

5.2 Wissenssammlung

E-Mail per Node-RED verschicken:

Mit der Extension „node-red-node-email“ kann man im Node-RED unter spezifischen Bedingungen eine E-Mail verschicken. Die Bedingung, die Ich gewählt habe, ist, dass es nur eine

E-Mail verschickt, wenn meine count Variable von der Wissenssammlung von Mittwoch um 1 erhöht wird.

Damit man eine E-Mail verschicken kann muss man diese Felder ausfüllen:

✉ To	<input type="text" value="maxmuster@gmail.com"/>
🌐 Server	<input type="text" value="smtp.gmail.com"/>
🔌 Port	<input type="text" value="465"/> <input checked="" type="checkbox"/> Use secure connection.
👤 Userid	<input type="text" value="maxmuster@gmail.com"/>
🔒 Password	<input type="password" value="....."/>
🔒 TLS option	<input checked="" type="checkbox"/> Check server certificate is valid
👤 Name	<input type="text" value="Name"/>

Quelle

To: Das To Feld beinhaltet die E-Mail der Person, an dem man die E-Mail verschicken.

Server: Das Server Feld beinhaltet den SMTP Server Name. Bei Gmail ist es „smtp.gmail.com“.

Port: Das Port Feld beinhaltet den Port des SMTP-Servers. Bei Gmail ist es 465.

Userid: Dieses Feld beinhaltet die E-Mail des Senders.

Password: Dieses Feld beinhaltet das App-Passwort des Senders. Um dieses Passwort zu bekommen, muss man in den Google Account Einstellungen unter Sicherheit ein App-Passwort erstellen. Um diese Einstellung zu sehen, muss man 2-Faktor Authentifizierung aktivieren.

Wenn man dann dieses Passwort bekommt, kann man es in diesem Feld einsetzen.

Name: Dieses Feld ist optional. Es beinhaltet der Name des Nodes. Wenn es leer gelassen wird, dann wird der Node mit der E-Mail vom „To“ Feld benannt.

Wenn man diese Felder richtig ausfüllt dann sollte eine E-Mail verschickt werden mit dem Inhalt vom msg.payload.

5.3 Reflexion

Heute haben wir unser Gerät getestet. Ich habe es spannend gefunden das Gerät richtig im Briefkasten zu sehen und es auch auszutesten. Ich hoffe wir werden eine gute Note bekommen, weil ich sehr viel und hart gearbeitet habe und es wäre gut, wenn es sich auch gelohnt hat. Heute habe ich auch ein Teil der Projektdokumentation geschrieben. Ich habe es zwar nicht gerne und finde es mühsam, aber ich weiss, dass es notwendig ist.

6 Tag 6

6.1 Tagesablauf

Tagesablauf vom Freitag, 25.11.2022

- Projektdokumentation geschrieben
- Lerndokumentation geschrieben

6.2 Wissenssammlung

Da wir heute nur unsere Dokumentationen fertig geschrieben haben, habe ich nichts neues gelernt, was ich in der Wissenssammlung hinschreiben kann.

6.3 Reflexion

Heute war ein wichtiger, aber für mich sehr langweiliger Tag. Der Grund dafür, ist, dass wir heute nur die Dokumentationen fertig geschrieben haben. Heute ist der letzte Tag des üKs und ich habe den üK im Ganzen sehr spannend und lustig gefunden. Ich finde solche Projekte, wo man allein gelassen wird und nicht immer Inputs anhören muss sehr erleichternd, bis man die Dokumentationen schreiben muss. Ich habe aber auch gefunden, dass wir in der zweiten Woche für mich fast zu viel schreiben mussten. Ich hatte immer fast keine Zeit mehr meine Lerndokumentation zu schreiben und musste es manchmal zu Hause fertig schreiben. Ich finde die Lerndokumentation schon wichtig, aber wenn ich dann auch noch eine Projektdokumentation schreiben muss, dann wird es für mich zu viel.

7 Quellenverzeichnis

Seite 3: IoE Bild

Quelle: <https://www.ilovephd.com/wp-content/uploads/2019/03/16c58-ioe.png>

Seite 4: Sicherheitsrisiken bei IoE und IoT Geräte Bild

Quelle: <https://unit42.paloaltonetworks.com/iot-threat-report-2020/>

Seite 5: MQTT Broker Bild

Quelle: <https://www.spiceworks.com/tech/iot/articles/what-is-mqtt/>

Seite 6: MQTT Broker Cluster Bild

Quelle: <https://www.emqx.io/docs/en/v3.0/>

Seite 6: Connect Package Bild

Quelle: <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>

Seite 7: Anschluss von esp-32 und Temperatursensor Bild

Quelle: Noser Young 00_modul-216-ku-getting-started_v1-4.pdf

Seite 7: Output vom Serial Monitor

Quelle: Filip Franeks iPhone

Seite 7-17: Den Rest der Bilder sind entweder von meinem Handy gemacht worden oder sie sind Screenshots.