

Projektdokumentation Ük 216

Modul: 216

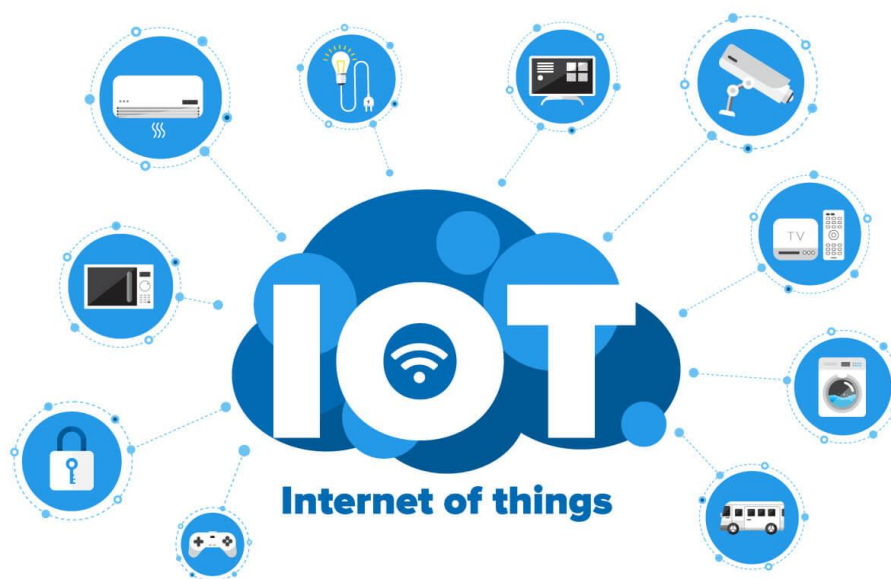
Thema: Internet of Everything-Endgeräte in bestehende Plattform integrieren

Projekt: Briefkasten

Eine Arbeit von: Jamie Giger

Kontakt: jamie.giger@noseryoung.com

Art: Projektdokumentation



Inhaltsverzeichnis

Inhalt

Ausgangslage.....	3
Planen	3
Mein Vorgehen und Resultate	4
Tag1:	4
Tag2:	4
Tag3:	5
Tag4:	5
Testverfahren	6
Test 1:	6
Test2:	6
Test3:	6
Testingprotokoll	7
Verbindung der Komponenten.....	8
Resultat	8

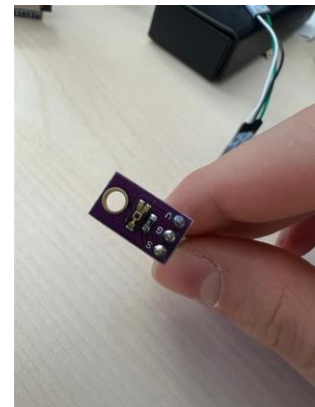
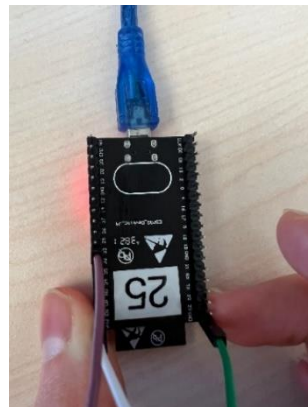
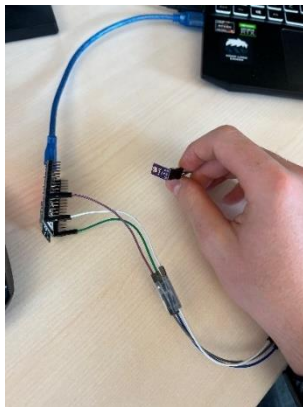
Ausgangslage

Ich habe das Projekt „Briefkasten“ erhalten oder besser gesagt ausgesucht. Um was geht es? Es geht darum, dass ich einen Lichtsensor an den esp32 anschliesse und dieser erfassen kann, ob der Briefkasten offen oder zu ist. Er muss das ganze erfassen und die Daten müssen dann im Node-RED ersichtlich sein. Es soll automatisch, wenn der Briefkasten offen ist eine Mitteilung an eine gewisse Person schicken, dass er nun offen ist.

Da ich das Ganze allein umsetzen musste, hatte ich nicht die Zeit, um alles zu machen, ich habe es so gut probiert, wie es irgendwie ging.

Was habe ich verwendet?

Ich habe ein esp32 verwendet in der neuen Version. 3 Male, 6 Female und 3 Female/Male Kabel, sowie einen Lichtsensor. Natürlich noch das USB-A zu Mikro USB. 4 Kabel aneinander machen, Female zu Male zu female zu male, female. Danach den Lichtsensor nehmen und die Kabel befestigen. Auf dem Board haben wir den Slot, 33 mit dem auf dem Lichtsensor out slot verbunden, den 3v3 slot mit dem vcc und den GND mit dem GND auf dem Lichtsensor.



Planen

Ich habe mir überlegt, dass ich beginnen mit dem esp32 und ab Donnerstag mit der Projektdokumentation beginne. Ich habe so begonnen den Lichtsensor mit dem Mikrocontroller zu verbinden mit insgesamt 12 Kabel. Ich wollte, dass ich bis Ende Donnerstag, die Grundfunktionen habe und am Freitag noch das mit dem Versenden von Nachrichten machen kann. Mein Ausblick war nicht so großartig, da ich wusste, da ich alles allein machen muss, darum habe ich mir kein Ziel gesetzt, da es sehr anstrengend und schwierig ist, eine Aufgabe, die für 3 Personen ist, alleine zu machen.

Mein Vorgehen und Resultate

Tag1:

Wie schon vorhin erwähnt habe ich, mit dem Code begonnen, sodass der Mikrocontroller das Licht messen kann mit einem Code, der misst, ob das Licht über 15 ist oder unter 15. Das ist wichtig für den Briefkasten, ob er offen ist, wenn es über 15 oder ob er zu ist unter 15. Das war mein erster Schritt, um ein gewisser Grundcode zu haben, um darauf weiterzuschreiben.

```
if (analogValue < 15 && isopen == 1) {  
  Serial.println(" => letter box closed");  
  isopen = 0;  
} else if (analogValue > 15 && isopen == 0) {  
  Serial.println(" => letter box open");  
  isopen = 1;  
}
```

Tag2:

Der zweite Schritt war es das in einen Loop zu bringen ich habe mich für ein while-Loop entschieden und habe den dann wiederum ein void Loop gepackt. Mit dem lightbuffer konnte ich die Daten ins Node-RED veröffentlichen und somit hat der Code für die Funktion schon mal funktioniert.

```
void loop() {  
  int isopen = 0;  
  while (true) {  
    if (!client.connected()) {  
      reconnect();  
    }  
    // reads the input on analog pin (value between 0 and 4095)  
    int analogValue = analogRead(LIGHT_SENSOR_PIN);  
  
    Serial.print("Analog Value = ");  
    Serial.print(analogValue); // the raw analog reading  
  
    // We'll have a few thresholds, qualitatively determined  
    if (analogValue < 15 && isopen == 1) {  
      Serial.println(" => letter box closed");  
      isopen = 0;  
    } else if (analogValue > 15 && isopen == 0) {  
      Serial.println(" => letter box open");  
      isopen = 1;  
    }  
    char lightBuffer[10];  
    sprintf(lightBuffer, "%d", analogValue);  
    client.publish(topic1, lightBuffer);  
  
    delay(1000);  
  }  
}
```

Tag3:

Der dritte schritt war es den esp32 mit dem Internet zu verbinden, ich hatte am Anfang etwas Probleme, da ich nicht wusste, welches Stück Code ich aus Teams brauchte. Ich hatte zum Glück Dominic zur Hilfe, er hat mir gezeigt, welche Codes ich brauche und ich konnte dann damit etwas anfangen, leider hat es immer noch nicht funktioniert, somit habe ich Andrew gefragt und er hat mir das ganze erklärt und einen Trick gezeigt (mehr dazu im Testverfahren).

```
Code-Briefkasten.ino
22
23 WiFiClient espClient;
24
25 PubSubClient client(espClient);
26
27
28
29 void setup_wifi() {
30
31   Serial.print("Connecting to ");
32   Serial.print(ssid);
33   WiFi.begin(ssid);
34   while (WiFi.status() != WL_CONNECTED) {
35     delay(500);
36     Serial.print(".");
37   }
38   Serial.println("done!");
39 }
40
41 void callback(char* topic, byte* payload, unsigned int length) {
42   char message[length];
43   for (int i = 0; i < length; i++) { message[i] = (char)payload[i]; }
44   Serial.print("Received Message: ");
45   Serial.println(message);
46 }
47
48
49
50
51
52
53
54
55
56
57
58
59 void reconnect() {
60   Serial.print("Attempting MQTT connection...");
61   while (!client.connected()) {
62     if (client.connect("esp8266")) {
63       Serial.println("done!");
64       client.subscribe(topic1);
65       client.subscribe(topic2);
66     } else {
67       delay(5000);
68       Serial.print(".");
69     }
70   }
71 }
72 }
```

Tag4:

Ich habe vor allem an der Produktdokumentation und Lerndokumentation geschrieben und das Projekt an sich etwas vernachlässigt, da ich einfach keine Zeit für alles hatte.

Testverfahren

Test 1:

Mein erster Test, war nach dem Erstellen des Grundcodes. Ich habe den Loop gemacht und habe das noch ohne Internetzugriff getestet. Es hat zum ersten Mal nicht funktioniert, weil die Kabel falsch eingesteckt waren und somit die Erdung nicht an der richtigen Stelle war, und somit hatte er kein Strom. Ich habe dann die Kabel so umgesteckt, dass es richtig war und es hat, funktioniert.

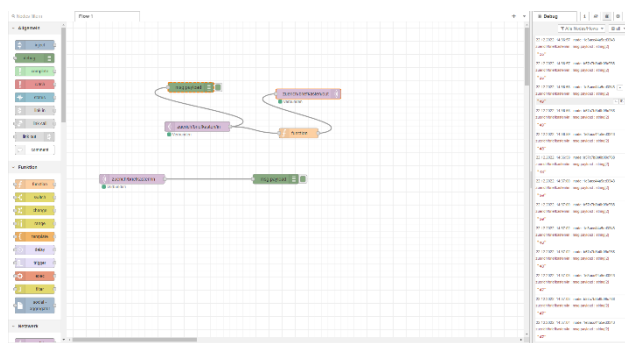
Test2:

Mein zweiter Test war nach der Implementierung des Internetzugriffes. Ich habe die vorgegebenen Codes von Teams eingefügt und noch etwas auf meinen Code angepasst. Ich hatte Probleme, weil es einfach nicht funktioniert, hat auch mit Dominics Hilfe und wir haben verschiedene Ding probiert, aber Saad kam danach auch noch und hat mir geholfen. Das Problem war, dass gewisse Pins abschalten, wenn man den esp32 mit dem Internet verbindet und ich hatte einen Pin auf einem dieser Pins, die abschalten und somit konnte er die Signale nicht mehr empfangen. Ich musste somit im Programm, den Port von 26 auf 33 ändern. Danach konnte er sich verbinden und ich habe auch alle gemessenen Daten im serial Monitor erhalten.

```
Code-Briefkasten.ino
1
2 #include <WiFi.h>
3
4 #include <PubSubClient.h>
5
6 #define LIGHT_SENSOR_PIN 33
7
8
```

Test3:

Der dritte Test war, dass ich mit dem Handy geschaut habe, ob der Sensor wirklich erkennen kann, wann der Briefkasten aufgeht und wann nicht, mit dem Erkennen des „Flashlight“. Das hat funktioniert, danach habe ich im MQTT-Explorer geschaut, ob es die Daten auch bekommt, und das hat es. So hatte ich die Resultate auf dem MQTT- Explorer und konnte auch sehen, wann der Briefkasten offen sein könnte und wann nicht.



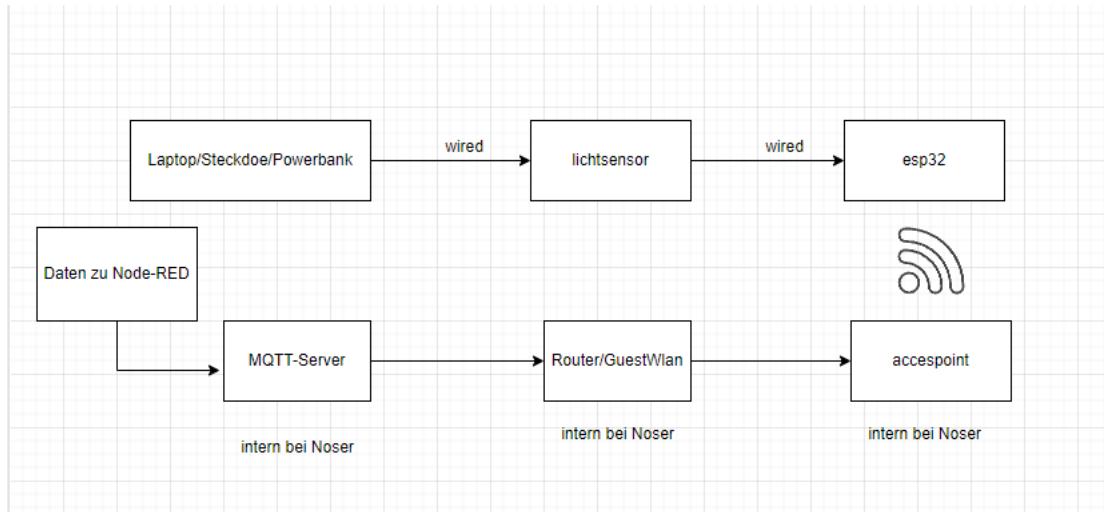
Link zum Node-
RED:

<http://ioe.noseryoung.ch:1887/>

Testingprotokoll

Was wird getestet?	Resultat	verbessern
Test1 Grundcode testen, ob der Sensor erkennen kann, wie fest das Licht leuchtet	- Er kann es erkennen, auch mit den gewünschten Anpassungen	-
Test2 Implementierung des Wi-Fi Zugriffs.	- Es hat nicht funktioniert, Fehlermeldung auch durch Anpassungen	Ich habe Hilfe von Dominic, Saad und Andrew erhalten und es hat Schluss endlich funktioniert, die Lösung war, dass ich ein Stück Code vergessen und den falschen Port verwendet habe
Test3 Mit Handy und „Flashlight“ geschaut, ob er es erkennt und auch es auch bei Node-RED ankommt	- Es hat funktioniert, es zeigt einfach nur die Lichtstärke an und nicht, ob der Briefkasten offen oder zu ist, man es herleiten. - Unter 15 = gleich zu darüber offen	-

Verbindung der Komponenten



Hier sieht man, wie die Daten von meinem esp32 zu Node-RED kommen. Es beginnt bei einem Gerät, das Strom geben kann, in meinem Fall war es ein Laptop. Der Strom geht via Mikro-USB Kabel an den esp32, welcher mit Kabel mit dem Lichtsensor verbunden ist und ihm Strom gibt. Der esp32 verbindet sich mit einem AccesPoint mit dem Guest Wi-Fi und dieser esp32 kann sich danach mit einem MQTT-Server verbinden, der die Daten an Node-RED weiterleitet.

Resultat

Mein Lichtsensor kann erkennen, wann ein Briefkasten aufgeht und wann er wieder zu geht. Er sendet diese Daten an den MQTT Explorer und der an Node-RED. Man sieht Datum und Zeit der Übermittlung und kann dort auch sehen, wie hoch der Lichteinfang ist. Ich bin nicht fertig geworden, da ich die ganze Arbeit allein machen musste, aber ich denke ich bin so weit gekommen wie ich konnte. Es hat aber trotzdem Spass gemacht, da ich komplett auf mich gestellt war und eigentlich die Arbeit von Grund auf selbst planen und umsetzen musste. Ich habe alles in allem trotzdem viel gelernt.