Bern, 12.01.2022

ÜK 216 IoE-Projekt

Gruppen-Dokumentation

Luca Glogger und Silas Rochemuttu

Dokumentinformation

|  |  |  |
| --- | --- | --- |
| Erstelldatum | 12.01.2022 | |
| Autoren | Luca Glogger und Silas Rochemuttu | |
| Coach | Sandro Rüfenacht | |
| Dateiname | UEK\_216\_Projekt\_Silas\_Luca.docx | |
| Versionen | 1.0 | Erstellt am – 12.01.2022 |
|  |  |

Inhaltsverzeichnis

[1. Dokumentinformation 2](#_Toc94105490)

[3. Einleitung 4](#_Toc94105491)

[4. Übersicht der Komponenten: 5](#_Toc94105492)

[5. Pins - CCS811 8](#_Toc94105493)

[Power Pins: 8](#_Toc94105494)

[Logic pins: 8](#_Toc94105495)

[Other Pins: 8](#_Toc94105496)

[6. Pins - Si7021 9](#_Toc94105497)

[Power Pins: 9](#_Toc94105498)

[I2C Logic pins: 9](#_Toc94105499)

[7. Pins - SparkFun ESP32 Thing 10](#_Toc94105500)

[Verbinden der Komponenten (Checkliste vom 12.01.2022) 11](#_Toc94105501)

[8. Unser Vorgang 12](#_Toc94105502)

[Unsere Ziele 12](#_Toc94105503)

[Der Anfang 12](#_Toc94105504)

[Unser Gerät mit Node-RED verbinden 12](#_Toc94105505)

[Zwischenstand 12](#_Toc94105506)

[Node-RED 13](#_Toc94105507)

[Zugriff auf TRADFRI Steckdose 14](#_Toc94105508)

[Automatische Anschaltung des Ventilators 15](#_Toc94105509)

[9. Datenfluss aller Komponenten: 16](#_Toc94105510)

[10. Testen der Verknüpfungen 17](#_Toc94105511)

[11. Testing 18](#_Toc94105512)

[Was braucht man alles: 18](#_Toc94105513)

[Testziele: 18](#_Toc94105514)

[Testobjekte 18](#_Toc94105515)

[Testinfrastruktur 18](#_Toc94105516)

[Testorganisation 18](#_Toc94105517)

[12. Arbeitsaufteilung 21](#_Toc94105518)

[13. Ausgangslage 22](#_Toc94105519)

[14. Quellenverzeichnis 23](#_Toc94105520)

[Webseiten: 23](#_Toc94105521)

[Dokumente: 23](#_Toc94105522)

[15. Abbildungsverzeichnis 24](#_Toc94105523)

Einleitung

Dies ist unsere Gruppen-Dokumentation zu unserem Projekt des ÜK 216 (Internet of Everything-Endgeräte in bestehende Plattform integrieren).

Übersicht der Komponenten:

|  |  |  |
| --- | --- | --- |
| <https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor/pinouts> | Adafruit CCS811 Air Quality Sensor | Dieser kleine Gassensor erkennt und überwacht die Luftfeuchtigkeit in unseren Innenräumen. |
| <https://learn.adafruit.com/adafruit-si7021-temperature-plus-humidity-sensor> | Adafruit Si7021 Temperature + Humidity Sensor | Wie man im Namen schon hört, überwacht dieser Sensor die Temperatur sowie auch die Feuchtigkeit. |
| <https://www.exp-tech.de/plattformen/internet-of-things-iot/7828/sparkfun-esp32-thing> | Sparkfun ESP32 Thing | Man könnte es schon fast als Mainboard bezeichne. Auf diesem Embedded System werden alle Daten gesammelt und können auch weitergeleitet werden. |
| <https://www.ikea.com/ch/de/p/symfonisk-tradfri-gateway-set-ton-weiss-weiss-s19319164/> | IKEA SYMFONISK | Dies ist das Gateway, über welchen wir auch die TRADFRI Steckdosen zugreifen können. |
| <https://www.ricardo.ch/de/a/speaker-summer-fuer-pc-mainboards-neu-1193381784/> | Speaker Summer | Hiermit können laute und leise Töne ausgegeben werden. |

|  |  |  |
| --- | --- | --- |
| <https://www.ikea.com/ch/de/p/tradfri-steckdose-funkgesteuert-00473650/> | IKEA TRADFRI | Die ist unsere Steckdose, über welche wir dann unser Endgerät ansprechen können. |
| <https://ch.elv.com/elv-hochwertiges-steckkabel-set-stecker-auf-stecker-10-stueck-in-verschiedenen-farben-100-mm-251224> | ELV Steckkabel | Natürlich müssen wir auch alle Things mit unserem Embedded System verbinden. Diese Verbindung erflogt über die ELV Steckkabel. |
| <https://www.komforthaus.com/Elektronik/Zubehoer/Breadboard> | Breadboard Steckbrett | Alle Things sowie das Embedded System und die ELV Steckkabel werden in dieses Steckbrett eingebunden und es kann somit eine Verknüpfung der Komponenten stattfinden. |
| <https://www.kaiserkraft.ch/klimageraete/ventilatoren/standventilator-hoehenverstellbar/> | Ventilator | Dies ist unser Endgerät, welches wir über die TRADFRI Steckdose ein- sowie ausschalten können. |

|  |  |  |
| --- | --- | --- |
| Arduino Software (IDE) Download – kostenlos – CHIP  <https://www.chip.de/downloads/Arduino-Software-IDE_90789714.html> | Arduino | Auf diesem Programm können wir ein Skript schreiben, welches dann auf das Embedded System geladen werden kann. |
| Einstieg in Node-RED | Elektor Magazine  <https://www.elektormagazine.de/articles/einstieg-in-nodered> | Node-RED | Auf dem Node-RED können wir eine grafische Oberfläche erstellen, auf welcher die Daten visuell ausgelesen werden können. |

Pins - CCS811

|  |  |
| --- | --- |
| CCS811  <https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor/pinouts> | Power Pins: Vin – Dies ist der Stromanschluss. Da der Sensor 3,3V verwendet, haben wir einen integrierten Spannungsregler eingebaut, der 3-5VDC abnimmt und sicher herunterkonvertiert. Um die Platine mit Strom zu versorgen, geben Sie ihr die gleiche Spannung wie der Logikpegel Ihres Mikrocontrollers - z.B. für einen 5V-Mikrocontroller wie Arduino, verwenden Sie 5V.  3Vo – dies ist der 3.3V Ausgang des Spannungsreglers, von dem man bis zu 100mA abgreifen kann, wenn man möchte.  GND – gemeinsame Basis für Strom und Logik. (Er ist wie – und 3v3 wie + somit kann man ein kleines Stromnetz bauen) Logic pins: SCL – Dies ist der I2C-Takt-Pin, der an die I2C-Taktleitung Ihres Mikrocontrollers angeschlossen wird. Es gibt einen 10K Pullup an diesem Pin und er ist pegelverschoben, so dass Sie 3 - 5VDC verwenden können.  SDA – Dies ist der I2C-Daten-Pin, der an die I2C-Datenleitung Ihres Mikrocontrollers angeschlossen wird. Es gibt einen 10K Pullup an diesem Pin und er ist pegelverschoben, so dass Sie 3 - 5VDC verwenden können. Other Pins: INT – Dies ist der Interrupt-Ausgangs-Pin. Es handelt sich um eine 3-V-Logik, mit der Sie erkennen können, wann ein neuer Messwert bereit ist oder wann ein Messwert zu hoch oder zu niedrig wird.  WAKE – Dies ist der Wakeup-Pin für den Sensor. Er muss auf Masse gezogen werden, um mit dem Sensor zu kommunizieren. Dieser Pin ist pegelverschoben, so dass Sie 3-5VDC Logik verwenden können.  RST – Dies ist der Reset-Pin. Wenn er auf Masse gezogen wird, setzt sich der Sensor selbst zurück. Dieser Pin ist pegelverschoben, so dass Sie 3-5VDC Logik verwenden können. |

Pins - Si7021

|  |  |
| --- | --- |
| Si7021    <https://learn.adafruit.com/adafruit-si7021-temperature-plus-humidity-sensor> | Power Pins: Vin Dies ist der Power-Pin. Da der Chip 3 VDC verwendet, haben wir einen Spannungsregler auf der Platine, der 3-5 VDC aufnimmt und sicher herunterkonvertiert. Um die Platine mit Strom zu versorgen, geben Sie ihr dieselbe Spannung wie dem Logikpegel Ihres Mikrocontrollers - z.B. für einen 5V-Mikro wie Arduino, verwenden Sie 5V  3v3 - dies ist der 3.3V Ausgang des Spannungsreglers, von dem man bis zu 100mA abgreifen kann, wenn man möchte (+ Strom gnd = - Strom).  GND - gemeinsame Basis für Strom und Logik. I2C Logic pins: SCL - I2C-Takt-Pin, verbinden Sie ihn mit der I2C-Taktleitung Ihres Mikrocontrollers.  SDA - I2C-Datenpin, verbinden Sie ihn mit der I2C-Datenleitung Ihres Mikrocontrollers. |

Pins - SparkFun ESP32 Thing

Die wichtigsten Funktionen dieser Pins wurden auf den vorherigen zwei Seiten beschrieben.

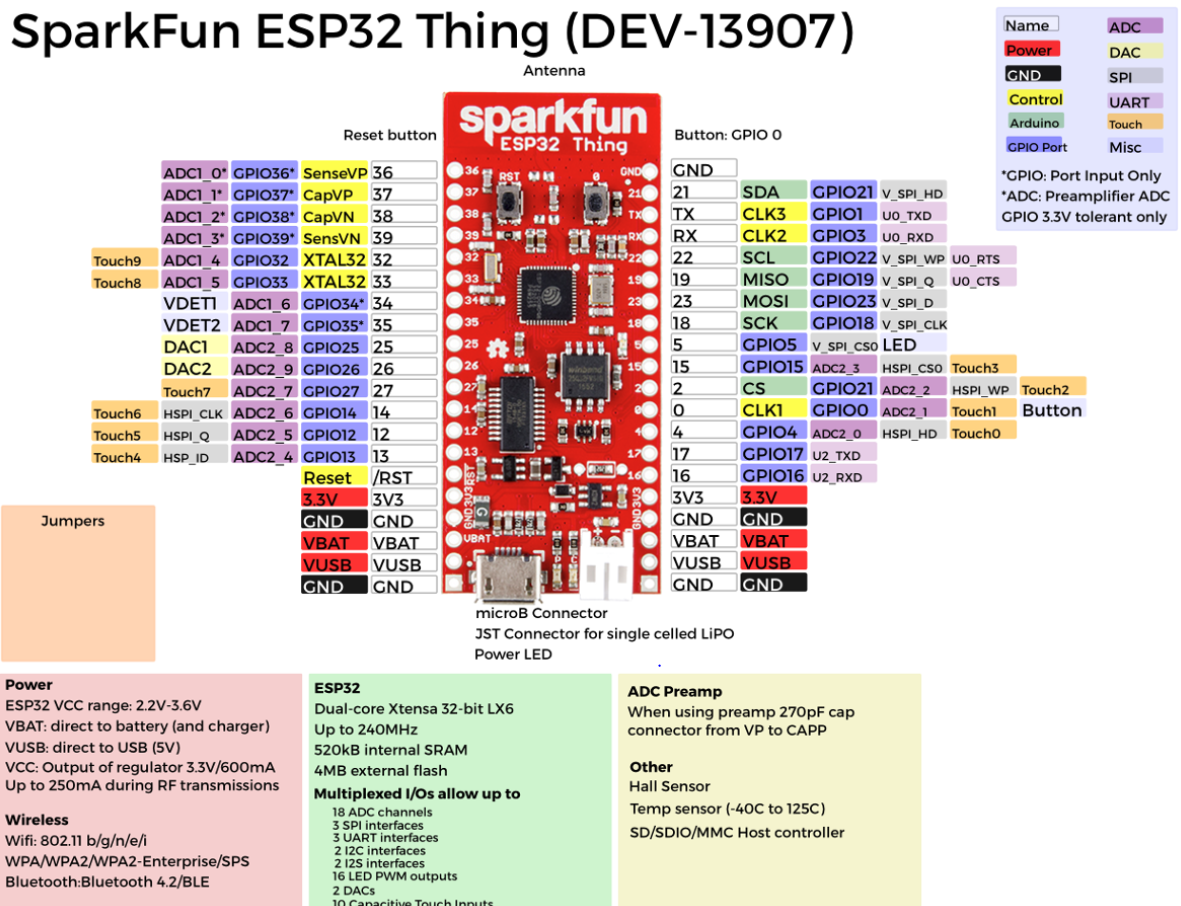


Abbildung 1 ESP32 Ports

Verbinden der Komponenten (Checkliste vom 12.01.2022)

|  |  |  |  |
| --- | --- | --- | --- |
| ESP32 Thing | SI7021 | CCS811 | Richtig Verbunden? |
| 21 | SDA | SDA | ✓ |
| 22 | SCL | SCL | ✓ |
| GND | - | WAKE | ✓ |
| GND | GND | GND | ✓ |
| 3V3 | 3Vo | 3V3 und VIN | ✓ |

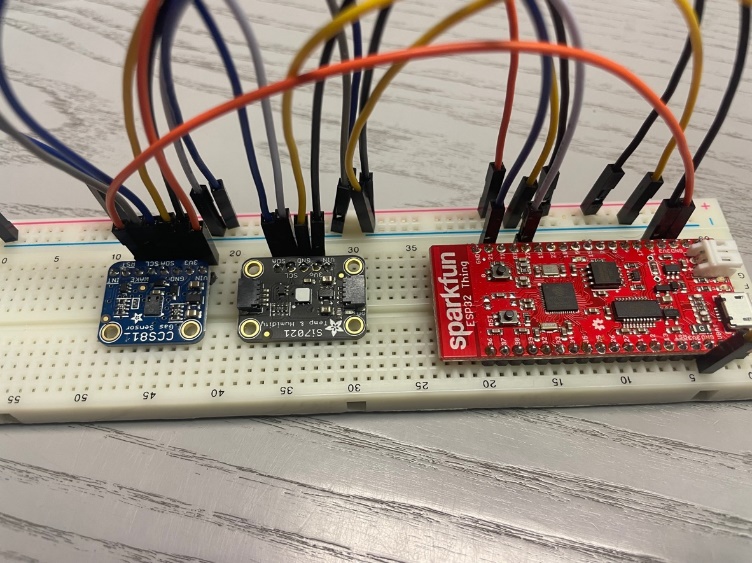


Abbildung 2 Unsere Komponenten

Unser Vorgang

# Unsere Ziele

Wir wollen ein IoT-Gerät entwickeln, welches die Temperatur, den CO2 Wert und die Luftfeuchtigkeit in unserer Umgebung messen und ausgeben kann. Dazu werden wir dieses Gerät mit den nötigen Komponenten verkabeln und unser Thing in eine Plattform einbinden, auf welcher wir die gemessenen Werte grafisch darstellen lassen können. Dazu soll ein Ventilator angehen, wenn die Temperatur in unserem Raum über eine gewisse Grenze ansteigt.

# Der Anfang

Als erstes haben wir alles verkabelt, das heisst dass Sparkfun ESP32 Thing mit dem SI7021, CCS811 und einem Speaker Summer verbunden. Anschliessend haben wir unsere Konstruktion an einem Laptop angeschlossen und fingen an, auf Arduino ein kleines Skript zu schrieben.

# Unser Gerät mit Node-RED verbinden

Über den Link localhost:1880 kamen wir auf die Node-RED Applikation, auf der wir unter der Kategorie Netzwerk den «mqtt in» auswählen mussten. Somit konnten wir einen Server, eine Action, einen Topic etc. festlegen, worüber wir eine Verbindung mit unserem Geräte herstellen konnten. Natürlich mussten wir es jetzt noch schaffen, dass auch die richtigen Daten übertragen werden. Zu beginn übertrugen wir erst ein Wort, nämlich «Hallo Welt!», als dieses erfolgreich ankam konnten wir da drauf aufbauen.

# Zwischenstand

Zum jetzigen Zeitpunkt haben wir unser gerät mit den benötigten Sensoren verknüpft damit wir sämtliche Daten auslesen können. Wir haben bereits das Skript auf der Plattform Arduino geschrieben und es auf unserem IoT-Gerät abgespeichert. Nebenbei haben wir eine erfolgreiche Verbindung zu Node-RED hergestellt, wodurch wir jetzt an dem Front-End arbeiten können.

# Node-RED

Jetzt mussten wir es nur noch hinbringen, dass die Daten wie Temperatur oder auch Luftfeuchtigkeit etc. auf unserem Node-RED Dashboard ersichtlich wurden. Wir mussten im Arduino ein Paar Zeilen Code schreiben, womit die richtigen Daten an das Node-RED übermittelt wurden. Nun konnten wir dort an unsren mqqt in ein sogenanntes Diagramm anschliessen und danach die Daten auf dem Node-RED Dashboard auslesen.

|  |  |
| --- | --- |
| Abbildung 3 Humidity | Abbildung 4 Temperature |
| Abbildung 5 CO2 | Abbildung 6 TVOC |

# Zugriff auf TRADFRI Steckdose

Damit sich z.B. ein Ventilator automatisch einschaltet, wenn es in einem Raum zu heiss ist, muss dieser natürlich mit den anderen Komponenten verbunden sein und auch irgendwie mithilfe von einem Stromsignal eingeschalte werden. Somit konnten wir unser Node-RED mit einem Server verbinden, dieser Server stellte dann eine Verbindung mit dem TRADFRI Gateway her, worüber wir dann unsre Steckdose an der unser Ventilator angeschlossen ist einschalten können.

In der folgenden Abbildung sieht man eine URL, über die wir unsere Steckdosen ansprechen können. Jede Steckdose hat eine andere Nummer (z.B. 192.168.90.73:8080/plugs/65540/enable), wodurch sich die verschiedensten IKEA TRADFRI Steckdosen unterschieden und somit gezielt angesprochen werden können.

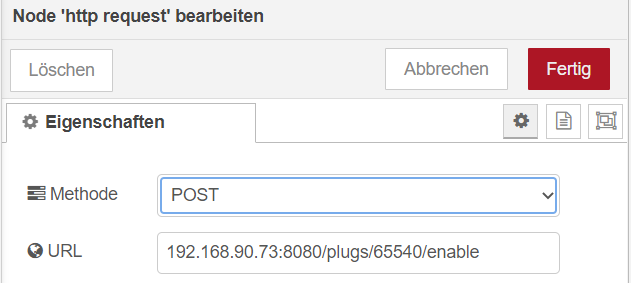


Abbildung 7 http request

# Automatische Anschaltung des Ventilators

Wenn es zu heiss wird, sollte sich unser Ventilator Selbstständige an- sowie ausschalten können. Hierfür konnten wir im Node-RED eine Verknüpfung herstellen (Siehe Abbildung: Node-RED-Verknüpfung) worüber die Temperatur mit der Steckdose verbunden wurde. Darüber hinaus mussten wir eine Grenze für der Temperatur definieren, (Siehe Abbildung: Grenzwerte) damit unser Programm auch weis, wann es was machen muss. Wir haben unseren Grenzwert auf 27 definiert. Nach diesem Vorgang kann die Steckdose automatisch sowie auch manuell betätigt werden.

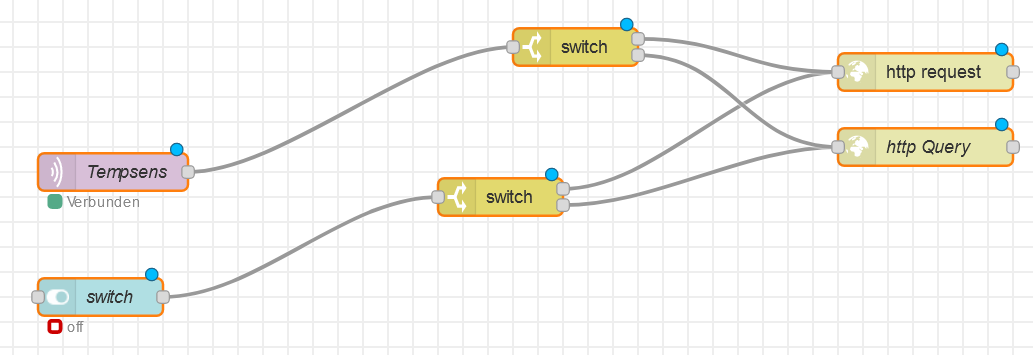


Abbildung 8 Node-RED-Verknüpfung



Abbildung 9 Grenzwert

Datenfluss aller Komponenten:

Auf den unteren Grafiken ist der Datenfluss unserer Komponenten abgebildet.

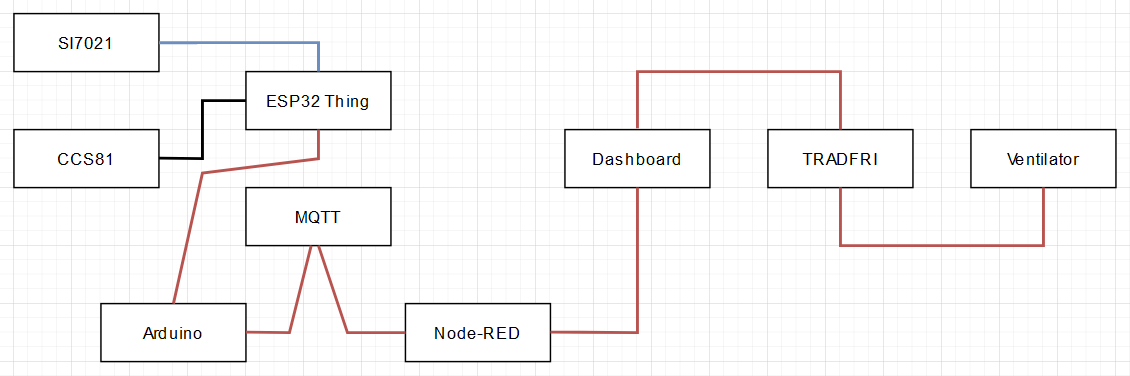
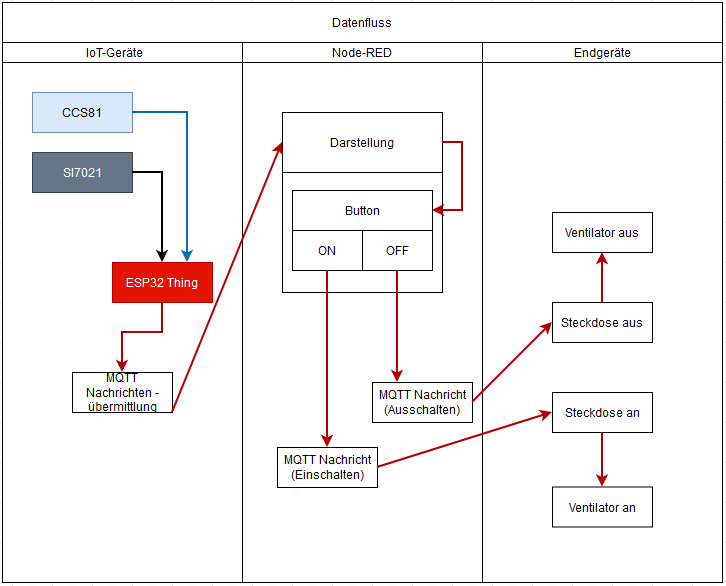


Abbildung 10 Datenfluss 1



1

Abbildung 11 Datenfluss 2

Testen der Verknüpfungen

Auf folgender Grafik sind die Verknüpfungen unserer IoT-Geräte erkennbar.

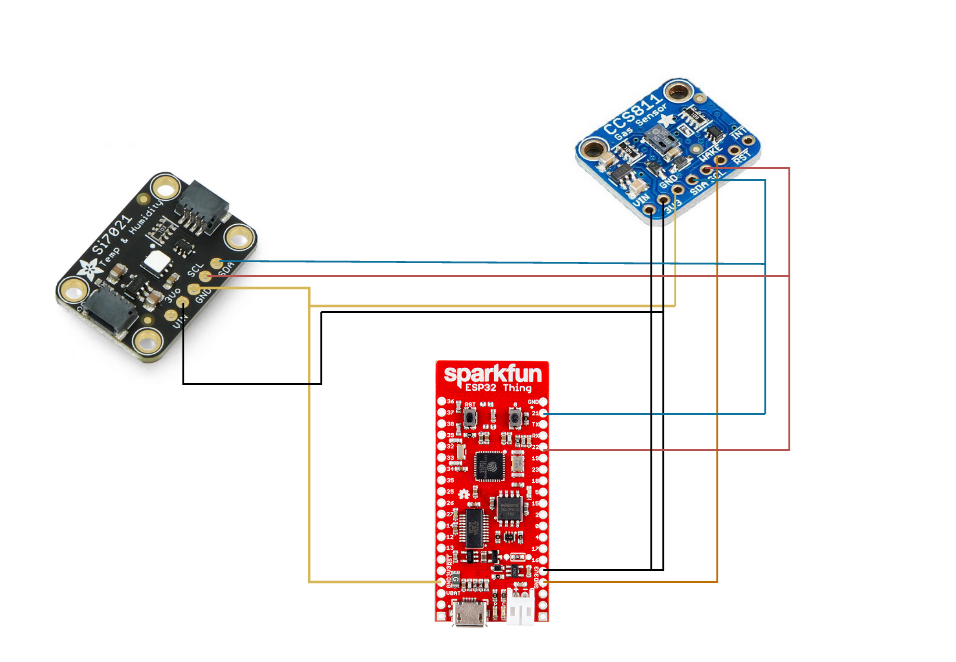


Abbildung 12 Die Verknüpfungen

Testing

Als wir mit unserem Dashboard so gut wie fertig waren, erstellten wir noch einen kleine Black-Box-Test den wir dann von unseren Mitlernenden durchspielen lassen können. In diesem Test geht es darum, dass man keine Ahnung von dem Code hat und nur dies testet, was jeder ohne Informatikerfahrungen auch ausführen könnte. Zuerst erstellten wir eine Tabelle, in welcher die Schritte ganz simpel erklärt wurden. Anschliessend übergaben wir diese Black-Box-Tests an unsere Mitlernenden.

# Was braucht man alles:

Damit man diesen Test durchführen kann, braucht man natürlich das Thing, womit die Daten gemessen werden. Darüber hinaus braucht man einen Laptop, dazu müssen die richtigen Einstellungen auf dem Arduino Programm getroffen werden. Man muss auch über das Node-RED Dashboard verfügen, auf welchem die Daten grafisch dargestellt werden. Am besten macht man diesen Test auf dem Gerät, auf welchem alles erstellt wurde.

# Testziele:

Unser Ziel ist es, das man manuell eine Steckdose ansprechen kann und diese ein- und ausschalten. Dazu sollte ab einer bestimmten Temperatur der Ventilator eingeschalten werden, damit das Zimmer auch wieder abkühlen kann. Natürlich soll der Tester auch einen kleinen Blick auf unser Dashboard werfen können und auch Änderungen vorschlagen.

Testobjekte:

Es wird eigentlich die ganze Funktion unseres IoE-Endgerätes getestet, das heisst am Schluss sollte der Ventilator automatisch eingeschaltet werden.

Testinfrastruktur:

Am wichtigsten sind die Daten der Temperatur, welche wir für die automatische An- Ausschaltung des Ventilators brauchen.

Testorganisation:

Luca ist für die Überprüfung der Durchführungen bei den Black-Box-Tests zuständig.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testfall-NR** | | | **1** | | | |
| **Testfall-Bezeichnung** | | | **Testen der Endgeräte** | | | |
| **Testumgebung** | | | **ACER Arbeitslaptop** | | | |
| **Zu testende Funktionalität** | | | **Testen der IoE-Endgeräte und deren Funktionen** | | | |
| **Datum der Testdurchführung** | | | **26.01.2022** | | | |
| **Tester/in** | | | **Nadja Ehrengruber** | | | |
| **Testschritte:** | | | | | | |
| **Nr.** | **Aktion** | **Erwartetes Ergebnis** | | **Effektives Ergebnis** | **Erfüllt** | **Kommentar** |
| 1 | Öffnen Sie im Browser die Webseite:  localhost:1880/ui | Das Dashboard und die Diagramme werden angezeigt. | | Wie erwartet | **Ja** |  |
| 2 | Versuchen Sie die Steckdose per Switch an und aus zu schalten. | Steckdose wird an und ausgeschaltet. | | Wie erwartet | **Ja** |  |
| 3 | Gehen Sie auf das Burger Menu, danach auf Temperatur. | Nur ein Diagramm wird ausgegeben. | | Wie erwartet | **Ja** |  |
| 4 | Drücken Sie den RST Knopf auf dem ESP32 | Das Diagramm reagiert ein Paar Minuten nicht mehr. | | Wie erwartet | **Ja** |  |
| 5 | Warten Sie bis die Temperatur auf über 27 Grad ansteigt. | Der Ventilator schaltet sich ein. | | Wie erwartet | **Ja** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testfall-NR** | | | **2** | | | |
| **Testfall-Bezeichnung** | | | **Testen der Endgeräte** | | | |
| **Testumgebung** | | | **ACER Arbeitslaptop** | | | |
| **Zu testende Funktionalität** | | | **Testen der IoE-Endgeräte und deren Funktionen** | | | |
| **Datum der Testdurchführung** | | | **26.01.2022** | | | |
| **Tester/in** | | | **Felix Stalder** | | | |
| **Testschritte:** | | | | | | |
| **Nr.** | **Aktion** | **Erwartetes Ergebnis** | | **Effektives Ergebnis** | **Erfüllt** | **Kommentar** |
| 1 | Öffnen Sie im Browser die Webseite:  localhost:1880/ui | Das Dashboard und die Diagramme werden angezeigt. | | Wie erwartet | Ja |  |
| 2 | Versuchen Sie die Steckdose per Switch an und aus zu schalten. | Steckdose wird an und ausgeschaltet. | | Wie erwartet | Ja |  |
| 3 | Gehen Sie auf das Burger Menu, danach auf Temperatur. | Nur ein Diagramm wird ausgegeben. | | Wie erwartet | Ja |  |
| 4 | Drücken Sie den RST Knopf auf dem ESP32 | Das Diagramm reagiert ein Paar Minuten nicht mehr. | | Wie erwartet | Ja |  |
| 5 | Warten Sie bis die Temperatur auf über 27 Grad ansteigt. | Der Ventilator schaltet sich ein. | | Wie erwartet | Ja |  |

Arbeitsaufteilung

|  |  |  |
| --- | --- | --- |
| **Bereiche** | **Silas** | **Luca** |
| Gruppendokumentation | 40% | 60% |
| Arduino | 40% | 60% |
| Verkabelung | 70% | 30% |
| Node-RED | 50% | 50% |
| Node-RED Dashboard | 40% | 60% |

Ausgangslage

Unser Embedded System ist ohne uns bekannte Fehler mit den Sensoren verbunden. Dazu haben wir auf dem Arduino ein einigermassen gutes Skript geschrieben, welches wir auf unser Thing übertragen konnten. Nebenbei können wir ohne Probleme unsere Daten auslesen und in Diagrammen Darstellen lassen. Die Steckdose kann Manuell wie auch automatisch angesprochen werden und so unseren Ventilator einschalten. Darüber hinaus kann man die Daten in unserem Dashboard über das Burger Menu in mehreren Kategorien ausgeben lassen. Auf den folgenden Abbildungen wird unser kleines Projekt grafisch dargestellt und so nochmals eine kleine Übersicht geschaffen.

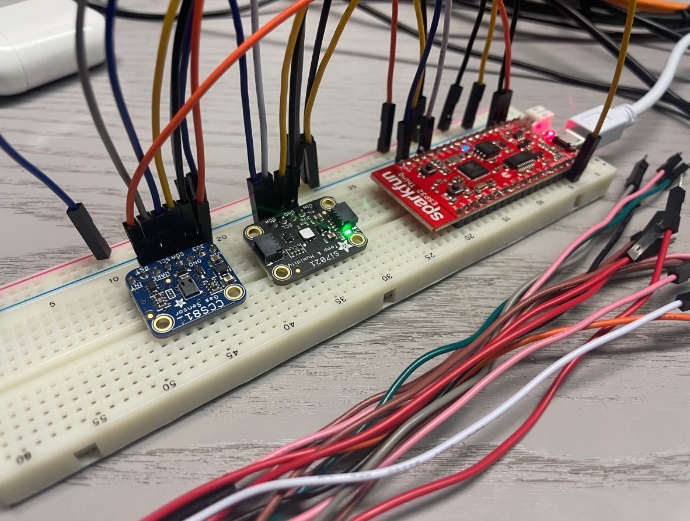
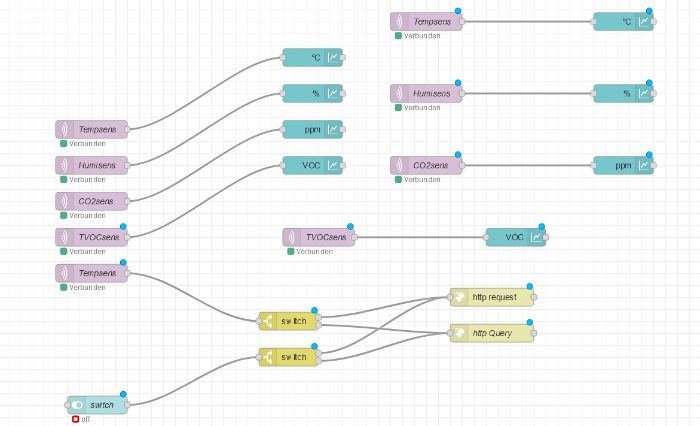


Abbildung 13 Node-RED

Abbildung 14 Komponenten

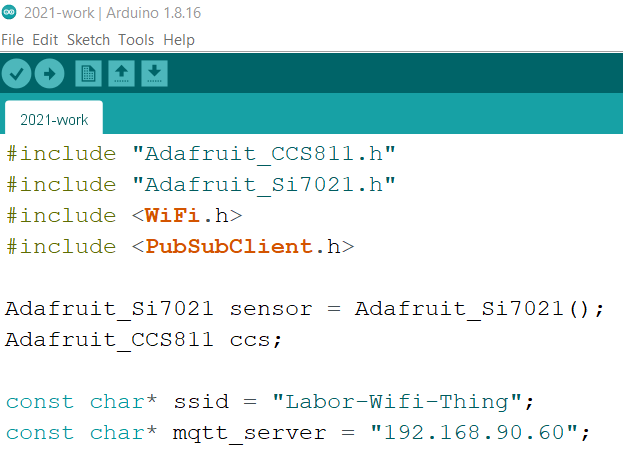
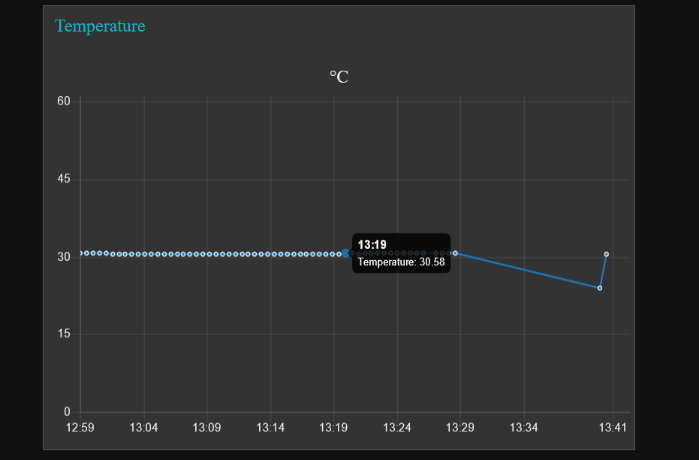


Abbildung 15 Dashboard

Abbildung 16 Arduino

Quellenverzeichnis

Webseiten:

<http://localhost:1880/ui/#!/0?socketid=krRWg1CPZuv0HyDHAAAZ>

<https://cdn.sparkfun.com/r/600-600/assets/learn_tutorials/5/0/7/ESP32ThingV1a.png>

<https://www.electrokit.com/en/product/sparkfun-esp32-thing-2/>

<https://botland.store/multifunctional-sensors/7681-si7021-temperature-humidity-sensor-breakout-5904422310721.html>

<https://www.evelta.com/3566-ccs811-air-quality-sensor-breakout-voc-and-eco2/>

<https://app.diagrams.net/>

Dokumente:

01\_modul-216-ku-grundlagen-ioe-iot\_v1-3.pdf

03\_modul-216\_ku\_einfuehrung-testing\_v1-4.pdf

M216\_Theorie.pdf

node-red.pdf

Abbildungsverzeichnis

[Abbildung 1 ESP32 Ports 10](#_Toc94104713)

[Abbildung 2 Unsere Komponenten 11](#_Toc94104714)

[Abbildung 3 Humidity 13](#_Toc94104715)

[Abbildung 4 Temperature 13](#_Toc94104716)

[Abbildung 5 CO2 13](#_Toc94104717)

[Abbildung 6 TVOC 13](#_Toc94104718)

[Abbildung 7 http request 14](#_Toc94104719)

[Abbildung 8 Node-RED-Verknüpfung 15](#_Toc94104720)

[Abbildung 9 Grenzwert 15](#_Toc94104721)

[Abbildung 10 Datenfluss 1 16](#_Toc94104722)

[Abbildung 11 Datenfluss 2 16](#_Toc94104723)

[Abbildung 12 Die Verknüpfungen 17](#_Toc94104724)

[Abbildung 13 Node-RED 22](file:///E:\00_Dokumente\00_Dokumentationen\00_Word\07_ÜK\UEK_216\UEK_216_Silas_Luca\UEK_216_Projekt_Silas_Luca.docx#_Toc94104725)

[Abbildung 14 Komponenten 22](file:///E:\00_Dokumente\00_Dokumentationen\00_Word\07_ÜK\UEK_216\UEK_216_Silas_Luca\UEK_216_Projekt_Silas_Luca.docx#_Toc94104726)

[Abbildung 15 Dashboard 22](file:///E:\00_Dokumente\00_Dokumentationen\00_Word\07_ÜK\UEK_216\UEK_216_Silas_Luca\UEK_216_Projekt_Silas_Luca.docx#_Toc94104727)

[Abbildung 16 Arduino 22](file:///E:\00_Dokumente\00_Dokumentationen\00_Word\07_ÜK\UEK_216\UEK_216_Silas_Luca\UEK_216_Projekt_Silas_Luca.docx#_Toc94104728)