

Weekly Reports

Luftqualität in Innenräumen - Gruppe 1

4. Dezember 2021

| Name | Matrikel Nr. | Arbeitsaufwand (h) |
|--------------------|--------------|--------------------|
| Friedrich Just | 1326699 | 17,00 |
| Stipe Knez | 1269206 | 17,00 |
| Lucas Merkert | 1326709 | 17,00 |
| Achim Glaesmann | 1309221 | 19,00 |
| Max-Rene Konieczka | 1211092 | 16,00 |
| Can Cihan Nazlier | 1179244 | 21,00 |

Tabelle 1: Arbeitsaufwand dieser Woche

1 Überblick

1.1 Friedrich Just

In dieser Woche habe ich mich mit dem Pflichtenheft befasst, mir angesehen, welche Sensoren wir haben und was diese können. Ich habe mir die passenden Datenblätter [3] [4] [5] zusammengesucht. In diesen sind mir einige Begriffe unklar gewesen, wie VOC bzw. TVOC [12], eCO₂ [11] und relative Luftfeuchtigkeit [9]. Dazu habe ich recherchiert. Danach habe ich diese durchgearbeitet, um die Sensoren und deren Funktionsweise genauer zu verstehen. Außerdem habe ich mich mit den Funkmodulen auseinandergesetzt und wollte den Temperatursensor aus dem ZigBee-Set(LM73) und das ZigBee-Netzwerk miteinander verbinden, so dass die Sensordaten an den Koordinator weitergeleitet werden. Während dieser Zeit habe ich mich erneut mit dem I²C-Bus und der Übertragung von Daten beschäftigt. Abbildung 2 i2c Adresse 0x62 [4]Seite 7

1.2 Stipe Knez

In der ersten Woche lag der Fokus auf zwei Bereichen: Dem Nachbessern des Pflichtenhefts nach einer internen Besprechung und dem Schaffen einer geeigneten Programmierungsumgebung für den weiteren Verlauf des Projektes auf meinem Rechner samt der Einarbeitung in Javascript. Neben dem ausbessern vorhandenem Inhalts haben wir uns Gedanken über unsere Vorgehensweise gemacht

| Write (hexadecimal) | Input parameter: - | | Response parameter: CO ₂ , Temperature, Relative Humidity | | Max. command duration [ms] |
|---|-------------------------------------|-----------------------|--|--|----------------------------|
| | length [bytes] | signal conversion | length [bytes] | signal conversion | |
| 0xec05 | - | - | 3 | CO ₂ [ppm] = word[0] | 1 |
| | | | 3 | T = - 45 + 175 * word[1] / 2 ¹⁶ | |
| | | | 3 | RH = 100 * word[2] / 2 ¹⁶ | |
| Example: read sensor output (500 ppm, 25 °C, 37 % RH) | | | | | |
| Write (hexadecimal) | 0xec05 Command | | | | |
| Wait | 1 ms command execution time | | | | |
| Response (hexadecimal) | 0x01f4 CO ₂ = 500 ppm | 0x7b CRC of 0x01f4 | 0x6667 Temp. = 25 °C | 0xa2 CRC of 0x6667 | 0x5eb9 RH = 37 % |
| | | | | | 0x3c CRC of 0x5eb9 |

Abbildung 1: Auswertung einer Messung [4]

| Domain | Command | Hex. Code | I ² C sequence type (see chapter 3.3) | Execution | |
|--|--|--------------|---|--------------|------------------|
| | | | | time [ms] | During meas.* |
| Basic Commands Chapter 3.5 | start_periodic_measurement | 0x21b1 | send command | - | no |
| | read_measurement | 0xec05 | read | 1 | yes |
| | stop_periodic_measurement | 0x3f86 | send command | 500 | yes |
| On-chip output signal compensation Chapter 3.6 | set_temperature_offset | 0x241d | write | 1 | no |
| | get_temperature_offset | 0x2318 | read | 1 | no |
| | set_sensor_altitude | 0x2427 | write | 1 | no |
| | get_sensor_altitude | 0x2322 | read | 1 | no |
| | set_ambient_pressure | 0xe000 | write | 1 | yes |
| Field calibration Chapter 3.7 | perform_forced_rec calibration | 0x362f | send command and fetch result | 400 | no |
| | set_automatic_self_calibration_enabled | 0x2416 | write | 1 | no |
| | get_automatic_self_calibration_enabled | 0x2313 | read | 1 | no |
| Low power Chapter 3.8 | start_low_power_periodic_measurement | 0x21ac | send command | - | no |
| | get_data_ready_status | 0xe4b8 | read | 1 | yes |
| Advanced features Chapter 3.9 | persist_settings | 0x3615 | send command | 800 | no |
| | get_serial_number | 0x3682 | read | 1 | no |
| | perform_self_test | 0x3639 | read | 10000 | no |
| | perform_factory_reset | 0x3632 | send command | 1200 | no |
| | reinit | 0x3646 | send command | 20 | no |
| Low power single shot (SCD41 only) Chapter 3.10 | measure_single_shot | 0x219d | send command | 5000 | no |
| | measure_single_shot_rht_only | 0x2196 | send command | 50 | no |

Abbildung 2: Befehle zum Ansteuern des SCD41 Sensors [5]

woraufhin ich nach Vorgehensweisen recherchiert habe [**scrum**], von denen wir uns für unser eigenes Projekt inspirieren lassen können. Anschließend habe ich unsere Vorgehensweise nach kurzer Absprache und Beratung mit der Gruppe ausformuliert und für das Pflichtenheft vorbereitet.

Des Weiteren spielte wie zuvor erwähnt auch das Schaffen einer geeigneten Programmierumgebung samt der Einarbeitung in Javascript diese Woche für

mich eine große Rolle. Zuerst habe ich die beiden IDEs Webstorm und IntelliJ IDEA Ultimate eingerichtet. Dabei soll Webstorm der Entwicklung in Javascript und IntelliJ der Entwicklung in Java dienen. Weil ich in Javascript noch nicht allzu viel Programmiererfahrung gesammelt habe, habe ich mich in die Sprache eingearbeitet. Hilfreich waren dabei eine Javascript Dokumentation [10] sowie Lerninhalte in Videoform [8]. Außerdem habe ich noch die GitHub Desktop Anwendung auf meinem Rechner eingerichtet und mich mit LaTeX auseinander-gesetzt und die dazugehörigen Anwendungen installiert.

1.3 Lucas Merkert

Einarbeitung SHT21: Der SHT21 Sensor wird über den I2C-Bus angesprochen Abbildung 3 um die Temperatur und relative Luftfeuchtigkeit zu messen. Der Sensor gibt die Temperatur in einer 14 Bit Auflösung und die Relative Luftfeuchtigkeit in einer 12 Bit Auflösung zurück. Die Werte können dann mit den Formel aus Abbildung 4 und Abbildung 5 berechnet werden. Dabei ist das Problem auf-getreten wie genau man den Sensor über HAL_WriteI2cPacket() anspricht.

| Command | Comment | Code |
|------------------------|----------------|-----------|
| Trigger T measurement | hold master | 1110'0011 |
| Trigger RH measurement | hold master | 1110'0101 |
| Trigger T measurement | no hold master | 1111'0011 |
| Trigger RH measurement | no hold master | 1111'0101 |
| Write user register | | 1110'0110 |
| Read user register | | 1110'0111 |
| Soft reset | | 1111'1110 |

Abbildung 3: Befehle zum Ansteuern des SGT21 Sensors, T für Temperatur, RH für relative Luftfeuchtigkeit [5]

$$T = -46.85 + 175.72 \cdot \frac{S_T}{2^{16}}$$

Abbildung 4: Formel zur Berechnung der Temperatur [5]

$$RH = -6 + 125 \cdot \frac{S_{RH}}{2^{16}}$$

Abbildung 5: Formel zur Berechnung der relativen Luftfeuchtigkeit [5]

1.4 Achim Glaesmann

In der vergangenen Woche wurden von mir folgende Aufgaben bearbeitet. Zunächst wurde das Pflichtenheft ausgebessert, wobei von mir die Projektbeschreibung

angefertigt wurde. Außerdem war es nötig die für die Entwicklung vorraussichtlich benötigte Software zu installieren. Dazu zählte die Installation von Github, so wie eine entsprechende Einarbeitung. Die Installation von IntelliJ IDEA als IDE für die Java Entwicklung sowie eine entsprechende Einarbeitung. Die Installation von WebStorm als IDE für die Javascript Entwicklung so wie eine entsprechende Einarbeitung. Die Installation von MikTex als Compiler für Tex files so wie eine entsprechende Einarbeitung in die Syntax von LaTeX. Weiterhin wurde sich in einer kleinen Gruppe Mittwochs getroffen um die Sensoren in Verbindung mit den Mikrokontrollern zu testen. Bis jetzt war es uns nicht möglich die Daten auszulesen, es ist beabsichtigt das Problem in der kommenden Woche zu lösen. Eine ausgiebige Recherche der Datenblätter sollte hierbei helfen. [5]q Es wurden weiterhin Recherchen betrieben zur Risikoabschätzung der Aerosolbelastung basierend auf dem CO₂ gehalt der Umgebungsluft. Hierbei wurden mehrere Paper gelesen wobei eines bis jetzt die vielversprechensten Informationen lieferte. [6] Die Recherche wird in den kommenden Wochen fortgeführt. Weiterhin wurde die Präsentation zu Analog Digital Wandlern angefertigt. Das Übungsblatt 2 wurde korrigiert. Es wurden insgesamt 4 Meetings mit der Gruppe gehalten.

1.5 Max-Rene Konieczka

Aufbauend zur letzten Woche, hat man sich mit der korrekten Einrichtung des Projektes beschäftigt, welches von Can Cihan Nazlier letzte Woche konfiguriert wurde. Darüber hinaus wurde influxDB installiert, was sich gut dafür eignet Zeitreihen-Daten zu verwalten. Da die Applikation eine Reactive Web App sein wird, wurden Recherchen zum Thema Websockets und SocketIO gemacht. SocketIO ist eine JavaScript-Bibliothek, welche für Echtzeit-Webanwendungen verwendet wird. Diese ermöglicht bidirektionale Echtzeit-Kommunikation zwischen dem Browser und einem Server. Dadurch werden Benutzereingaben schneller behandelt und die App läuft flüssiger. Websocket wiederum ist ein Netzwerkprotokoll, was auf TCP basiert. SocketIO macht sich das Websocketprotokoll zunutze.

1.6 Can Cihan Nazlier

Im Verlauf der Woche habe ich unserer Projekt konfiguriert und auf Github gepusht. Er besteht aus einem backend, frontend, models und einem electron Teil. Ich habe das frontend mit node so konfiguriert, dass es sich in den electron Ordner buildet und wir eine desktop application daraus erstellen können. Das backend habe ich mit express aufgesetzt und die von Herrn Merkl empfohlene Datenbank influxDB integriert. Zudem habe ich angefangen das Zeichentool zu programmieren und einen ersten Prototypen zu entwickeln. Des weiteren habe ich mich mit den Schnittstellen befasst und dem Datenaustausch zwischen den modulen und den Mockup. Ich habe erste Datentransfermodelle entwickelt und im Laufe der Woche werde ich Mockup Daten erstellen und mit diesen erste use-cases nachstellen. Ich habe mich auch noch mit Socket IO auseinandergesetzt,

weil wir mit Websockets arbeiten werden, um eine reactive app zu gestalten.

Literatur

- [1] DeKay Arts. *Timeseries Database: A gentle introduction to Influxdb in Nodejs*. URL: <https://www.youtube.com/watch?v=wwAsjs2XcBU&t=3s>.
- [2] *Compare InfluxDB to SQL databases*. URL: <https://docs.influxdata.com/influxdb/v1.8/concepts/crosswalk/>.
- [3] *Datasheet CCS811*. URL: <https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor?view=all#documents>.
- [4] *Datasheet SCD41*. URL: <https://www.sensirion.com/en/environmental-sensors/evaluation-kit-sek-environmental-sensing/evaluation-kit-sek-scd41/#c50745>.
- [5] *Datasheet SHT21*. URL: <https://www.sensirion.com/de/umweltsensoren/feuchtesensoren/feuchte-temperatursensor-sht2x-digital-i2c-genauigkeit/>.
- [6] *Exhaled CO₂ as a COVID-19 Infection Risk Proxy for Different Indoor Environments and Activities*. URL: <https://pubs.acs.org/doi/10.1021/acs.estlett.1c00183>.
- [7] *JavaScript Tutorials*. URL: <https://developer.mozilla.org/en-US/docs/Web/Tutorials>.
- [8] *Learn JavaScript - Full Course for Beginners*. URL: <https://www.youtube.com/watch?v=PkZNo7MFNFg>.
- [9] *Relative Luftfeuchtigkeit*. URL: <https://www.ratgeber-luftfeuchtigkeit.de/relative-luftfeuchtigkeit/>.
- [10] *The Modern Javascript Tutorial*. URL: <https://javascript.info>.
- [11] *TVOC and ECO₂ Sensors*. URL: <https://wolles-elektronikkiste.de/en/tvoc-and-eco2-sensors>.
- [12] *Understanding TVOC: What You Need To Know About Volatile Organic Compounds*. URL: <https://learn.kaiterra.com/en/air-academy/understanding-tvoc-volatile-organic-compounds>.