Die hard- und softwaretechnische Implementierung eines CO_2-Sensors zur Messung der Raumluftqualität

Julius Caesar, Péter Egermann, Paul Görtler, Johannes Leyrer

4. Mai 2022

Inhaltsverzeichnis

Abkürzungsverzeichnis

1 Motivation

Im Laufe der Corona-Pandemie und der damit verbundenen Ausgangsbeschränkungen musste man sich zunehmen in Innenräumen aufhalten, um der Verbreitung des Virus entgegenzuwirken. Aus diesem Grund musste man sich zwangsweise mit der Luftqualität im Homeoffice und besonders in Büroräumen auseinandersetzen, um das Gesundheitsrisiko zu minimieren und konzentriert arbeiten zu können. Da die Luftqualität durch Menschen meist nur subjektiv wahrgenommen werden kann, benötigt man dafür eine Messstelle. Da die gesundheitlichen Auswirkungen von schlechter Raumluftqualität ausreichend erforscht sind, existieren genügend konkrete Werte zur Orientierung.
Diese Arbeit soll einen Überblick darüber geben, welche medizinischen Risiken durch eine unzureichende Luftqualität entstehen können und welche positiven Auswirkungen eine ausreichende Versorgung mit Frischluft haben kann.
Danach soll ein grober Leitfaden zur Einrichtung eines CO_2-Sensors zur Bestimmung der Luftqualität in Arbeitsräumen erstellt werden. Dabei soll über die Anforderungen an die Hardware, die benötigten Komponenten, die Einbindung in das Netzwerk und die Umsetzung der dazugehörigen Software eingegangen werden. Die Arbeit richtet sich dabei hauptsächlich an technisch versierte Leser, welche bereits grundlegende Kenntnisse in den Bereichen Hard- und Software besitzen.

2 CO_2-Grenzwerte und deren Auswirkungen auf den Menschen

2.1 CO_2-Grenzwerte für eine unbedenkliche Atemluft

2.2 Physiologische Auswirkungen eines zu hohen CO_2-Gehaltes in der Atemluft

3 Hardwaretechnische Umsetzung

3.1 Technische Anforderungen an die benötigte Hardware

3.2 Überblick über die verwendete Hardware

4 Softwaretechnische Umsetzung

4.1 Benötigte Software

Um die Programme rund um den CO_2-Monitor in Betrieb nehmen zu können, wird folgende Software benötigt:

- PiOS mit mitgelieferter Standardsoftware
- Docker
- docker-compose

Ist die benötigte Software installiert und eingerichtet, kann mit der Implementierung der Auslese- und Verarbeitungssoftware begonnen werden. Alle in den folgenden Kapiteln genannten Softwarekomponenten sind auf GitHub zu finden.

4.2 Zusammenspiel der Softwarekomponenten

Die Daten des CO_2-Sensors werden persistent gespeichert, damit die Werte abrufbar sind und auch über längere Zeiträume ausgewertet werden können. Dafür werden die CO_2- und Temperaturwerte mittels einer Software-Komponente zum Auslesen der Daten an ein backend gesendet und gespeichert. Diese Daten können dann mittels eines frontends angezeigt werden. Diese Verknüpfung der Software-Komponenten ist in Abb. 4.1 auf der nächsten Seite zu sehen.

4.3 Aufbau und Einrichten der Software

4.3.1 Aufbau und Einrichten des backends

Die zentrale Stelle, an der Daten eingehen, gespeichert und abgerufen werden können, wird mittels der *CO2MonitorAPI* realisiert. Diese ist in Python geschrieben und verwendet FastAPI als Grundlage für das Bereitstellen einer API. Um das Bereitstellen der Anwendung und deren Isolierung vom Betriebssystem zu erleichtern, wird Docker als Containervirtualisierungssoftware eingesetzt.
Nachdem die Anwendung von GitHub bezogen wurde, müssen noch Konfigurationswerte angepasst werden. Beispielsweise muss der Port festgelegt werden, auf dem die Application Programming Interface (API) erreichbar sein soll. Hierfür kann in der `docker-compose.yml`-Datei besagter Port angegeben werden, welcher dem Frontend und dem CO2Reader Zugang zur Backend-Logik erlaubt, um Daten abzuspeichern und abzurufen.
Durch das Öffnen eines Terminal-Fensters im Ordner der API und der Eingabe des Befehls

```
docker-compose -f docker-compose.yml up -d
```

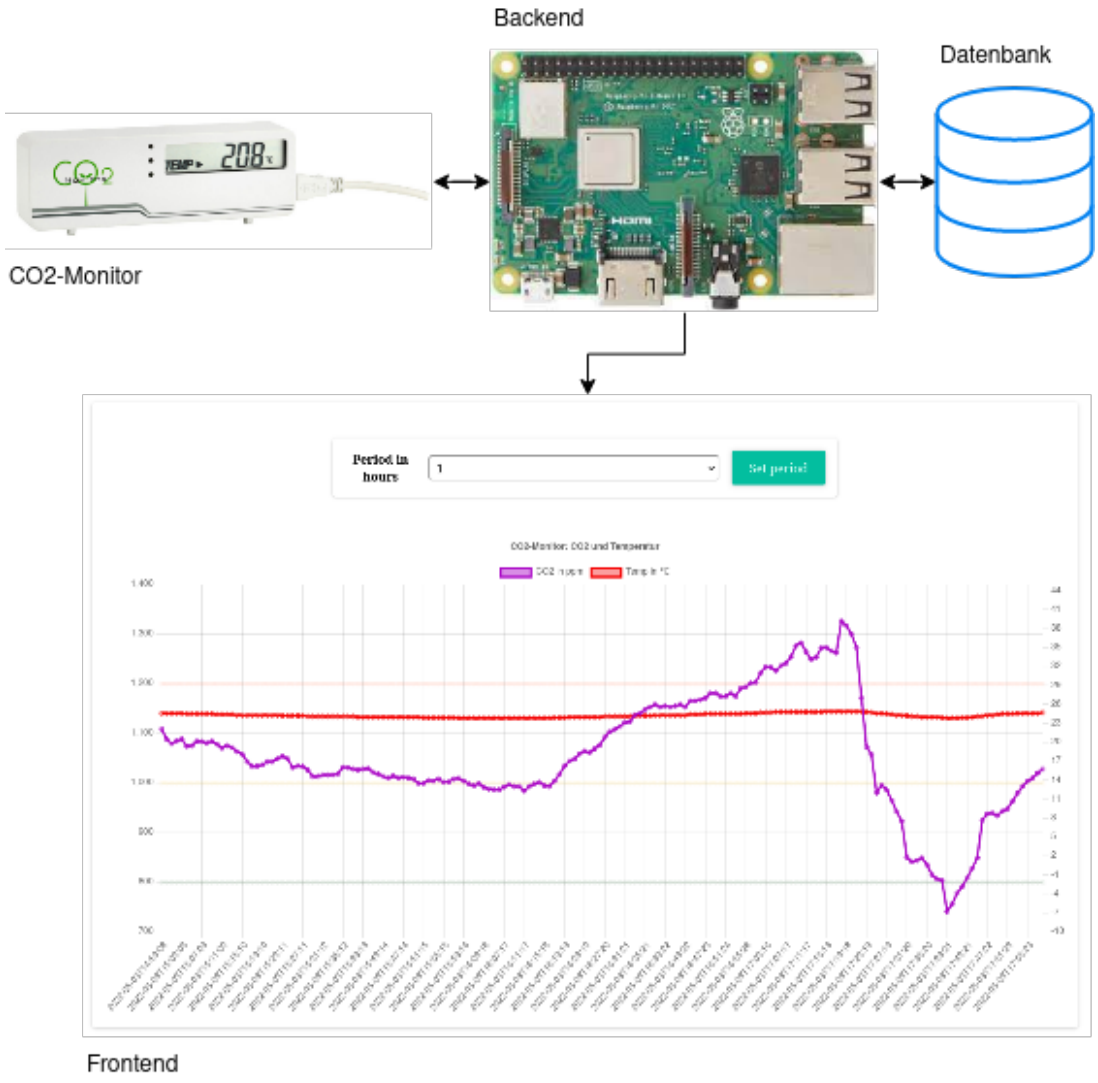

Abbildung 4.1: Verbundplan der Komponenten

wird die Anwendung gestartet. Der Docker-Container läuft ab jetzt im Hintergrund und wartet auf Speicher- oder Abrufbefehle. Ob die Applikation richtig funktioniert kann mittels `IpAdresseDesPis:angege` getestet werden.

4.3.2 Aufbau und Einrichten der Lese-Software

Die Daten des CO_2-Sensors werden mittels der USB-Schnittstelle ausgelesen. Diese Lesesoftware ist in Python geschrieben und nutzt die CO2Meter-Bibliothek von Vladimir Filimonov. [1]
Dazu muss die `docker-compose.yml`-Datei des *Readers* angepasst werden. Um den richtigen USB-Port in die Datei schreiben zu können, können alle angeschlossenen USB-Geräte mit dem in Listing 4.1 zu sehenden Bash-Script angezeigt und die hidraw-Id des Geräts *Holtek Semiconductor, Inc. USB-zyTemp* ausgelesen werden. [2]

Listing 4.1: Bash-Script zum Erkennen der hidraw-Id

```
1 #!/bin/bash
2
3 FILES=/dev/hidraw*
4 for f in $FILES
5 do
6   FILE=${f##*/}
7   DEVICE="$(cat /sys/class/hidraw/${FILE}/device/uevent | grep HID_NAME
      | cut -d '=' -f2)"
8   printf "%s \t %s\n" $FILE "$DEVICE"
9 done
```

Ist der USB-Port bestimmt, kann der Teil vor dem Doppelpunkt der `devices`, in diesem Fall „`/dev/hidraw0`“, mit dem ausgelesenen Port ersetzt werden, zu sehen in Listing 4.2.

Listing 4.2: Anpassen des USB-Ports in der docker-compose.yml

```
1 ...
2   devices:
3     - /dev/hidraw0:/dev/hidraw16
4 ...
```

Soll der *Reader* auf einem anderen Gerät als die API ausgeführt werden, muss die `co2Reader.ini`-Datei angepasst werden. Diese ist in `app/co2Reader.ini` zu finden. Hier muss die IP-Adresse der API anstelle der bestehenden IP-Adresse angegeben werden. Auch kann hier der Ort, an dem sich der Sensor befindet, eingetragen werden.
Nach dem Abspeichern der Datei kann ein Terminal-Fenster im Ordner des *Readers* geöffnet werden und mittels

```
docker-compose -f docker-compose.yml up -d
```

die Anwendung gestartet werden. Der Docker-Container läuft ab jetzt im Hintergrund, liest die Daten des Sensors aus und schickt diese an die angegebene IP-Adresse der API.

4.3.3 Aufbau und Einrichten des Frontends

Um die Daten ansehnlich darstellen zu können, kann ein Frontend eingebunden werden. Das hier verwendete Frontend ist mit React [6] und ChartsJs [3] umgesetzt worden. Ein Beispiel der Datenvisualisierung ist in Abb. 4.2 zu sehen.

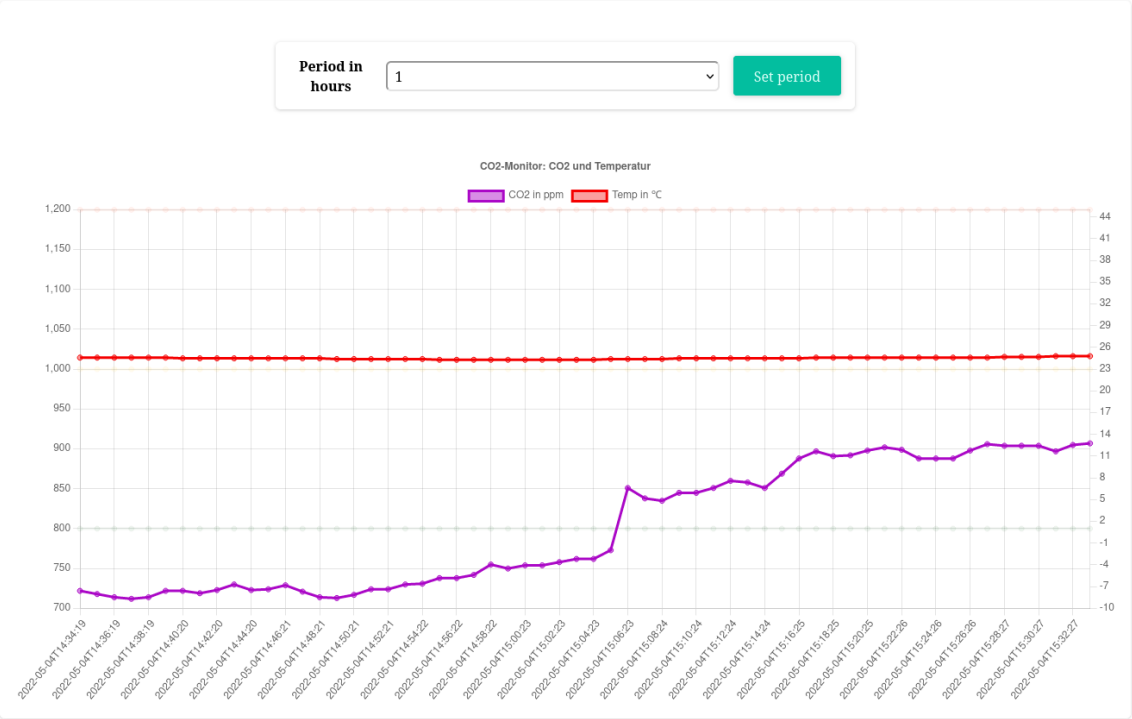

Abbildung 4.2: Datenvisualisierung mittels Frontend

Nachdem das Frontend bezogen wurde muss die `docker-compose.yml`-Datei angepasst werden. In dieser Datei muss die IP-Adresse der `REACT_APP_API_URL` mit der IP des Raspberry Pis, auf dem die API läuft ausgetauscht werden, zu sehen in Listing 4.3.
Der ausgehende Port, von dem das Frontend am Ende erreichbar ist, kann ebenfalls angepasst werden. Hierzu muss lediglich der ports-Abschnitt verändert werden, ebenfalls zu sehen in Listing 4.3. Hier muss der Port vor dem Doppelpunkt auf den gewünschten Port gesetzt werden.

Listing 4.3: Anpassen der API-IP und des Ports in der docker-compose.yml

```
1  ...
2  environment:
3    REACT_APP_API_URL: http://192.168.178.33:8008/api/
4  ports:
5    - 3000:3000
6  ...
```

Mit dem Öffnen eines Terminal-Fensters im Ordner des Frontends und mittels

```
docker-compose -f docker-compose.yml up -d
```

wird Anwendung gestartet. Der Docker-Container läuft ab jetzt im Hintergrund und kann mittels der IP-Adresse des ausführenden Gerätes sowie dem in der docker-compose angegebenen Port aufgerufen werden.

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Glossar

backend Als Backend wird der Teil eines IT-Systems bezeichnet, der sich mit der Datenverarbeitung im Hintergrund beschäftigt – der Data Layer. Der Begriff dient der Unterteilung bei komplexeren Softwarestrukturen. Die Schreibweise wird vom Duden nicht genau vorgegeben. [4]. 2, 7

frontend ADer Begriff Frontend dient bei komplexeren Softwarestrukturen der Unterteilung. Bei einem IT-System bezeichnet das Frontend die Presentation Layer, also den Teil eines IT-Systems, der näher am Anwender ist. [5]. 7

Literatur

[1] Vladimir Filimonov. *CO2meter*. Verfügbar unter: `https://github.com/vfilimonov/co2meter`.

[2] *Matching /dev/hidraw* devices with physical devices*. en. 10/2019. Verfügbar unter: `https://arvchristos.github.io/post/matching-dev-hidraw-devices-with-physical-devices/` [04. 05. 2022].

[3] N. N. *Chart.js | Open source HTML5 Charts for your website*. Verfügbar unter: `https://www.chartjs.org/`. abgerufen am 27.04.2022.

[4] N. N. *Definition Backend Erklärung Backend*. Verfügbar unter: `http://www.softselect.de/business-software-glossar/backend`.

[5] N. N. *Definition Frontend Erklärung Frontend*. Verfügbar unter: `http://www.softselect.de/business-software-glossar/frontend`.

[6] N. N. *Getting Started – React*. en. Facebook Inc. Verfügbar unter: `https://reactjs.org/docs/getting-started.html`. abgerufen am 27.04.2022.

Anhang

A this, that, etc.

B Something something