

Die hard- und softwaretechnische Implementierung eines CO₂-Sensors zur Messung der Raumluftqualität

Julius Caesar, Péter Egermann, Paul Görtler, Johannes Leyrer

3. Mai 2022

Inhaltsverzeichnis

1	Motivation	4
2	CO₂-Grenzwerte und deren Auswirkungen auf den Menschen	5
2.1	CO ₂ -Grenzwerte für eine unbedenkliche Atemluft	5
2.2	Physiologische Auswirkungen eines zu hohen CO ₂ -Gehaltes in der Atemluft . . .	5
3	Hardwaretechnische Umsetzung	6
3.1	Technische Anforderungen an die benötigte Hardware	6
3.2	Überblick über die verwendete Hardware	6
4	Softwaretechnische Umsetzung	7
4.1	Benötigte Software	7
4.2	Zusammenspiel der Softwarekomponenten	7
4.3	Einrichten der Software	7
4.3.1	Einrichten des Backends	7
4.3.2	Einrichten der Lese-Software	8
4.3.3	Einrichten des Frontends	8
	Abbildungsverzeichnis	9
	Tabellenverzeichnis	10
	Listings	11
	Anhang	14

Abkürzungsverzeichnis

API Application Programming Interface	7
--	---

1 Motivation

Im Laufe der Corona-Pandemie und der damit verbundenen Ausgangsbeschränkungen musste man sich zunehmend 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 CO₂-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 CO₂-Grenzwerte und deren Auswirkungen auf den Menschen

2.1 CO₂-Grenzwerte für eine unbedenkliche Atemluft

2.2 Physiologische Auswirkungen eines zu hohen CO₂-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 CO₂-Monitor in Betrieb nehmen zu können, wird folgende Software benötigt:

- PiOS mit mitgelieferter Standardsoftware
- Docker [2, 4, 5]
- Python [1, 3]
- docker-compose

Ist die Software installiert und eingerichtet, kann mit der Implementierung der Auslese- und Verarbeitungssoftware begonnen werden.

4.2 Zusammenspiel der Softwarekomponenten

Um die Daten des CO₂-Sensors persistent zu gestalten, die Werte stets im Blick zu behalten und auch längere Zeiträume auswerten zu können, werden die Werte des Sensors mittels einer Auslesesoftwarekomponente an ein Backend gesendet und gespeichert. Diese Daten können dann mittels eines Frontends angezeigt werden. Diese Verknüpfung von Software-Komponenten ist in Abbildung xyz zu sehen.

4.3 Einrichten der Software

4.3.1 Einrichten des Backends

Die zentrale Stelle, an der Daten eingehen und gespeichert sowie abgerufen werden können, wird mittels der CO2MonitorAPI realisiert. Hier kann in der docker-compose-Datei der Port bestimmt werden, auf dem die Application Programming Interface (API) erreichbar ist.

Hinweis: Wird dieser geändert, müssen im *Reader* und im Frontend der Port der API ebenfalls angepasst werden.

Mit dem Öffnen eines Terminal-Fensters im Ordner der API und mittels `docker-compose up` 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:8008/api/test`
getestet werden.

4.3.2 Einrichten der Lese-Software

Die Daten des CO₂-Sensors werden mittels der USB-Schnittstelle ausgelesen. Dazu muss die docker-compose-Datei des *Readers* angepasst werden. Um den richtigen USB-Port in die Datei schreiben zu können, muss dieser vorher bestimmt werden, was mit folgendem Befehl in einem Terminal-Fenster funktioniert:

Ist der USB-Port bestimmt, kann der vordere Teil der devices, also `"/dev/hidraw0"` mit dem ausgelesenen Port ersetzt werden.

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 Ort, in 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 up` 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 Einrichten des Frontends

Um die Daten ansehnlich darstellen zu können, kann ein Frontend eingebunden werden. Dazu muss lediglich das Frontend bezogen werden und die docker-compose-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.

Der ausgehende Port, von dem das Frontend am Ende erreichbar ist, kann ebenfalls angepasst werden. Hierzu muss lediglich der ports-Abschnitt angepasst werden.

Mit dem Öffnen eines Terminal-Fensters im Ordner des Frontends und mittels `docker-compose up` 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

Docker ist eine freie Softwareplattform der Docker, Inc. Das Ziel von Docker ist es, das Erstellen, Testen und Bereitstellen von Software im wesentlichen zu beschleunigen. Mit Docker können Softwarepakete, sogenannte Container, erstellt werden. Diese gewährleisten eine Trennung und Verwaltung der lokalen Computerressourcen. Die am 13.03.2013 erstmals veröffentlichte Software ist unter anderem für die Betriebssysteme Linux, Microsoft Windows und macOS verfügbar. 7

Python ist eine plattformunabhängige, multiparadigmatische, stark und dynamisch typisierte Programmiersprache. Die Programmiersprache zeichnet sich besonders durch ihre Trivialität und vielseitige Einsetzbarkeit aus. Python ist eine der meist verbreiteten Programmiersprachen weltweit. Die Weiterentwicklung der Programmiersprache wird durch die Python Software Foundation und den ursprünglichen Entwickler Guido van Rossum durchgeführt. 7

Literatur

- [1] N. N. *3.10.4 Documentation*. Verfügbar unter: <https://docs.python.org/3/>. abgerufen am 27.04.2022.
- [2] N. N. *Docker overview*. en. 04/2022. Verfügbar unter: <https://docs.docker.com/get-started/overview/>. abgerufen am 03.05.2022.
- [3] N. N. *History and License — Python 3.10.4 documentation*. Verfügbar unter: <https://docs.python.org/3/license.html>. abgerufen am 03.05.2022.
- [4] N. N. *Install Docker Engine*. en. 04/2022. Verfügbar unter: <https://docs.docker.com/engine/install/>. abgerufen am 27.04.2022.
- [5] N. N. *Orientation and setup*. en. 04/2022. Verfügbar unter: <https://docs.docker.com/get-started/>. abgerufen am 27.04.2022.

Anhang

A this, that, etc.

B Something something