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

Julius Caesar, Péter Egermann, Paul Görtler, Johannes Leyrer

14.01.2022

1 Motivation

1.1 Warum interessiert uns die Fragestellung?

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.

1.2 Welche Ziele wollen wir erreichen?

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 Softwaretechnische Umsetzung

2.1 Benötigte Software

Um die benötigte Software in Betrieb nehmen zu können, wird folgende Software benötigt:

- PiOS mit mitgelieferter Standardsoftware
- Docker
- docker-compose

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

2.2 Einrichten der Software

2.2.1 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 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.

2.2.2 Einrichten der Lese-Software

Die Daten des CO_2-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.

2.2.3 Einrichten des Frontends

Um die Daten ansehnlich darstellen zu koennen, 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.