

# **Die hard- und softwaretechnische Implementierung eines CO<sub>2</sub>-Sensors zur Messung der Raumluftqualität**

Julius Caesar, Péter Egermann, Paul Görtler, Johannes Leyrer

4. Mai 2022

# Inhaltsverzeichnis

<b>1</b>	<b>Motivation</b>	<b>4</b>
<b>2</b>	<b>CO<sub>2</sub>-Grenzwerte und deren Auswirkungen auf den Menschen</b>	<b>5</b>
2.1	CO <sub>2</sub> -Grenzwerte für eine unbedenkliche Atemluft . . . . .	5
2.2	Physiologische Auswirkungen eines zu hohen CO <sub>2</sub> -Gehaltes in der Atemluft . . .	5
<b>3</b>	<b>Hardwaretechnische Umsetzung</b>	<b>6</b>
3.1	Technische Anforderungen an die benötigte Hardware . . . . .	6
3.2	Überblick über die verwendete Hardware . . . . .	6
<b>4</b>	<b>Softwaretechnische Umsetzung</b>	<b>7</b>
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
	<b>Abbildungsverzeichnis</b>	<b>9</b>
	<b>Tabellenverzeichnis</b>	<b>10</b>
	<b>Listings</b>	<b>11</b>
	<b>Anhang</b>	<b>14</b>

## Abkürzungsverzeichnis

<b>API</b> 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 CO<sub>2</sub>-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<sub>2</sub>-Grenzwerte und deren Auswirkungen auf den Menschen**

### **2.1 CO<sub>2</sub>-Grenzwerte für eine unbedenkliche Atemluft**

### **2.2 Physiologische Auswirkungen eines zu hohen CO<sub>2</sub>-Gehaltes in der Atemluft**

## 3 Hardwaretechnische Umsetzung

### 3.1 Technische Anforderungen an die benötigte Hardware

Um einen CO<sub>2</sub>-Monitor betreiben zu können werden ein Computer und ein CO<sub>2</sub>-Sensor benötigt. Die Mindestanforderungen an den Computer sind 512MB Arbeitsspeicher, einen ARM Cortex-A53 Quad-Core 1GHz 64-bit Prozessor und mindestens einem USB Type A Anschluss. Diese Werte sind die Mindestanforderungen von einem minimalen Linux Betriebssystem, welches benötigt wird um die Softwareanforderungen zu erfüllen.

Der CO<sub>2</sub>-Sensor sollte ein Messbereich von 0ppm bis einschließlich 2000ppm vorweisen. Mit dieser Spannweite können die in Kapitel 2 auf der vorherigen Seite vorgestellten Werte für die Aussage über Luftqualität gemessen und aufgezeichnet werden.

### 3.2 Überblick über die verwendete Hardware

Für die Verwendung der Programme rund um den CO<sub>2</sub>-Monitor wird folgende Hardware benötigt:

- Raspberry Pi 3 Model B+
- TFA-Dostmann CO<sub>2</sub>-Monitor AIRCO2NTROL MINI

Der Raspberry Pi ist ein Minicomputer auf einer nur Scheckkarten-großen Platine, der sich vielfältig erweitern lässt. Er wurde ursprünglich zu Lern- und Demonstrationszwecken entwickelt, eignet sich aber auch für die Realisierung verschiedener privater oder professioneller Anwendungen.

Mit dem CO<sub>2</sub>-Monitor AIRCO2NTROL MINI kann die Luft in Räumen, in denen sich Personen aufhalten, z.B. Schulen, Büros, öffentliche Einrichtungen, kontrolliert werden. Der Monitor zeigt auf einem Display den CO<sub>2</sub>-Gehalt der Raumluft im Wechsel mit der Temperatur an. Eine farbige Ampelanzeige neben dem Display wechselt je nach Luftqualität von Grün über Gelb zu Rot.

## 4 Softwaretechnische Umsetzung

### 4.1 Benötigte Software

computer [1]

Um die Programme rund um den CO<sub>2</sub>-Monitor 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.

### 4.2 Zusammenspiel der Softwarekomponenten

Um die Daten des CO<sub>2</sub>-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 CO<sub>2</sub>-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

**computer** is a programmable machine that receives input, stores and manipulates data, and provides output in a useful format. 7

## Literatur

- [1] N. N. *3.10.4 Documentation*. Verfügbar unter: <https://docs.python.org/3/>. abgerufen am 27.04.2022.

## **Anhang**

**A this, that, etc.**

**B Something something**