Indoor air measuring device

Tatiana Boe (70474)

Supervisor: Sune TB Nielsen

Bachelor Project

Roskilde University Center

Spring 2023

Characters with spaces: 45012

Abstract

Indoor air quality significantly affects human health and comfort as individuals spend a majority of their lives in enclosed spaces, making it important to prioritize the quality of the air we breathe indoors. This research study specifically focuses and provides information on two common indoor pollutants, namely volatile organic compounds (VOCs) and carbon dioxide (CO_2). The growing awareness surrounding air pollution has resulted in increase popularity of indoor air monitoring devices and air purifiers. In line with this trend, this project aims to develop and implement a functional minimum viable indoor air measuring device using an ESP32 microcontroller unit, CO_2 sensor and total VOC (tVOC) sensor. The testing phase is conducted in city studio apartment and examines how various daily household activities influence CO_2 and VOC levels. The findings of this study underscore existing knowledge on indoor air quality's impact on human health and the importance of monitoring CO_2 and VOC levels in indoor spaces. It also emphasizes the role of ventilation in reducing these indoor pollutants, as observed during routine household activities. This study's finding can be used as a base to establish household recommendations that would reduce indoor pollutants and the measuring device can be further expanded into a personal household monitoring product.

Extended Abstract

How can different household activities influence indoor air quality?

Author: Tatiana Boe

Supervisor: Sune TB Nielsen

Figure 1: Household air pollution

Indoor air quality (IAQ) can be defined as the air within enclosed spaces including homes, workplaces, schools, and other buildings. It takes into account factors such as the existence of pollutants, gases, particles, humidity levels, temperature, and ventilation conditions. Because humans spend a large portion of their lives indoors, the IAQ significantly influences human health and comfort[1]. According to the World Health Organization's (WHO) data from 2022 on diseases caused by household air pollution, more than 3 million people die prematurely each year due to air quality related illnesses. Recent evidence suggests that household air pollution may contribute to additional health problems, including low birth weight, stillbirth, asthma, ear infections, upper respiratory infections, tuberculosis, nasopharyngeal and laryngeal cancers and cervical cancer[2].

The most common indoor air pollutants are carbon monoxide (CO), carbon dioxide (CO_2), volatile organic compounds (VOCs), particulate matter (PM), and ozone (O3) [3]. This project is closely examining indoor VOCs and CO_2 levels. High indoor emissions of CO_2 have a negative impact on human health and it is among the most ordinary pollutant found indoors[4]. The safe indoor levels are below 1000 ppm (particles per million). Once the levels are higher, people might experience dizziness, headache, fatigue or loss of concentration[5].

The source of indoor VOC pollutants is most commonly from building materials, cleaning products and furnishing[6]. Other sources can be from cooking or tobacco products. The health consequences of VOCs can be even more major than from high CO_2 levels. VOCs can primarily cause respiratory irritations, migraines, throat discomfort or fatigue but a prolonged exposure can result in long term respiratory illnesses or various types of cancer[7][8]. A safe indoor level for VOCs is 200 µg/m3 for total VOCs[9].

Other common pollutants are particulate matter (PM) pollutants. They can also be harmful when inhaled into the lungs. Activities resulting in higher PM levels are mostly cooking, cleaning, smoking and burning candles. They often enter indoor air from outdoors from for example vehicle exhaust, industrial processes, wildfires or even pollen from plants[10]. Additionally, outdoor air pollution can influence negatively an indoor air quality. Cities with industrial zones and high traffic have higher air pollution[11]. High outdoors air pollution impact indoor air pollution. Furthermore, inadequate ventilation, elevated indoor temperatures and high humidity can lead to a higher concentrations of pollutants in indoor air[6].

In order to reduce indoor air pollutants it is recommended to enhance ventilation of the indoor space. Ventilation can be manual of mechanical. Manual ventilation is simple as opening windows or doors to dilute indoor air with outdoor air and mechanical ventilation consists of electronic ventilators and air purifying devices which are nowadays becoming more and more popular[12][13][14]. Monitoring devices, as well as air purifiers, are currently gaining increasing popularity due to an increase interest in air pollution as well as recent experience of COVID-19 pandemic[15].

This project examines how daily household activities can influence levels of two common indoor pollutants, VOC and CO_2. In order to measure these levels a simple measuring device is engineered and implemented. The device is build using Microcontroller unit ESP32, CO2 sensor, tVOC sensor, OLED display on a Printed circuit board. It is assembled with the help of secondary tools such as female/male pin header connectors, electrical wires, solder, soldering iron and wire stripper. The assembled device is powered through an USB cable into a computer. The code of this device is implemented using integrated development environment PlatformIO inside of a source-code editor called Visual Studio Code. The programming language used is C/C++. Final product can be seen in figure 6a below.

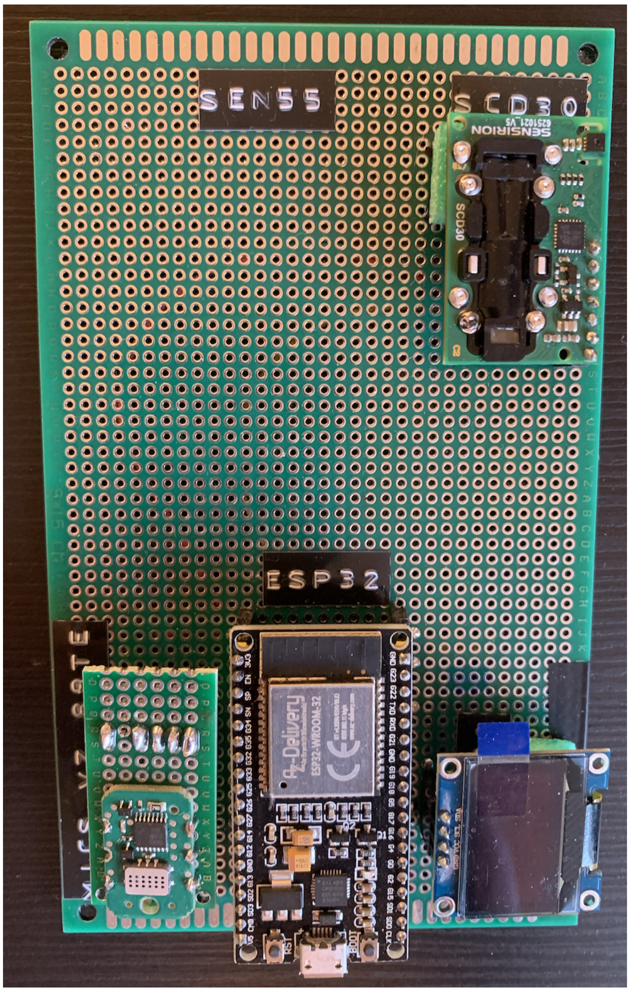


Figure 2: Final board

Firstly, the CO_2 sensor was implemented and tested and later the VOC sensor. To test the CO_2 sensor a simple breathing test was conducted. The test included two parts. First part was with a person standing in a larger distance to the sensor performing deep inhaling and exhaling and second part with a very close distance. This test was selected because exhaling emits a high values of CO_2. The results confirmed the sensor's functionality and displayed live readings on LED display. After implementing tVOC sensor a larger 31 hour experiment was conducted inside of a city studio apartment while performing basic household activities to determine how different activities and application of ventilation can influence CO_2 and VOC levels. The reason for such a larger time frame was to investigate the levels both during the day and during the night. The outcome of this experiment brought a lot of insight into how much VOC and CO_2 different activities generate and how easily a harmful levels can be reached, as well as, how much a manual ventilation can lower these values.

The discussion is based on the project's future improvements and scalability of the project. Future improvements can include additional device testings on further household activities that based on Theory section2 emit high VOC pollutants. These activities include indoor renovation processes and a usage of furniture which contains harmful VOCs. These additional tests can highlight even more the effect of household activities on indoor pollutants levels.

Based on all experiments findings an additional improvement could be to design a recommendation guidelines of actions that a household could take when values are higher than recommended. Additionally, these guidelines could include a list of recommendations on which actions to undertake in order to avoid a high pollutants levels indoors.

Due to the project's focus on creating a minimum viable product, it possesses significant scalability potential. The scale up of it could be a personal household device that can measure and warn the user when pollutants levels are higher than recommended. The warning could come in various forms, for instance in a simple form as a LED light that would turn on and blink once values are exceeded. A more complex idea could be a connection between the device and a mobile app that could send the user notifications regarding high pollutants values. The mobile app is a complex idea which can develop in different directions. Another scale up idea would be to include additional sensors in order to measure more indoor pollutants. Particulate matter sensor would be a natural choice as it can measure one of the most common indoor air pollutants. Initial plan was to incorporate this sensor in the device, however due to technical issues it was later emitted.

In conclusion by testing the final device a measurements were conducted to collect data during everyday household activities. The findings of this project highlight the importance of monitoring levels of CO_2 and VOCs in indoor spaces, due to encountered higher pollutant levels than the recommended values while performing certain household tasks. The experiment also highlights the role of a proper ventilation in order to reduce these indoor pollutants.

Table of Contents

1 Introduction

Indoor air has a significant influence on the quality of people's lives. According to the World Health Organization (WHO), poor air quality resulted in 4.2 million deaths in 2016. Numerous studies indicate that indoor air pollutant concentrations are generally two to four times higher than outdoor air pollutant concentrations. The most vulnerable to indoor these pollutants are children, the elderly, and cardiopulmonary patients [15]. The National Human Activity Pattern Survey of the USA has found that on average, adults spend 87% of their time inside enclosed buildings[16]. Hence, our living environment plays a major role in our physical health. Indoor air pollutants include carbon monoxide (CO), carbon dioxide (CO_2), volatile organic compounds (VOCs), particulate matter (PM), and ozone (O3) among others[3]. This paper will mainly focus on CO_2 and total volatile organic compound (tVOC) pollutants. Carbon dioxide (CO_2) is one of the most common indoor emissions with substantial influence on human health[4]. It is an odorless and colorless gas. It is produced naturally by exhaling and as a result of human activities, such as burning gasoline, coal, oil, and wood. The level of CO2 indoors can be influenced by various settings including the number of inhabitants, fresh air entering the area, the size of the room or area, air contaminants (e.g., idling vehicles near air intakes, leaky furnaces, tobacco smoke), the outdoor concentration[17]. Higher CO_2 levels indicates a poorer indoor quality which can result in increased health problems such as headache, dizziness, nausea with levels above 5,000 ppm and asphyxiation which replaces oxygen in the blood with CO_2 around 40,000 ppm which is immediately dangerous to life and health[18][17].
Indoor VOC pollutants are released into the air from various sources such as building materials, cleaning products, and furniture. VOCs can cause, same as CO_2, health problems from temporary mild symptoms such as headaches, fatigue to chronic health effect such as respiratory problems, organ damage or cancer [6]. Due to health risks it is fundamental to monitor indoor air quality and install ventilation systems equipped with accurate and affordable CO2 sensors in residential, commercial, and industrial buildings [19][20]. However, existing sensing technologies are very expensive and have limitations such as complex device fabrication or high power consumption. This is making them impractical for widespread use. Therefore, there is a need for a cost-effective and energy-efficient CO2 sensor[20].

Based on initial research the research question was formulated as:

Research Question: Is it viable to propose an indoor air quality measuring device that can measure CO_2 and tVOC levels and determine how various household activities affect these values?

Having established the negative influences of various domestic indoor pollutants, it is relevant to determine the effects that common indoor activities have on the indoor air quality. I therefore wish to build a minimum viable product(MVP) of a home-usage device that can track the levels of two different pollutants, CO_2 and volatile organic compounds (VOCs). A test will be conducted while performing household tasks such as cooking, cleaning, sleeping among other everyday activities. This experiment will examine what kind of effect these activities have on these indoor pollutants levels.

2 Theory

This section is divided in 2 sections: Air Quality and Smart homes. Air quality section describes the most common air pollutants and how can be indoor air improved. Section Smart homes investigates various techniques and devices that can improve indoor quality.

2.1 Indoor Air Quality

The term "indoor air quality" (IAQ) refers to the air present in enclosed environments such as houses, work spaces, schools, and other premises. IAQ considers various elements like the presence of pollutants, gases, particles, humidity, temperature and ventilation[1].

2.1.1 VOCs

Nowadays regulations for air pollution target outdoor air. On the other hand, indoor air is way less defined and known for air pollution[21]. In order to decrease a risk of indoor health issues it is important to recognise indoor air pollutants[6]. Indoor air pollutants can for instance come from building materials and furniture[21][6]. During building renovations, numerous pollutants are generated, which can have harmful effects on human health. Highly toxic substances, like formaldehyde, are present in many building materials. Formaldehyde is one of the most toxic volatile organic compound (VOC). VOCs are carbon based molecules, very common in air in small quantities. They are toxic to humans in high concentrations over long-term exposure [22]. VOCs mainly result in unpleasant smells, pulmonary irritations, headaches, sore throat, and weakness. Prolonged inhaling of formaldehyde above certain concentrations can generate methanol inside the body, which is capable of damaging the human retina and causing vision problems, reduce sense of smell, obstruct breathing, or even incur memory loss. It can also target the body's immune system, making it more susceptible to various illnesses[7][8]. Exposure to high levels of VOCs during pregnancy can have negative consequences on fetus development, and can result in premature birth [22].
Another significant VOC pollutant is benzene. It has a liquid form commonly found in construction materials such as paints and coatings[7][8]. Other sources can be automobile emissions from house attached garages, stored fuels or environmental tobacco smoke[14]. It can cause severe damage to the respiratory system and skin. It is more damaging for children attacking blood cell causing leukemia. It is therefore categorised as carcinogenic to humans[7][8].

Other indoor air VOCs can be found in cleaning products and personal care products[21][6]. Cleaning is a basic human activity. According to a study conducted in the US, people typically spend approximately 20 to 30 minutes each day on domestic cleaning. Many cleaning products are sources of VOCs that can cause health problems when inhaled[23]. A study from 2015 examined a domestic use of bleach across several European countries and its health impact on children between 6-12 years. The researchers collected data on the children's demographics, household characteristics, use of bleach and other cleaning products and health history of respiratory diseases and infections. The study results showed that frequent use of bleach in the household had negative effect on respiratory infections in children. The authors recommended to limit use of bleach and consider alternative less harmful cleaning products[24]. The World Health Organization (WHO) specifies that a safe indoor level for VOCs is 200 µg/m3 for total VOCs[9].

Burning tobacco products are also a source of harmful pollutant[6]. Environmental tobacco smoke contains hundreds of VOCs[25].

2.1.2 Indoor CO_2

Carbon dioxide CO_2 is also considered as an indoor pollutant. The reason why CO_2 concentrations are higher in indoor environments compared to outdoor environments is because humans generate and release CO_2 through their respiration. Typical outdoor CO2 concentrations are around 380 ppm (particles per million), but in urban areas, outdoor levels can go as high as 500 ppm. However, it is recommended to keep indoor CO2 levels below 1000 ppm. Studies have found that when the levels of CO_2 in indoor spaces are higher than normal, people feel like the air quality is poor, and they are more likely to experience short-term health problems like headaches and fatigue which can lead to a lower work productivity. More extreme health symptoms can occur when CO2 levels exceeds 5000 ppm, these symptoms can include breathlessness, elevated heart rate, cognitive confusion, and, in severe instances, even loss of consciousness[5].

2.1.3 Other indoor pollutants

Particulate matter (PM) pollutants refer to small particles in the air that can be harmful when inhaled into the lungs from both indoor and outdoor air. Indoor PM usually occurs from cooking, cleaning, smoking, burning candles and not least from ordinary dust accumulation. Outdoor PM can come from vehicle exhaust, industrial processes, wildfires and plants' pollen. Exposure to PM is linked to a range of health problems like inflammation resulting in diseases including respiratory infections, heart disease, stroke, lung cancer, asthma and other respiratory conditions[10]. A high concentration of particulate matter pollutants is mainly found in developing countries. Based on a global map that uses satellite data to measure PM2.5 (particulate matter that has a diameter of 2.5 micrometers or less), the regions with the highest pollution levels worldwide are Northern and Eastern China as well as Northern India[26].

Secondly, outdoor air pollutants such as radon, pesticides, air pollution also influences indoor air [6]. Industrial cities have higher pollution due to the high concentration of pollutants generated by industrial processes. A study conducted in Beijing China from 2008 found that indoor air pollution was significantly higher in homes located near heavy traffic areas and industrial zones, with increased levels of PM, VOCs, and nitrogen dioxide (NO2) [11].

Additionally, the pollutants in indoor air can be more concentrated due to high indoor temperature, high humidity and poor ventilation [6].

2.1.4 Improving indoor air

To improve indoor air it is important to have safe values of CO_2. Increase ventilation with a use of mechanical ventilation systems or opening windows and doors can reduce CO_2 levels. Because outdoor CO_2 levels are lower the outdoor air will dilute the indoor concentration up to 70%[13]. However, in industrial cities with high outdoor CO_2 levels opening windows wouldn't be very efficient. VOC contamination sources from outdoor air, can be emissions from chemical plants, or from vapors from underground spills[22]. Modern technologies such as air purifiers (see subsection 2.2.2), that use absorption process which removes CO_2 from indoor air, can reduce CO_2 by up to 50% [12]. The use of indoor plants is also one of efficient ways to decrease indoor CO_2 levels. Plants consume CO_2 and emit oxygen. Studies have shown that having plants in indoor areas can reduce CO_2 levels by up to 25%[27].

One of the main VOC pollutants is often building materials. Therefore, it's important to make smart choices when selecting building materials as some can contain harmful gases. Choosing materials that are environment friendly and non-polluting should be a priority to help maintaining a healthy indoor environment and air quality[7]. Another way to reduce indoor VOC is to increase ventilation while using products that release VOCs. Proper ventilation can help to reduce indoor pollutant levels and make the indoor environment safer. It's important to make sure that the ventilation lasts long enough to remove harmful gases, because it can take a while to reduce the pollutants[7]. It is also important to identify which daily usage products contain harmful VOCs such as formaldehyde and replace them with safer alternatives or reduce their usage. Storing partially used containers of paints and other similar materials should be outside living quarters[14].

The most effective way to remove VOCs from indoor air is to use air purifiers with activated carbon filters[22] (see section 2.2.2.

2.2 Smart homes

When the idea of the smart home was first introduced, it was supposed to help people use energy more efficiently during the transition towards sustainable energy. The term of a 'smart home' is nowadays associated with improving your homes by integrating home automation systems, household appliances, as well as communication and entertainment electronics[28]. More and more people are conscious about air pollution and its consequences for their health[29][28]. The outdoor air in many large cities is polluted with harmful pollutants, and it is therefore not recommended to bring in air from outside without filtering it first because it can make the indoor air even more polluted[28]. Hence indoor air quality sensors and purifiers are increasing in demand[28][29]. The higher demand might be also influenced by the recent worldwide experience with lockdowns, self-quarantines and use of face masks due to the COVID-19 pandemic. Due all the mentioned reasons, the indoor air sensors availability and variety is increasing[29].

2.2.1 Indoor air detector

Indoor air detectors are devices that measure the concentration of various pollutants and chemicals in indoor air. The field of air quality sensor technology is rapidly expanding and has the potential to enhance the practicality, reliability, and cost-effectiveness of air pollution measurements over time. Low-cost air quality sensor products are becoming increasingly available on the market. Some detectors are standalone units, while others are part of air purifiers or temperature regulation technologies such as air conditioning units, heating and ventilation systems[15].

After examining the indoor air detector market, there is available a number of devices that are capable of measuring one or more parameters related to indoor air quality. For instance device called Wave Plus by Airthings can measure both VOC and CO_2 levels and its price is around 1700DKK[30] which is an average price of these devices.

2.2.2 Air purifier

Air purifiers are electronic devices designed to remove harmful pollutants, allergens, and other particles from indoor air. The demand for home-usage air purifiers is increasing partly due to the rise of respiratory diseases[31]. The US Department of Energy states that an air purifier that utilizes HEPA (High-Efficiency Particulate Air) filters can eliminate at least 99.97% of dust, pollen, mold, bacteria, and any particulate airborne particles that are 0.3 micrometers or bigger in diameter[26][32]. According to recent clinical studies utilizing HEPA purifiers can result in health benefits such as decreased asthma symptoms, lower levels of inflammation and heart disease markers, and a lower occurrence of fungus/mold infection among humans[33].

Air purifiers that use other than HEPA filters are targeting different types of pollutants. Namely, activated carbon filters are designed to absorb volatile organic compounds (VOCs), cigarette smoke, odors from cooking, pets and other sources. They work by pulling air into an activated carbon purifier, it flows through a filter containing activated carbon. As the air moves through the carbon filter, any harmful pollutants and chemicals present in the air are captured in the carbon's pores, which allows clean air to flow through and spread back into the room.[34].

Looking at economic perspective of air purifiers, a study from 2016 conducted in China investigated how air pollution levels affect consumer behavior, specifically their willingness to pay for home-use air purifiers. According to this study results, the demand for air purifiers increases, when the air quality worsens and consumers are keen to pay a higher price for a better-quality air purifiers [26].

3 Design & Tools

This section describes the device components, their limitations and comparisons as well as device's design and assembly.

3.1 Primary components

A range of electronic components were used to monitor and log the air quality parameters. These components are listed in table 1, and their individual functions in this project are further stated in the following subsections 3.2 3.3.

Component	Model (Brand)	Functionality
Microcontroller unit (MCU)	ESP32-DevKitC (AZ-Delivery)	Used as a MCU to control sensors, OLED monitor and to interface with desktop software.
CO2 sensor	SCD30 (Sensirion)	This sensor is mainly used to measure CO2 values in an environment, but it also measures temperature and relative humidity. Its CO2 measurement range goes from 400 ppm to 10,000 ppm with an accuracy of ± 30 ppm.
tVOC sensor	MiCS-VZ-89TE (SGX Sensortech)	This sensor measures total volatile organic compounds (tVOC) in an environment with a high sensitivity. It allegedly can also estimate a CO2-equivalent (eCO2) measure, but the accuracy of this measurement seems very low.
0.96" OLED display	128x64 resolution and SSD1306 chip (AZ-Delivery)	A generic OLED display that utilizes the SSD1306 chip to control 128x68 pixels.

Table 1: List of primary device components.

3.2 CO_2 sensor characteristics

The SCD30 CO2 sensor is made by the Swiss company Sensirion and generally holds acclaim as a very reliable and precise sensor. When concerned with human health, CO2 measurement accuracy is usually acceptable within 100 ppm, so with an accuracy of 30 ppm the sensor is more than enough to cover the CO2 focus area of this project. The sensor uses non-dispersive infrared (NDIR) spectroscopy to measure the density of CO2 molecules in the air, which in principle means that it shines infrared light through probed air and detects the light spectrum of the infrared light after having moved through the air. The detection values will vary according to the composition of molecules in the probed air, and thereby also with the ranged composition of CO2 molecules which gives the sensor the ability to measure the parts-per-million concentration of carbon dioxide in the air[35].

Figure 3: Sensirion SCD30 - CO_2 sensor[35]

The sensor can be interfaced by a microcontroller unit (MCU) with digital data transfer protocols like UART or I2C. For this project, I2C was used, because the available software libraries prescribed this interface method and due to its simple two-wire connection. Inter-Integrated Circuit (I2C) is a serial communication method where multiple slaves (in this case sensors) can be connected to a master (ESP32 MCU) via two wires data transfer wires: A serial-clock (SCL) wire that the master uses to synchronize the baud rate, which is the rate at which information is transferred in a communication channel, as well as a serial-data (SDA) wire that the sensor(s) use to transmit data to the master[36].

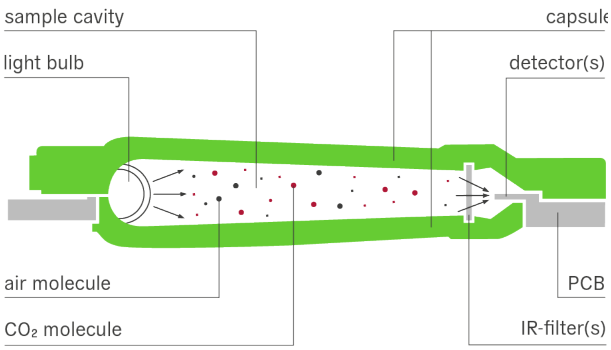


Figure 4: The principle of non-dispersive infrared (NDIR) spectroscopy[37]

3.3 tVOC sensor characteristics

The MiCS-VZ-89TE tVOC sensor is made by the Swiss company SGX Sensortech and uses metal oxide semiconductor (MOS) technology to approximate the molecule composition of probed air. The MOS sensor technology is relatively cost-effective, but this comes with a penalty on accuracy and reliability in distinguishing molecules. MOS works by measuring the electrical resistance change of the metal oxide caused by adsorption of gases, which gives the MiCS-VZ-89TE sensor the ability to approximate the density of total VOCs in its surrounding air in parts-per-billion (ppb)[38].

This sensor advantages are low power, calibration-free, high sensitivity and high resistance to shocks and vibrations[39].

The tVOC sensor can be interfaced by a microcontroller unit (MCU) that reads pulse-width-modulation (PWM), which essentially consists of voltage spikes (pulses) that are modulated to a certain duration (width) according to the data values that are being transmitted by the sensor. For the MiCS-VZ-89TE, these spikes are multiplexed, meaning that there are several segments of the voltage pulse-width that contain individual data. When the voltage spikes take up 5-45% of a time period of 33.3 milliseconds, i.e., a segment or plex, the sensor is indicating that it is showing tVOC values between 0 and 1000 ppb. Similarly, when the voltage spikes take up 55-95% of a measured time range, the sensor is indicating that it is showing eCO_2 values between 400 and 2000 ppm[40]. The approach to gather measurements from the sensor via its PWM output is addressed further in the section Implementation and testing 4.

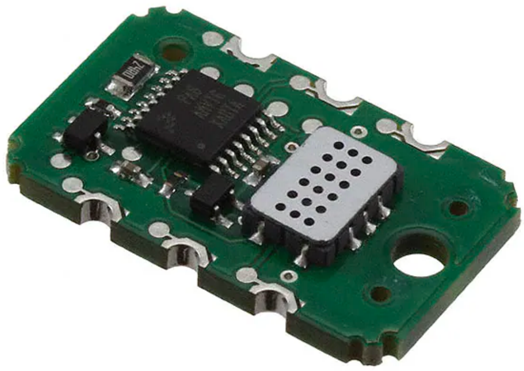


Figure 5: MiCS-VZ-89TE - VOC sensor[40]

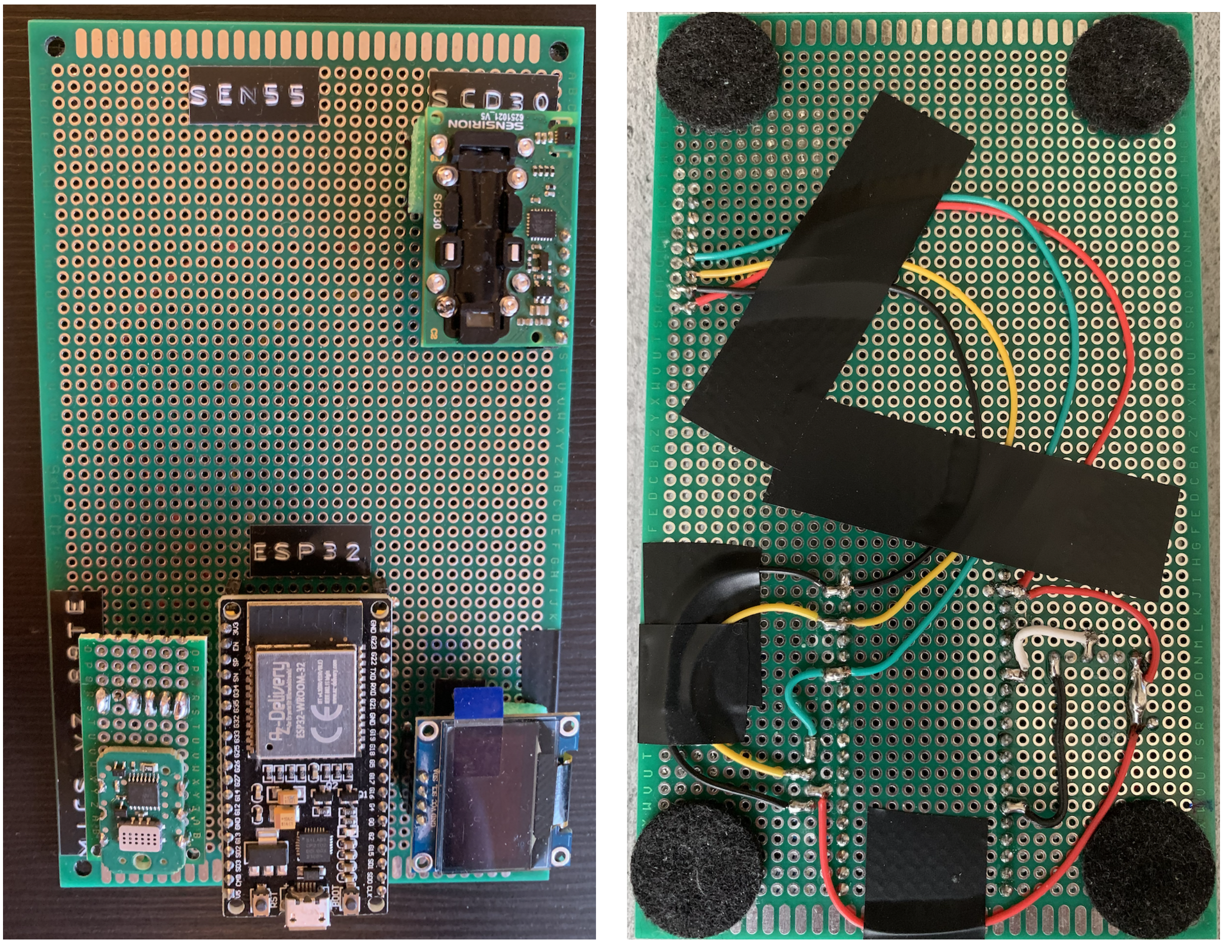
3.4 Component limitations and comparisons

It is important to note that the MOS sensor technology used in the MiCS-VZ-89TE tVOC sensor is highly affected by factors such as humidity, temperature and has cross-sensitivity to molecules with similar electrical properties as VOCs. Therefore, MOS sensors should not be used for scientific purposes where accuracy and reliability are crucial. In this project, we will therefore also only use the tVOC sensor measurements in conjunction with the CO2 sensor measurements to give an indication of when air quality is being reduced.

Since the SCD30 sensor is fully calibrated and linearized upon delivery (and can therefore be relied upon to give baseline measurements for evaluating other sensors), it was used to validate the eCO_2 measurements of the MiCS-VZ89TE sensor, which led to the conclusion that the latter sensor was much too unreliable for any CO_2 measurements. The MiCS-VZ89TE sensor is therefore only used for tVOC measurements, which must also be regarded with some accuracy reservations. As we will see in section Implementation and testing4, though, the tVOC readings of the MiCS-VZ89TE correlate significantly with CO_2 readings of the SCD30 sensor when performing certain air quality-reducing activities.

For this project, the SCD30 sensor was purchased on the private consumer market for around €50, and due to this relative high cost compared to other available CO_2 with decent accuracy and measurement ranges, it is mainly used in high-end consumer products. However, for this project, accuracy of CO2 measurements was crucial to conduct a valid proof-of-concept, where measurement results could be fully trusted, so it was prioritized to implement this specific sensor into the project. The MiCS-VZ89TE sensor is about half the price of the SCD30, which is presumably due to the simplicity of the MOS sensor technology compared to NDIR, which requires a probing chamber with an infrared light source and several light filters and detectors. The cost of the SCD30 sensor was 50.9€ (DKK379), the cost of MICS-VZ-89TE was 27.0€ (DKK201) and 7.56€ (DKK56) for ESP32. The total cost for these main components was 85.46€ (DKK636).

3.5 Design and assembly



(a) Final board from front side (b) Final board from back side

Figure 6: Final board

Since this project is aiming to propose a viable project for measuring air quality in a domestic setup, it was prioritized to connect the individual components as robustly as possible without relying on too many prototyping methods (e.g., breakout boards and molex pin cables). Portability was also a concern, since I wanted to monitor several rooms and be able to move the device around without being concerned with pins disconnecting. Due to these factors, I started by laying out the components on the PCB to estimate the most suitable placements, then I soldered the pin header connectors to the board and connected the pins on the bottom side of the board to their designated spots. The bottom-side connection of pins was done with 2.62 mm2 gauge wire. On overview of the components used for assembly of the device can be seen in Table 2 below.

Component	Specifications	Functionality
Printed circuit board (PCB)	90x150 mm with soldering hole spacing of 2.54 mm.	PCB for connecting components with soldered joints and connectors.
Female pin header connectors	Female connectors for soldering onto 2.54 mm PCB: 2x19 pins for ESP32 MCU, 1x6 pins for tVOC sensor, 1x4 pins for OLED display, 1x7 pins for CO2 sensor.	Used to connect the primary components to the PCB so that they can be easily removed for replacement or isolated tampering.
Male pin header connectors	Male connectors for soldering onto ESP32 MCU, OLED display and sensors. Same pin layout as with the female pin header connectors.	Used to make the components compatible for electrical connections.
Electrical wires	Wire with a 2.62 mm2 gauge (13 AWG) in assorted colors: Red, black, yellow, green, and white.	Used for wiring components and connectors together.
Electrical wires	Wire with a 2.62 mm2 gauge (13 AWG) in assorted colors: Red, black, yellow, green, and white.	Used for wiring components and connectors together.
Solder	Sn60Pb40 as 1 mm thread.	Used for soldering wires and connector pins together.
Soldering iron	Soldering iron capable of reaching around 350°C.	Used to melt solder to make soldered joints.
Wire stripper	E.g., Weicon No.5	Used to strip insulation from wire to expose conductive material.
USB cable	USB-A to micro USB	Used to interface the ESP32 MCU with a PC.

Table 2: List of secondary device components and tools required for assembly.

4 Implementation & Testing

This section focuses on the device building, device code description and device testing.

4.1 Embedded software development and microcontroller interfacing

For developing the embedded firmware for the proposed air monitoring device, the integrated development environment (IDE) PlatformIO[41] was used. PlatformIO is essentially an extension for the source-code editor Visual Studio Code (VS Code)[42] and allows for relatively easy development of embedded software for common microcontroller units like the ESP32. Commonly, the Arduino IDE is used for developing embedded software (especially for the well-known Arduino development boards), but as this project was expected to require much trial-and-error in regard to testing various sensors and hardware constellations, the ease-of-use of the PlatformIO library management functionality was chosen. An additional benefit of using PlatformIO is also that you get the full live-feedback functionality of VS Code on syntax errors, reference errors and other common coding mistakes, so that you don't have to wait for the compiler to point out these errors. Starting to use PlatformIO for embedded software development only requires installing VS Code, adding the PlatformIO extension, and then simply creating a new project in the PlatformIO interface. From there, the coding follows basic C/C++ project structure with an .INI file for defining board platform, and dependencies, as well as a main .CPP file consisting of dependency reference declarations, setup block and loop block.

To interface a developer's computer with the ESP32 a USB cable is needed. From there, one only needs to use the PlatformIO functionalities to build (compile), upload or monitor runtime. The upload process is much more streamlined than with Arduino, and one only needs to plug in the device via USB, and PlatformIO automatically detects the serial port for interfacing upload and monitoring.

4.2 CO_2 sensor implementation

I have created a basic structure of the Air quality detector hub (based on ESP32 micro controller), where I have connected the CO_2 sensor to the ESP32. The SCD30 CO2 sensor is connected to the ESP32 via four cables: Power cables (positive and negative, indicated by red and black), as well as two cables for I_2C interfacing (clock synchronization and serial-data). As mentioned in the previous section, I_2C is a data transfer protocol that allows hardware devices to communicate with each other over two wires (see figure 7). The board with CO_2 sensor product can be seen in figure 8.

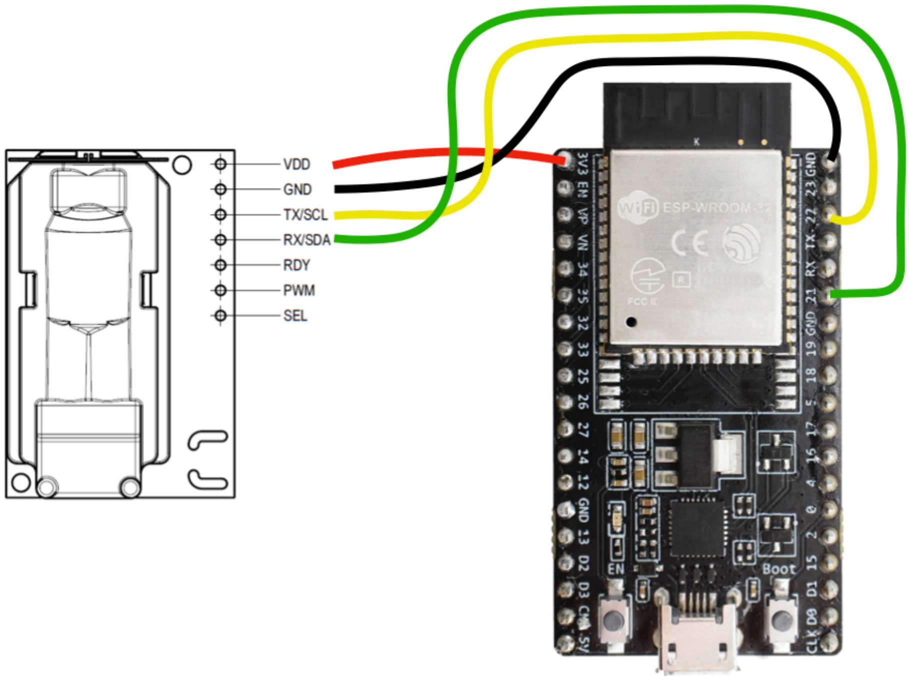
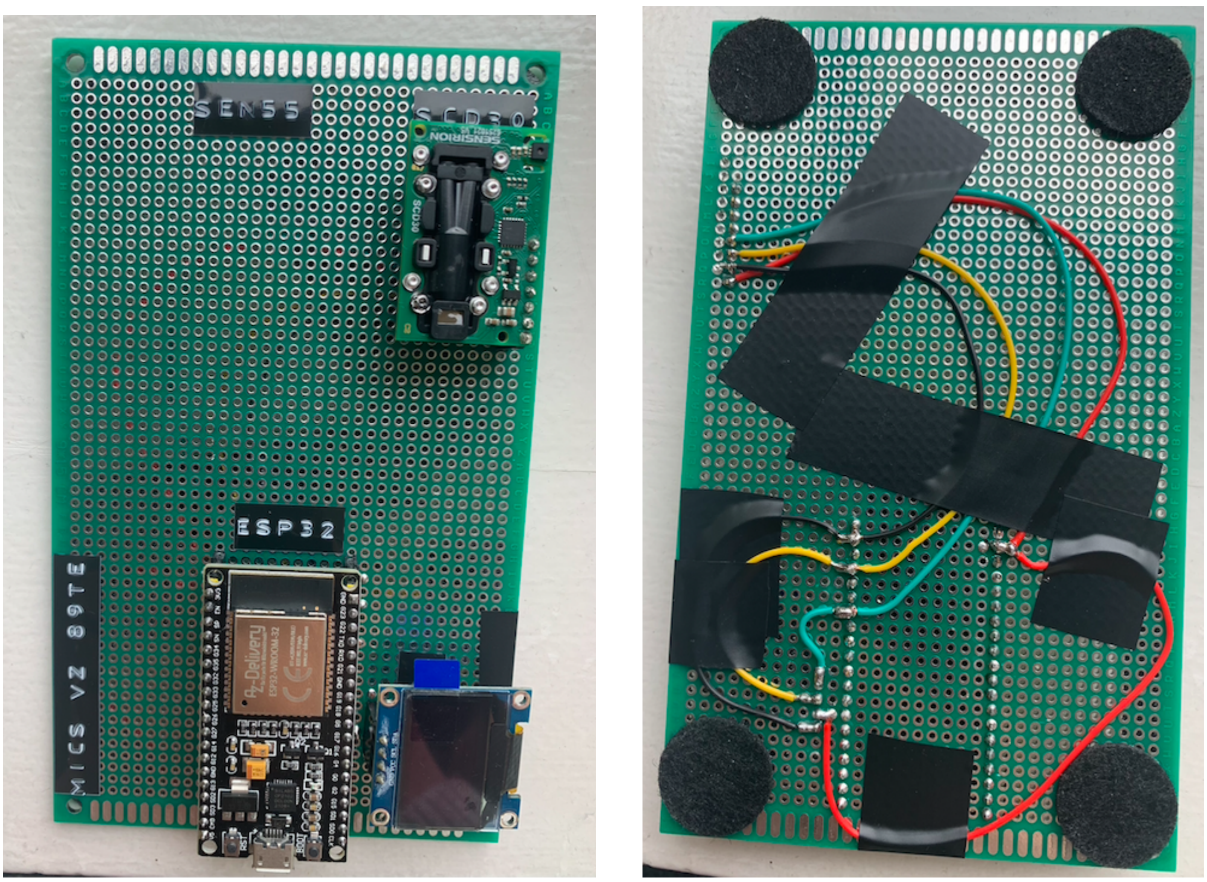


Figure 7: Sensirion wiring with ESP32 [43]



(a) Board from front side (b) Board from back side

Figure 8: My board with CO_2 and monitor

Data readings from the SCD30 to the ESP32 were managed via the SparkFun SCD30 Arduino Library[44] that was implemented into the software by the build-in PlatformIO library management functionality. From there, one simply makes a reference to the SparkFun_SCD30_Arduino_Library.h package in the main .CPP file, initialize the sensor in the setup code block, and then getting the sensor readings by calling sensor class methods. All of these steps are easily implemented by following the available instructions upon invoking the library in PlatformIO, and the general outline of the code is illustrated in figure 9. It is important to note, however, that given the relative complexity of sending/receiving data over I_2C, what we see here is only the top-most level of abstraction, and there is a lot more going on in the library classes that takes care of the I_2C protocol and sensor management.

```
#include "SparkFun_SCD30_Arduino_Library.h"

SCD30 scd30;

int co2 = 0;
int temperatureC = 0;
int humidity = 0;

void setup() {

}

void loop() {
  if (scd30.dataAvailable())
  {
    co2 = scd30.getCO2();
    temperatureC = scd30.getTemperature();
    humidity = scd30.getHumidity();
  }
}
```

Figure 9: General outline of SCD30 software implementation with the setup block omitted

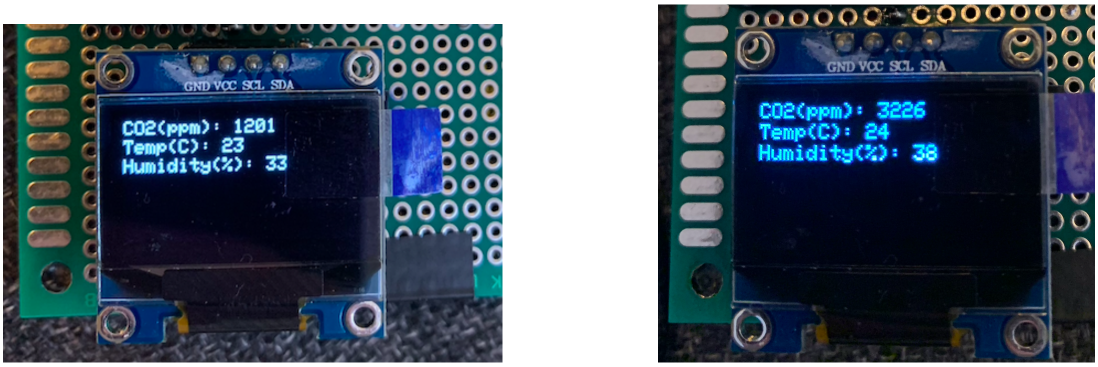
4.3 Monitor implementation and preliminary test of CO2 sensor

In order to both debug and test the device more efficiently while adding sensor components and building it, an OLED display monitor was added early in the building process. The monitor will also serve as a way to show sensor readings to the user in its final version. Same as with the CO2 sensor, implementing the display was as simple as identifying a suitable library in the PlatformIO library menu and invoking it into the project. The display uses an SSD1306 microchip to control several OLED pixels, and user can interface with this chip via I_2C, which means that only need four wires are needed in total to implement the display into the device. To show data on the display, two Adafruit libraries are used, as shown in figure 10 below. In the code, you will also notice that the CO2 measurement can be shown on the display by setting the cursor to a specific starting point, and then letting the libraries handle the text formatting on the display.

```
#include "SparkFun_SCD30_Arduino_Library.h"
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

SCD30 scd30;
int co2 = 0;

void setup() {

}

void loop() {
  if (scd30.dataAvailable())
  {
    co2 = scd30.getCO2();

    display.clearDisplay();

    display.setCursor(0, 0);
    display.println("CO2(ppm): ");
    display.setCursor(60, 0);
    display.println(String(co2));

    display.display();
  }
}
```

Figure 10: General outline of code for showing sensor value on display (setup block omitted for readability).

After having implemented both the SCD30 and OLED display into the project, I could now start monitoring CO2 values in real-time. I initially conducted a simple test to ensure the CO2 sensor functionality by exhaling air in close distance to the sensor. I observed on the monitor that the CO2 level almost tripled in comparison to when breathing further away from it. This effect is illustrated in figure 11a by data points with larger distance from the sensor and figure 11b for data points with close distance exhaling.



(a) Data points with larger distance to sensor (b) Data points after close exhaling

Figure 11: Testing of CO_2 levels

I have conducted several more preliminary tests with the sensor constellation, and noted that I was able to have the sensor reading reach the maximum of 40,000 ppm when exhaling continuously very close to the sensor. The ambient sensor readings with windows opened and without any human proximity to the sensor showed values around 600-800 ppm, which was expected based on the theory 2.1.2.

4.4 tVOC sensor implementation

Implementing the MiCS-VZ89TE tVOC sensor into the project proved to be more difficult than with the SCD30 and OLED display, as no libraries supporting I_2C for the sensor could be found. Instead I had to rely on pulse-width-modulation (PWM) interfacing, which required to implement software to monitor voltage spikes and translate these into sensor readings. A benefit of PWM is that it only requires one wire for data transfer, as we are simply reading an analog value from a sensor pin on the MiCS-VZ89TE. The C++ class to read the tVOC sensor values is documented in Apendix 8.

4.5 Device testing

In order to meaningfully comparing sensor readings to activities, I needed to log the readings for later graphing. The logging was done by using the PlatformIO monitor functionality to output serial data from the ESP32 to a computer, where the data could then be stored in a file. An approach to gather the data into CSV format was implemented into the C++ code so that the printed serial output would give elapsed milliseconds and all the sensor data separated by semicolon. The log file on the computer could then easily be converted into CSV, which could be later used to graph the data.

For the graphing, I used Python[45], as I was familiar with plotting data via this programming language. In figure 12 below, you will notice that we have simply imported the CSV data into a Pandas dataframe[46] , where we then reformat the elapsed millisecond readings into specific times by setting a static start time for the initialization of the sensor readings.

```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt

df = pd.read_csv(r'C:\Users\tboe\OneDrive\Desktop\T BA\may-22-data.csv', sep=';')

df['time'] = dt.datetime.strptime('22/05/23 20:40:00', '%d/%m/%y %H:%M:%S')
df['time'] = df['time'] + pd.to_timedelta(df['ms'], 'milliseconds')

fig, ax1 = plt.subplots()

# Make a nice time format in the plots.
xformatter = mdates.DateFormatter('%H:%M')
plt.gcf().axes[0].xaxis.set_major_formatter(xformatter)

# CO2 data.
color1='blue'
ax1.set_ylabel('CO\u2082 (ppm)', color=color1)
ax1.plot(df['time'], df['CO2'], color=color1)
ax1.tick_params(axis='y', labelcolor=color1)

# Instantiate a second axis that shares the same x-axis.
ax2 = ax1.twinx()

# tVOC data.
color2='red'
ax2.set_ylabel('tVOC (pbb)')
ax2.plot(df['time'], df['tVOC_mics'], color=color2)
ax2.tick_params(axis='y', labelcolor=color2)

fig.tight_layout()

plt.show()
```

Figure 12: Python code to plot sensor reading data.

After having reformatted the time series, I then use Matplotlib[47] to finally plot the data into a graph with the time series on the horizontal axis and the tVOC/CO2 readings on the vertical axis. This subsection contains a visualization of a 31-hour device test with both sensors that was performed in the author's 35m2 studio apartment in the middle of Copenhagen central area, occupied by two people. Common daily household activities were performed during this time which is documented in the table 3.

Time	Activity
14:21	Device turned on (2 windows opened).
16:20	Toaster usage.
18:53	Closing window.
18:55 - 19:20	Cleaning oven.
19:20 - 19:30	Dusting.
19:27	Opening window.
19:35	Heating food in the oven.
21:05	Closing window.
09:34	Brewing filter coffee.
11:00	Toaster usage.
12:00	Opening window.
12:30 - 13:03	Robot vacuum cleaning.
13:30	Hanging washed laundry.
15:30 - 16:10	Grocery shopping outside the apartment.
16:30 - 17:30	Preparing chicken in the oven.
18:32	Closing a window.

Table 3: List of activities with times of bigger experiment.

The graph of the device readings can be found in figure 13. In this graph CO_2 levels are visualized with blue

color and tVOC levels with red color. It is evident that daily activities such as opening/closing windows, using electronic devices such as coffee machine, toaster and oven influenced the levels of both CO_2 and VOC. The first usage of toaster around 16:20 has a significant tVOC spike which seems like an incorrect reading due to unknown reasons and will be disregarded. The second usage of toaster around 11:00 shows way more clear readings and shows a 100µg/m3 increase of tVOC. Cleaning of the oven around 19:00 with a usage of cleaning products highly influenced tVOC values with almost triple increase with tVOC of around 550µg/m3 which is a harmful value. tVOC levels were also highly influenced by cooking food in the oven with the highest spike of this experiment with the highest value of almost 800µg/m3.
The influence of manual ventilation (window opening/closing) is clearly visible in this experiment readings in both CO_2 and tVOC levels. For example, during night when no activities besides sleeping occurred, the windows were closed and tVOC levels very slowly accumulating. On the other hand CO_2 levels during this period are decreasing due to unclear reasons. On the other hand, after sleeping period, once the windows were opened again the CO_2 levels decreased by approximately 400ppm. Afterwards, when the windows were closed again at 18:32 both CO_2 and tVOC levels increases exponentially.

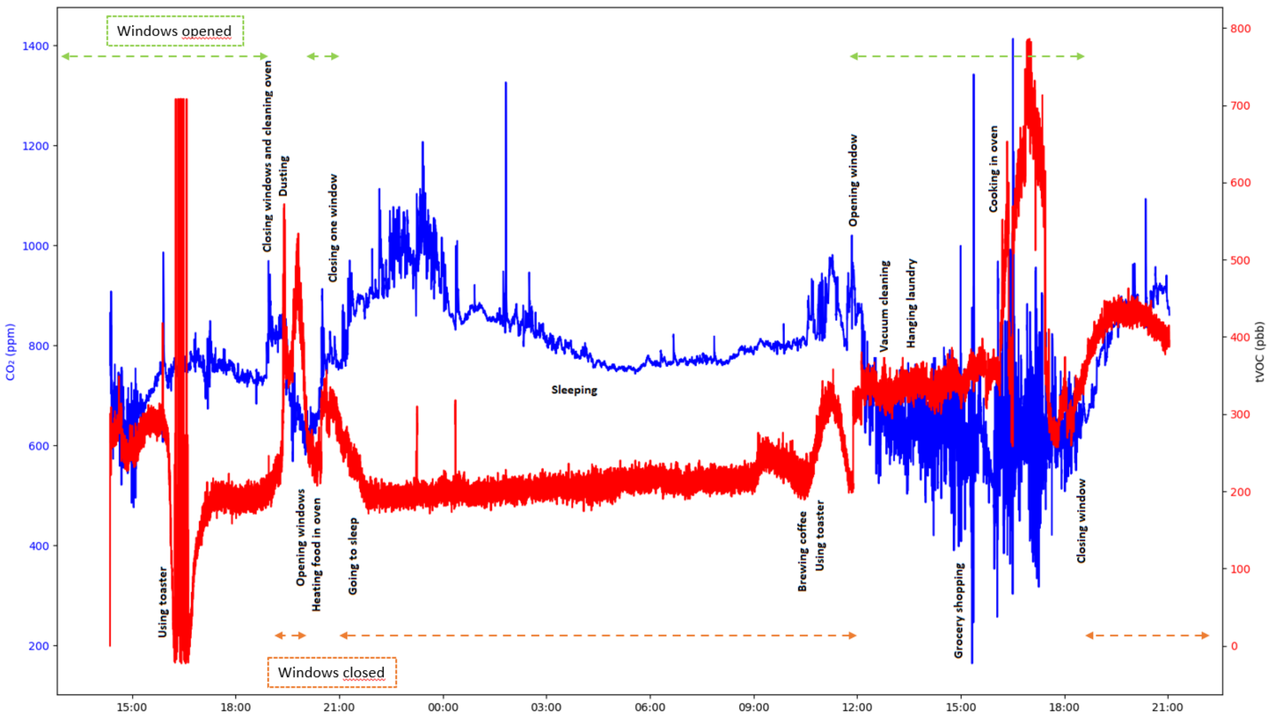


Figure 13: Graph showing the 31 hour monitoring

A shorter experiment was performed in order to test a usage of toaster which showed a questionable results in the previous experiment 13. This shorted experiment's length is 60 minutes starting with closing windows and finishing with closing of windows in last 10 minutes of the experiment. See table 4 for overview of performed activities during the experiment.

Time	Activity
20:39	Closing window.
20:40	Device turned on.
20:49	Kettle turned on.
21:07 - 21:11	Toasting bread in toaster.
21:19	Opening 2 windows.

Table 4: List of activities with times of smaller experiment.

The readings of this experiment are visually displayed in figure 14. The windows were closed to see the effect of 2 electronic cooking devices, kettle and toaster. When the kettle is turned on the CO_2 levels are increasing by 100ppm, but this can be also influenced by recent closure of windows. tVOC levels show only a very small increase which is also possibly influenced by recent window closure.

The usage of toaster shows similar results are the usage of kettle with a little increase of tVOC but no significant change of CO_2 levels.

In conclusion this shorted experiment is a good demonstration of a high influence of manual ventilation on both indoor pollutants. Once the windows are opened then both values drop significantly.

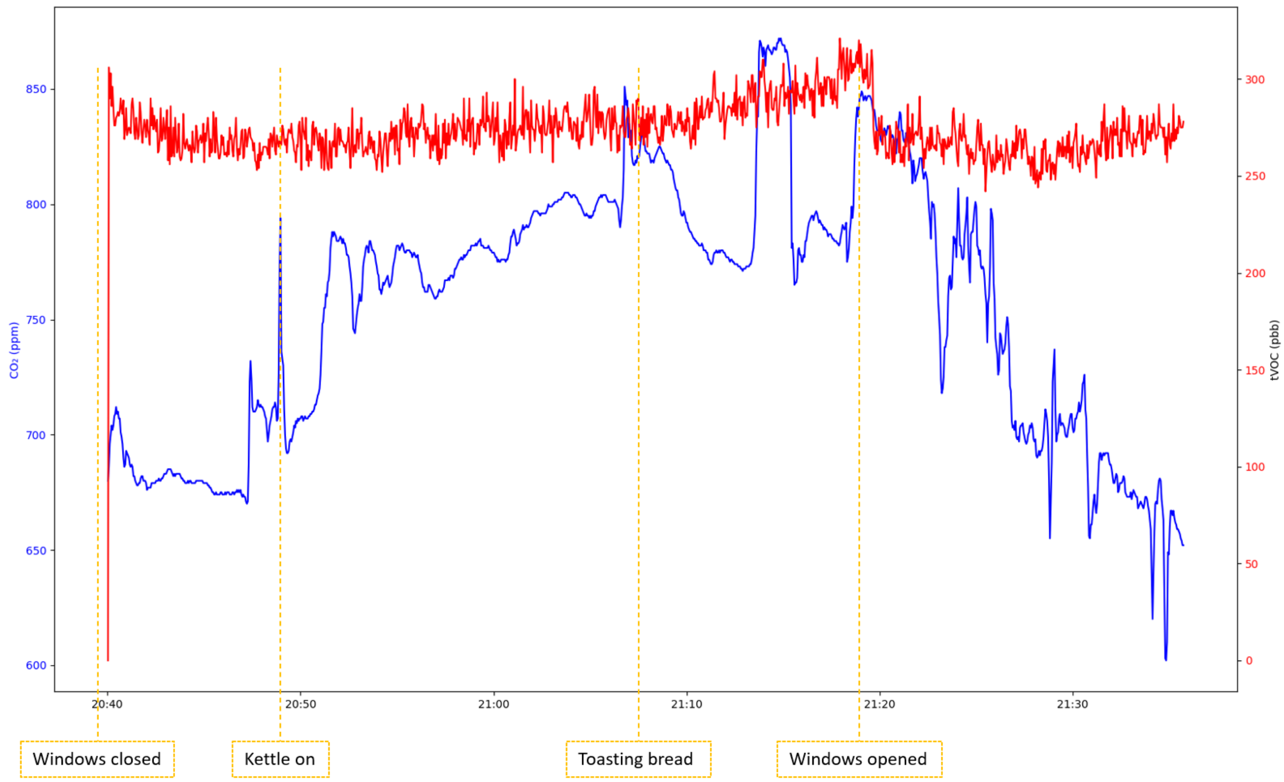


Figure 14: Graph showing the smaller monitoring

5 Discussion

This section discusses a possible project improvements and scalability of our program.

5.1 Project improvements

The area of this project which could be improved is testing. More testing experiments could have been performed in order to test a bigger variety of household tasks that can potentially have an influence on CO2 and VOC levels. As mentioned in Section Theory2 under subsection Indoor Air Quality2.1, one of the main VOC sources are building materials and construction materials. An experiment could have been performed on a construction environment while renovating, painting or cleaning with the use of a different cleaning products in the indoor space. This experiment findings could have demonstrated how this activities have an impact on pollutants levels in compare to other already performed measurements.

Another possible improvement could be analysing the variations in the pollutants across household activities and determining the effect that these activities indirectly have on people's health, as well as establish some tentative guidelines that can help improve indoor air quality. These improvement recommendation could be formulated on the basis of smaller experiments like the effects of opening a window while cooking, or having the bedroom door either opened or closes during sleep.

5.2 Scalability

5.2.1 Particulate matter sensor

In terms of scalability of this project, additional sensors can be installed to measure extra air pollutants. One of the major group of pollutants is particulate matter pollutants, therefore a PM sensor would be a natural next choice. An affordable and good performing sensor could be, same as CO2 sensor, from Sensirion series. It is called Sensirion SEN5x[48].

Figure 15: Sensirion SEN5x[48]

5.2.2 Smart home integration

As mentioned in Theory section2, it is essential that we prioritize finding ways to create healthy living environments in areas with poor air quality[28]. Smart home integration between indoor air sensors and air improving devices is therefore vital. I have created a minimum viable device that is highly scalable and can be implemented in a different directions. I came up with 3 specific ideas how to scale up my indoor air quality sensor device. First idea is to create an app that will be connected with the device. The app would receive live data from the device and notify the user of high values, recommending to take actions such as opening windows or turning on an air purifier.
The second idea is to build an integration between the device and an air improving devices, such as air purifier, ozone cleaner or air ventilation. Once the values gets close to harmful levels the sensors would send a message to a device that would control the turning on of an air purifier. However, this idea is very complex and would need an additional research. On figure 16 below you can see a visual representation of such a smart home integration.

Lastly, the simple design on my device makes it scalable for design improvements. In order to make this device more suitable for it to become a personal home air detector it requires a user friendly design where possibilities are endless. On the top of that, extra design components could be implemented, such as LED light that could warn the user of a high pollutants levels in his home or a sounds signaling.

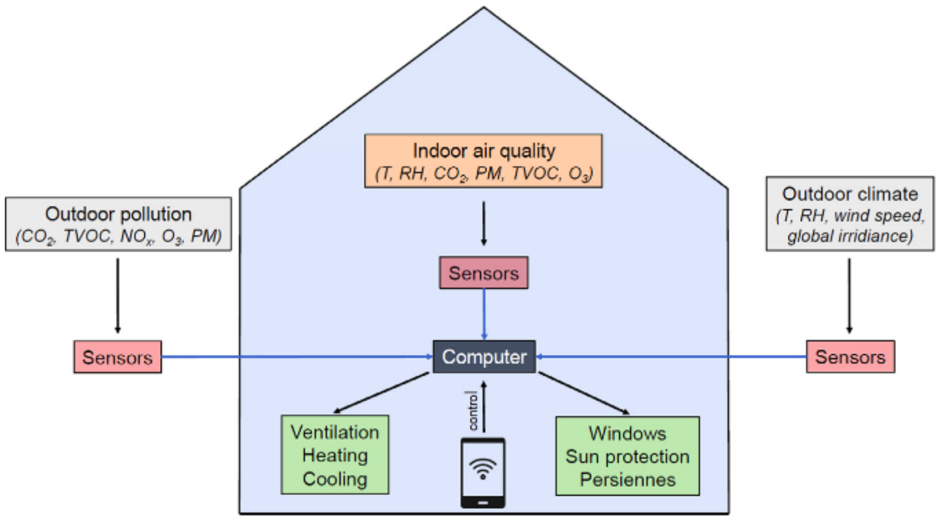


Figure 16: Smart home air controlling[28]

6 Conclusion

By implementing an air quality measuring device and collecting data during daily household activities, valuable insights were gained into the dynamics of indoor air pollution. The results obtained from this project highlight the significance of closely monitoring indoor carbon dioxide and total volatile organic compound levels, and how routine household activities in smaller city studio apartment impact these values. The obtained measurements identify activities that have higher than recommended pollutants values and can serve as a basis for implementing targeted recommendations to improve indoor air quality. Moreover, the findings underline the critical role of proper ventilation in reducing these pollutants. This knowledge can give individuals the ability to make smart decisions regarding ventilation, cleaning products, and lifestyle choices that promote healthier indoor environments.

7 Bibliography

References

[1] *Who Indoor Air Quality Guidelines: Household Fuel Combustion*. World Health Organization, 2014.

[2] URL: https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/sectoral-interventions/household-air-pollution/health-risks.

[3] Alejandro Moreno-Rangel, Tim Sharpe, Filbert Musau, and Gráinne McGill. "Field evaluation of a low-cost indoor air quality monitor to quantify exposure to pollutants in residential environments". In: *Journal of Sensors and Sensor Systems* 7.1 (2018), pp. 373–388.

[4] Gonçalo Marques, Cristina Roque Ferreira, and Rui Pitarma. "Indoor Air Quality Assessment using a CO2 monitoring system based on internet of things". In: *Journal of Medical Systems* 43.3 (Feb. 2019). DOI: 10.1007/s10916-019-1184-x.

[5] Usha Satish, Mark J. Mendell, Krishnamurthy Shekhar, Toshifumi Hotchi, Douglas Sullivan, Siegfried Streufert, and William J. Fisk. "Is co2 an indoor pollutant? direct effects of low-to-moderate CO2 concentrations on human decision-making performance". In: *Environmental Health Perspectives* 120.12 (2012), pp. 1671–1677. DOI: 10.1289/ehp.1104789.

[6] *Introduction to Indoor Air Quality*. Dec. 2022. URL: https://www.epa.gov/indoor-air-quality-iaq/introduction-indoor-air-quality.

[7] Jing Zhuo. "Research on prevention of Indoor Air Environment Pollution by building decoration materials". In: *IOP Conference Series: Earth and Environmental Science* 208 (2018), p. 012107. DOI: 10.1088/1755-1315/208/1/012107.

[8] *Building and paint products*. Nov. 2022. URL: https://www.lung.org/clean-air/at-home/indoor-air-pollutants/building-paint-products.

[9] World Health Organization et al. *WHO guidelines for indoor air quality: selected pollutants*. World Health Organization. Regional Office for Europe, 2010.

[10] Weidong Wu, Yuefei Jin, and Chris Carlsten. "Inflammatory health effects of indoor and outdoor particulate matter". In: *Journal of Allergy and Clinical Immunology* 141.3 (2018), pp. 833–844.

[11] Junfeng Zhang, Tong Zhu, Howard Kipen, Guangfa Wang, Wei Huang, David Rich, Ping Zhu, Yuedan Wang, Shou-En Lu, Pamela Ohman-Strickland, et al. "Cardiorespiratory biomarker responses in healthy young adults to drastic air quality changes surrounding the 2008 Beijing Olympics". In: *Research Report (Health Effects Institute)* 174 (2013), p. 5.

[12] S.C Lee and M Chang. "Indoor and outdoor air quality investigation at schools in Hong Kong". In: *Chemosphere* 41.1-2 (2000), pp. 109–113. DOI: 10.1016/s0045-6535(99)00396-3.

[13] William J. Fisk and Arthur H. Rosenfeld. "Estimates of improved productivity and health from better indoor environments". In: *Indoor Air* 7.3 (1997), pp. 158–172. DOI: 10.1111/j.1600-0668.1997.t01-1-00002.x.

[14] *Volatile Organic Compounds' Impact on Indoor Air Quality.* URL: https://www.epa.gov/indoor-air-quality-iaq/volatile-organic-compounds-impact-indoor-air-quality.

[15] He Zhang and Ravi Srinivasan. "A systematic review of air quality sensors, guidelines, and measurement studies for indoor air quality management". In: *Sustainability* 12.21 (2020), p. 9045.

[16] Eva Höllbacher, Thomas Ters, Cornelia Rieder-Gradinger, and Ewald Srebotnik. "Emissions of indoor air pollutants from six user scenarios in a model room". In: *Atmospheric Environment* 150 (2017), pp. 389–394. DOI: 10.1016/j.atmosenv.2016.11.033.

[17] *Carbon dioxide (CO2).* Mar. 2022. URL: https://www.health.state.mn.us/communities/environment/air/toxins/co2.html.

[18] W Fisk. "Do Indoor CO2 Levels Directly Affect Perceived Air Quality, Health, or Work Performance?" In: *ASHRAE journal* 61.9 (2019), pp. 70–77.

[19] Dikaia E. Saraga. *Indoor Air Quality.* MDPI - Multidisciplinary Digital Publishing Institute, 2020.

[20] Zachary A. Siefker, John N. Hodul, Xikang Zhao, Nikhil Bajaj, Kelly M. Brayton, Carsten Flores-Hansen, George T.-C. Chiu, James E. Braun, Jeffrey F. Rhoads, Bryan W. Boudouris, and et al. "Manipulating polymer composition to create low-cost, high-fidelity sensors for indoor CO2 Monitoring". In: *Scientific Reports* 11.1 (2021). DOI: 10.1038/s41598-021-92181-4.

[21] Dikaia E. Saraga. "Special issue on Indoor Air Quality". In: *Applied Sciences* 10.4 (Feb. 2020), p. 1501. DOI: 10.3390/app10041501.

[22] Shariat Mobasser, Yongli Wager, and Timothy M Dittrich. "Indoor Air Purification of Volatile Organic Compounds (VOCs) Using Activated Carbon, Zeolite, and Organosilica Sorbents". In: *Industrial & Engineering Chemistry Research* 61.20 (2022), pp. 6791–6801.

[23] William W Nazaroff and Charles J Weschler. "Cleaning products and air fresheners: exposure to primary and secondary air pollutants". In: *Atmospheric environment* 38.18 (2004), pp. 2841–2865.

[24] Esmeralda Krop, Dick Heederik, Benoit Nemery, Juha Pekkanen, Anne Hyvärinen, Martin Täubel, and Jan-Paul Zock. "Domestic use of bleach and infections in children: a multicentre cross-sectional study". In: (2015).

[25] Richard J Shaughnessy, TJ McDaniels, and Charles J Weschler. "Indoor chemistry: ozone and volatile organic compounds found in tobacco smoke". In: *Environmental science & technology* 35.13 (2001), pp. 2758–2764.

[26] Koichiro Ito and Shuang Zhang. "Willingness to pay for clean air: Evidence from Air Purifier markets in China". In: (2016). DOI: 10.3386/w22367.

[27] Jane Tarran, Fraser Torpy, and Margaret Burchett. "Use of living pot-plants to cleanse indoor air–research review". In: (2007), pp. 249–256.

[28] Alexandra Schieweck, Erik Uhde, Tunga Salthammer, Lea C. Salthammer, Lidia Morawska, Mandana Mazaheri, and Prashant Kumar. "Smart homes and the control of Indoor Air Quality". In: *Renewable and Sustainable Energy Reviews* 94 (2018), pp. 705–718. DOI: 10.1016/j.rser.2018.05.057.

[29] Donatella Puglisi. "Sensors for Air Quality Monitoring, Indoor and Outdoor". In: *Frontiers in Sensors* 2 (2021), p. 17.

[30] URL: https://www.airthings.com/da/wave-plus.

[31] Ga Hee Lee, Jeong Hee Kim, Sungroul Kim, Sangwoon Lee, and Dae Hyun Lim. "Effects of indoor air purifiers on children with asthma". In: *Yonsei medical journal* 61.4 (2020), pp. 310–316.

[32] *What is a HEPA filter?* URL: https://www.epa.gov/indoor-air-quality-iaq/what-hepa-filter.

[33] Zakir-Hussain Abdul Salam, Rubiyah Binte Karlin, Moi Lin Ling, and Kok Soong Yang. "The impact of portable high-efficiency particulate air filters on the incidence of invasive aspergillosis in a large acute tertiary-care hospital". In: *American Journal of Infection Control* 38.4 (2010). DOI: 10.1016/j.ajic.2009.09.014.

[34] Lance Wallace. "Effectiveness of home air cleaners in reducing indoor levels of particles". In: *Final Report Health Canada Contract# 4500172935* (2008).

[35] *Datasheet Sensirion SCD30 Sensor Module.* URL: https://sensirion.com/media/documents/4EAF6AF8/61652C3C/Sensirion_CO2_Sensors_SCD30_Datasheet.pdf.

[36] Scott Campbell. *Basics of the I2C communication protocol.* Nov. 2021. URL: https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol.

[37] URL: https://sensirion.com/products/technology.

[38] Paul Pickering, Shailja Tewari, and Collin Twanow. *Metal oxide gas sensing material and MEMS process.* Apr. 2018. URL: https://www.fierceelectronics.com/components/metal-oxide-gas-sensing-material-and-mems-process.

[39] URL: https://assets.alliedelec.com/v1595325893/Datasheets/da37d16aab40159b4eadcbee85ac2403.pdf.

[40] URL: https://www.sgxsensortech.com/content/uploads/2023/04/DS-0512-MiCS-VZ-89TE.pdf.

[41] URL: https://docs.platformio.org/en/latest/what-is-platformio.html.

[42] URL: https://code.visualstudio.com/Docs.

[43] *Create your own CO2 Monitor.* URL: https://developer.sensirion.com/archive/tutorials/create-your-own-co2-monitor/.

[44] Sparkfun. *An arduino library for the SCD30 Ndir CO2 sensor.* URL: https://github.com/sparkfun/SparkFun_SCD30_Arduino_Library.

[45] URL: https://docs.python.org/3/.

[46] URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html.

[47] URL: https://matplotlib.org/stable/index.html.

[48] *Sen5x environmental sensor node.* URL: https://developer.sensirion.com/sensirion-products/sen5x-environmental-sensor-node/.

8 Appendix

8.1 Code

8.1.1 platformio

```
; PlatformIO Project Configuration File
;
;   Build options: build flags, source filter
;   Upload options: custom upload port, speed and extra flags
;   Library options: dependencies, extra library storages
;   Advanced options: extra scripting
;
; Please visit documentation for the other options and examples
; https://docs.platformio.org/page/projectconf.html

[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
lib_deps =
    adafruit/Adafruit SSD1306@^2.5.7
    adafruit/Adafruit GFX Library@^1.11.5
    sparkfun/SparkFun SCD30 Arduino Library@^1.0.20
monitor_speed = 115200

monitor_filters = default, log2file
```

Figure 17: Code from platformio file

8.1.2 main

```
#include <Wire.h>
#include "SparkFun_SCD30_Arduino_Library.h"
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

SCD30 scd30;

int co2 = 0;
int temperatureC = 0;
int humidity = 0;
int tVOC = 0;
int eCO2 = 0;
```

Figure 18: Code from main file (part 1)

```
class MiCS_VZ_89TE {
  public:
    int pinVOC = 35;
    float pwmHighBorder = 10.0; // analogRead() will give a value that we need to distinguish between high and low to identify pulses.
    float probeCycles = 15000.0;

    float totalHighCycles_VOC = 0.0;
    float totalLowCycles_VOC = 0.0;
    float totalHighCycles_CO2 = 0.0;
    float totalLowCycles_CO2 = 0.0;

    void Update() {
      float cycles = 0;
      bool newDataVOC = false;
      bool newDataCO2 = false;

      while (cycles < probeCycles) {
        float highCycles = 0.0;
        float lowCycles = 0.0;
        bool firstHigh = false;
        bool firstLow = false;
        bool completedPulse = false;

        while (!completedPulse) {
          int reading = analogRead(pinVOC);
          if (reading > pwmHighBorder) {
            if (firstLow) {
              highCycles++;
              firstHigh = true;
            }
          }
          else {
            firstLow = true;

            if (firstHigh) {
              completedPulse = true;
              float pulsePercentHigh = (highCycles / lowCycles) * 100.0;
```

Figure 19: Code from main file (part 2)

```
                    // VOC:
                    // The duty cycle needs to be below 50% to be a VOC reading (otherwise it is CO2).
                    if (pulsePercentHigh > 4.0 && pulsePercentHigh < 46.0)
                    {
                      if (!newDataVOC)
                      {
                        ResetVOC();
                        newDataVOC = true;
                      }

                      totalHighCycles_VOC = totalHighCycles_VOC + highCycles;
                      totalLowCycles_VOC = totalLowCycles_VOC + lowCycles;
                    }
                    // CO2:
                    else if (pulsePercentHigh > 54.0 && pulsePercentHigh < 96.0) {
                      if (!newDataCO2)
                      {
                        ResetCO2();
                        newDataCO2 = true;
                      }

                      totalHighCycles_CO2 = totalHighCycles_CO2 + highCycles;
                      totalLowCycles_CO2 = totalLowCycles_CO2 + lowCycles;
                    }
                  }

                  lowCycles++;
                }

                cycles++;
              }
            }
          }
```

Figure 20: Code from main file (part 3)

```
    void ResetVOC() {
      totalHighCycles_VOC = 0.0;
      totalLowCycles_VOC = 0.0;
    }

    void ResetCO2() {
      totalHighCycles_CO2 = 0.0;
      totalLowCycles_CO2 = 0.0;
    }

    int GetVOC() {
      if (totalHighCycles_VOC > 0 && totalLowCycles_VOC > 0) {
        float percentHigh = (totalHighCycles_VOC / totalLowCycles_VOC) * 100.0;
        float pbbVOC = (25.0 * percentHigh) - 125.0; // Using linear regression on datasheet output range.
        return (int)pbbVOC;
      }
      else {
        return 0;
      }
    }

    int GetCO2() {
      if (totalHighCycles_CO2 > 0 && totalLowCycles_CO2 > 0) {
        float percentHigh = (totalHighCycles_CO2 / totalLowCycles_CO2) * 100.0;
        float ppmCO2 = (39.99 * percentHigh) - 1800.0;
        return (int)ppmCO2;
      }
      else {
        return 0;
      }
    }
};
```

Figure 21: Code from main file (part 4)

```
MiCS_VZ_89TE VZ89TE;

void setup() {
  Serial.begin(115200);
  Wire.begin();

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }

  delay(2000);
  display.clearDisplay();

  display.setTextSize(1);
  display.setTextColor(WHITE);

  //Start sensor using the Wire port, but disable the auto-calibration
  if (scd30.begin(Wire, false) == false)
  {
      Serial.println("Air sensor not detected. Please check wiring.");
      while (1)
          ;
  }

  uint16_t settingVal; // The settings will be returned in settingVal

  if (scd30.getForcedRecalibration(&settingVal) == true) // Get the setting
  {
    Serial.print("Forced recalibration factor (ppm) is ");
    Serial.println(settingVal);
  }
  else
  {
    Serial.print("getForcedRecalibration failed! Freezing...");
    while (1)
      ; // Do nothing more
  }

  if (scd30.getMeasurementInterval(&settingVal) == true) // Get the setting
  {
    Serial.print("Measurement interval (s) is ");
    Serial.println(settingVal);
  }
```

Figure 22: Code from main file (part 5)

```
else
{
  Serial.print("getMeasurementInterval failed! Freezing...");
  while (1)
    ; // Do nothing more
}

if (scd30.getTemperatureOffset(&settingVal) == true) // Get the setting
{
  Serial.print("Temperature offset (C) is ");
  Serial.println(((float)settingVal) / 100.0, 2);
}
else
{
  Serial.print("getTemperatureOffset failed! Freezing...");
  while (1)
    ; // Do nothing more
}

if (scd30.getAltitudeCompensation(&settingVal) == true) // Get the setting
{
  Serial.print("Altitude offset (m) is ");
  Serial.println(settingVal);
}
else
{
  Serial.print("getAltitudeCompensation failed! Freezing...");
  while (1)
    ; // Do nothing more
}

if (scd30.getFirmwareVersion(&settingVal) == true) // Get the setting
{
  Serial.print("Firmware version is 0x");
  Serial.println(settingVal, HEX);
}
else
{
  Serial.print("getFirmwareVersion! Freezing...");
  while (1)
    ; // Do nothing more
}
```

Figure 23: Code from main file (part 6)

```
    Serial.print("Auto calibration set to ");
    if (scd30.getAutoSelfCalibration() == true)
        Serial.println("true");
    else
        Serial.println("false");

}
```

Figure 24: Code from main file (part 7)

```
void loop() {
  if (scd30.dataAvailable())
  {
    display.clearDisplay();

    Serial.print(millis());
    Serial.print(";");

    co2 = scd30.getCO2();
    Serial.print(co2);
    Serial.print(";");
    display.setCursor(0, 0);
    display.println("CO2(ppm): ");
    display.setCursor(60, 0);
    display.println(String(co2));

    temperatureC = scd30.getTemperature();
    Serial.print(temperatureC);
    Serial.print(";");
    display.setCursor(0, 10);
    display.println("Temp(C): ");
    display.setCursor(54, 10);
    display.println(String(temperatureC));

    humidity = scd30.getHumidity();
    Serial.print(humidity);
    Serial.print(";");
    display.setCursor(0, 20);
    display.println("Humidity(%): ");
    display.setCursor(78, 20);
    display.println(String(humidity));

    tVOC = VZ89TE.GetVOC();
    Serial.print(tVOC);
    Serial.print(";");
    display.setCursor(0, 30);
    display.println("tVOC(ppb): ");
    display.setCursor(66, 30);
    display.println(String(tVOC));

    eCO2 = VZ89TE.GetCO2();
    Serial.print(eCO2);
    Serial.print(";");
    display.setCursor(0, 40);
    display.println("eCO2(ppm): ");
    display.setCursor(66, 40);
    display.println(String(eCO2));
```

Figure 25: Code from main file (part 8)

```
    display.display();

    Serial.println();
  }

  VZ89TE.Update();
}
```

Figure 26: Code from main file (part 9)

8.1.3 Graphs code

```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt

df = pd.read_csv(r'C:\Users\hoged\OneDrive\Skrivebord\T BA\may-22-data.csv', sep=';')

df['time'] = dt.datetime.strptime('22/05/23 20:40:00', '%d/%m/%y %H:%M:%S')
df['time'] = df['time'] + pd.to_timedelta(df['ms'], 'milliseconds')

fig, ax1 = plt.subplots()

# Make a nice time format in the plots.
xformatter = mdates.DateFormatter('%H:%M')
plt.gcf().axes[0].xaxis.set_major_formatter(xformatter)

# CO2 data.
color1='blue'
ax1.set_ylabel('CO\u2082 (ppm)', color=color1)
ax1.plot(df['time'], df['CO2'], color=color1)
ax1.tick_params(axis='y', labelcolor=color1)

# Instantiate a second axis that shares the same x-axis.
ax2 = ax1.twinx()

# tVOC data.
color2='red'
ax2.set_ylabel('tVOC (pbb)')
ax2.plot(df['time'], df['tVOC_mics'], color=color2)
ax2.tick_params(axis='y', labelcolor=color2)

fig.tight_layout()

plt.show()
```

Figure 27: Python code of graphs